IBM Cloud Application Performance Management
June 2019

User's Guide

Note

Before using this information and the product it supports, read the information in "Notices" on page 1475.

This edition applies to the June 2019 version of IBM® Cloud Application Performance Management and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 8. Integrating with other products and components........ 929

Chapter 9. Administering........ 957

Chapter 1. What's new

New features, capabilities, and coverage are available in the latest release.

- For information about the agent version in each release or refresh, see "Change history" on page 54.
- For detailed system requirements, see the Software Product Compatibility Reports for IBM Cloud Application Performance Management - Agents V8.1. Expand the **Report filters** twisty, click **Edit**, select the agents and data collectors that you want to view, and then click **Apply**. The detailed information such as operating systems, prerequisites, and supported software is displayed for you.

September 2020

Expanded platform support for agents

CentOS 7 x86-64

- Hadoop agent

CentOS 8 x86-64

- Db2 agent
- Hadoop agent

Ubuntu 16.04 x86-64

- Hadoop agent

Ubuntu 18.04.3 x86-64

- Hadoop agent

Ubuntu 18.04 zLinux

- Web Logic agent
- Oracle Database agent

Ubuntu 20.04 x86-64

- Hadoop agent

Ubuntu 20.04 zLinux

- Db2 agent
- Oracle Database agent
- Web Logic agent

SUSE Linux Enterprise Server 15

- MySQL agent

Agent enhancements

Cassandra agent

Agent is now able to collect data for new node that is added to existing Cassandra cluster.

Db2® agent

- Added diagnostic information for failed SQL queries.
- Resolved the issue of crashing on pLinux LE platform.
- Resolved the issue of attachment failure due to password store limitation.

Hadoop agent

- Added support for monitoring of Journal Nodes and ZKFC on HDP 3.1.5 platform and Cloudera 5.4.3 Hadoop platforms.
- Added support for monitoring of Ambari services such as Infra Solr and Spark 2.
- Added support for monitoring of Cloudera Manager Services like: HDFS,Hive,Hue,Impala,Kafka,Oozie,Sentry,Solr,Spark,YARN and ZooKeeper.
- Added support for monitoring of Hortonworks HDP 3.1.5.

Microsoft Active Directory agent

Added three new attribute groups that are supported on Windows Server 2008 and later versions:

- User Group Computer Management

  It shows count of changes made in user, group and computer accounts such as addition, deletion and modification.
- Enabled Disabled User

  It shows count of enabled and disabled users.
- Logon Failure Count Per Error Code

  It shows count of failed logon attempts such as bad user name and password, disabled and expired account.

Microsoft IIS agent

- Added following new attribute groups:
  - `ASP .NET System Counter`
  - `HTTP method requests per second`
  - `Worker Process Requests`
- Added following new group widgets:
  - ASP.Net Applications - Request Statistics
  - Requests/sec - By HTTP Method
  - Internal Server Error (history)
  - Internal Server Error (history)
  - Worker Process Requests - Last 15 mins

Microsoft SharePoint Server agent

Added new attribute group `Document Library`.

NetApp Storage agent

- Added support for Windows Server 2019 Datacenter and Windows Server 2019 Standard.
- Implemented retry count enhancement that limits the number of connection attempts in the event of connection failure. Connection attempts can be configured via the environment variable KNU_DATA_PROVIDER_CONNECTION_RETRY_COUNT.
- Added a new configuration field KNU_API_SERVICES_PORT to agent configuration panel of OCUM API Services tab.
- Fixed the issue when API Services configuration is unavailable. Agent is able to fetch the storage objects Disks, Storage-VMs and Qtrees without API Services configuration.

VMware VI agent

- Added support for Windows Server 2019 Datacenter and Windows Server 2019 Standard.
- Implemented retry count enhancement that limits the number of connection attempts in the event of connection failure. Connection attempts can be configured via the environment variable KVM_DATA_PROVIDER_CONNECTION_RETRY_COUNT.

- Added new attribute Snapshot_Age in the VM_Snapshots group.
- Added new attribute Creation_Timestamp that supersedes Creation_Time.

June 2020

Expanded platform support for agents

The following agents and platforms are now supported:

Ubuntu 20.04 LTS

- Linux for System x:
  - Cassandra agent
  - Linux® OS agent
  - RabbitMQ agent
- Linux for System z:
  - Linux OS agent

Cent OS 7.6, 8.0, and 8.1

- DataPower® agent
- Oracle Database agent
- VMware VI agent
- WebSphere® Applications agent

Agent Builder enhancements

You can now create custom agents that can run on the IBM Cloud Pak for Multicloud Management.

You can also reconfigure existing custom agents to run on the Cloud Pak for Multicloud Management. For example, if you previously created a custom agent to run on IBM Tivoli Monitoring or IBM Cloud Application Performance Management, you can now reconfigure this agent and connect it to the Cloud Pak for Multicloud Management. For more information, see “Preparing the agent for Cloud Pak for Multicloud Management” on page 1345.

Agent enhancements

Cisco UCS agent

Added support for Windows Server 2019 Datacenter and Windows Server 2019 Standard.

Hadoop agent

- Label of widget **Block Capacity** is changed from **Block Capacity** to **Number of Blocks**.
- Added support for monitoring of Cloudera CDH 6.3.2 and 5.16

Internet Service Monitoring

- Added support for profileUpgrade tool for Windows platform.
- Two standalone utilities Properties Migration and Profile Migration have been made available for migrating Monitor properties and Internet Services profiles respectively from Legacy versions of the Internet Service Monitoring agent, such as 2.4 and ITM, to APM version of the agent. These utilities are supported with ISM agent version 8.1.4.0.12 onwards. For more information on these utilities, refer to “Migration support” on page 442.

Liberty data collector

Added monitoring support for Liberty 19 and 20.

Linux KVM agent

- Added support for Ovirt Java SDK 4.3. From 8.1.4.0.12 onward, the agent will support RHVM 4.0. Versions older than RHVM 4.0 are not supported from 8.1.4.0.12.
- Data provider is optimized to decrease the overall CPU consumption on Ovirt engine.
- Live snapshot support attribute is deprecated in the Hosts attribute group.

- A limit is added on the number of retries the agent attempts to connect to the data source (RHVM or Hyper-V) if connection fails. The default number for retry attempt is 6. This value can be changed by setting the following variable in the agent instance specific environment file: `KV1_DATA_PROVIDER_CONNECTION_RETRY_COUNT`

Microsoft .NET agent

Added the CLR Remote Monitoring group widget.

Microsoft Active Directory agent

Added the new widget Group Membership for changes of users.

Microsoft Cluster Server agent

Added the following new widgets to the Overview page:

- Quorum Configuration
- CPU Performance
- Cluster Availability

Added the following new widget to the Cluster Node details page:

- Disk Read and Disk Write

Added the following attributes to the existing widget Resource Groups:

- Current RG Node
- Previous RG Node
- RG Node Changed

Microsoft Exchange Server agent

Added the following new widgets:

- Server Health Status
- Email Statistics

Microsoft Hyper-V Server agent

Enabled the help content for the following attributes and attribute groups on dashboard:

- Server Memory Available
- Server Maximum Processor utilization
- Virtual Machine Disk Details
- Processor Load Details
- Disk Details
- Virtual Machines Processor Load
- Network Virtual Switch Details

Microsoft IIS agent

Added three new attribute groups:

- ASP.NET Application Statistics
- Website Inventory
- Website Response Information

Microsoft Office 365 agent

Added support for the following service communication APIs:

- GetServices
- GetCurrentStatus
- GetHistoricalStatus
- GetMessages

Microsoft SharePoint Server agent

Added four new attribute groups:

- Crawl History details
- Active Crawls details
- Web Application information
- Status of Database Connection

MySQL agent

Improved the logging mechanism for incorrect agent configuration.

SAP agent

- Added support of Master Control Panel to handle various configurations of SAP agent at the server.
- Added support for conversion of data received from SAP system to UTF8 format.
- Added support for Solution Manager 7.2 SPS 11.

SAP NetWeaver Java™ Stack agent

Added support for SAP Netweaver 7.5 Application server Java SPS 018.

Skype for Business Server agent

Added the following new widgets to the Overview page:

- IM CCCP Processing
- User Services Conference Notification
- SIP Authentication
- Server Details
- Skype Topology

Sterling File Gateway agent

Added support for Windows Server 2019 Datacenter and Windows Server 2019 Standard.

Sterling Connect Direct agent

Added support for Windows Server 2019 Datacenter and Windows Server 2019 Standard.

Tomcat agent

Enhanced JMX connection implementation that improves the agent performance in terms of threads or memory.

VMware

Added support for CentOS 7 and CentOS 8.

March 2020

Agent enhancements

Expanded platform support for agents

Added Cent OS 7.6, 8.0, and 8.1 support for the following agents:

- HTTP Server agent
- JBoss agent
- Linux OS agent
- PostgreSQL agent

Internet Service Monitoring

- Added support for SLES 15 on x86-64 (64 bit) platform.
- Enhanced the configuration panel to have Active and Description fields for every monitor element listed in profile edit dialog box.

- Added Activate and Deactivate buttons on profile edit grid, so that user can activate or deactivate multiple monitor elements at a time
- Enhanced **Delete** functionality in the profile edit dialog box for user to delete multiple monitor elements at a time.

Microsoft Active Directory agent
Added New widget for Last Logon Information of privilege users.

Microsoft Exchange Server agent
Added widget for DLPPolicyTips and show Data Loss Prevention Policies and Policy tips in Attribute details tab.

Microsoft IIS agent
Added new attribute group called `Web Application`. For more information about prerequisites for this attribute group, see "Prerequisite to install Web Application Attribute Group" on page 515

Microsoft SQL Server agent
Added tolerance support for SQL Server 2019

RabbitMQ agent

- Added support to HTTPS protocol.
- Added configuration panel parameters `Protocol`and `Path to truststore`.

Documentation enhancement
Added HTTP Server monitoring solution overview architecture. For more information, see "Overview of HTTP Server monitoring" on page 265.

December 2019

New Agent

MariaDB agent

The Monitoring Agent for MariaDB offers a central point of management for your MariaDB environment or application. The software provides a comprehensive means for gathering the information required to detect problems early and to prevent them. Information is standardized across the system. You can monitor multiple servers from a single console. By using the Monitoring Agent for MariaDB you can easily collect and analyze MariaDB specific information.

- For information about configuring the agent after installation, see "Configuring MariaDB monitoring" on page 485.
- For information about the dashboards, thresholds, and attributes, see the MariaDB agent Reference.

Expanded platform support for agents
The following agents and platforms are now supported:

Solaris X86-64

- Oracle Database agent
- WebLogic agent

Agent enhancements

Cassandra agent
Added two new attributes called `Agent Hostname` and `Agent Instance Name` in the `Cluster Details`, `Node Statistics`, and `Keyspace Details` attribute groups.

Db2 agent
Added support to monitor `Current Running SQL`.

IBM Integration Bus agent
Added two new group widgets, `TCPIP Client Connections` and `TCPIP Server Connections` in the `Integration Server Status - Detail` page.

Internet Service Monitoring

- Added two new configuration panel variables:
  - *Active*: To select a state for profile element as active or inactive.
  - *sniServerName*: Indicates the name of host/server for which a certificate from SNI enabled web server is required.
- **Default settings** under **Data Validation** Tab for HTTP,HTTPS and DNS monitors are now editable
- Agent now supports & character in **page** field for HTTP and HTTPS monitors
- Agent now support Danish characters in **regex** field of HTTP and HTTPS monitors

  Note: Set the locale to `da_DK` on Linux platform before agent installation to use this feature

Microsoft Active Directory agent

- Added a new widget called `KCC details` in the `Status Overview` page.
- Added following new attribute groups in the Attribute Details tab:
  - `Directory Services`
  - `Kerberos Consistency Checker`
  - `Kerberos Key Distribution Center`
  - `Name Service Provider`
  - `Exchange Directory Service`
- Added New widget for Last Logon Information of privilege users.

Microsoft .NET agent

Added a new attribute called `Request Name` in the `Database Call Details` attribute group. This attribute displays the name of the request that fires the database query.

Microsoft Exchange Server agent

- Added a new widget called `Transport SMTP Recive` in the `Status Overview` page.
- Added following new attribute groups in the Attribute Details tab:
  - `MS Exchange AB`
  - `MS Exchange ADAccess Processes`
  - `MS Exchange ADAccess Caches`
  - `MS Exchange ADAccess Domain Controllers`
  - `MS Exchange ADAccess Forest Discovery`
- Added the following in Attribute details tab.
  - Policyholders widget
  - Data Loss Prevention Policies
  - Policy tips

Microsoft SQL Server agent

Added tolerance support for SQL Server 2019.

Microsoft Hyper-V Server agent

Added support for Windows Server 2019.

Microsoft IIS agent

- Added new group widgets:
  - `System-Main Memory Statistics`
  - `IIS Server- Assigned Memory Usage`
  - `IIS Server- Assigned CPU Usage`

  - Worker Process Details
  - .Net Memory Management
- On each application pool name in Worker Process Details group widget a page is created that shows the historical trend of Requests processed per second, Elapsed time, Requests in Queue, Memory and CPU Utilization.
- On each application pool name in .Net Memory Management group widget a pop-up is added that shows the historical trend of percent time in GC.

Microsoft SharePoint Server agent

- Added a new attribute group called Trace_Log that provides the high severity logs information.
- Added two new group widgets called Trace Log Details and Last 1 Hour Trace Log Count in the Overview page to display the details of recent 100 trace log events and last 1 hour count of unexpected, monitor-able and high level trace logs.

MySQL agent

The agent collects data consistently after server restarts.

NetApp Storage agent

The agent now shows exact list of Qtress that are mapped to Volume.

PostgreSQL agent

The agent now supports PostgreSQL Server version 12.

Response Time Monitoring agent

- A new configuration parameter KT5AARIPTOUSERID is added. It allows you to save Client IP address in User Name property in AAR raw data. By default, it is set as NO. To change the setting, you need to restart the Response Time Monitoring agent.
  - KT5AARIPTOUSERID=NO: If the value is NO, the Response Time Monitoring agent agent will save username of transaction to userID property of AAR.
  - KT5AARIPTOUSERID=YES: If the value is YES, the Response Time Monitoring agent will save Source IP address of transaction to userID property of AAR.
- The Response Time Monitoring agent now supports specifying KT5AARIPTOUSERID value in silent configuration.
- The title of existing group widget Worst By User - Top 5 is changed to Worst By User - Top 20. The group widget is changed to display top 20 users with highest percentage of transaction failures over the selected period.

VMware

- Agent now supports fetching of the IP address or Host name of vCenter from vSphere API call, instead of showing Configured Address as it is from configuration panel. User can activate this feature by setting the flag in agent environment to Y. For example, KVM_RETRIEVE_HOSTNAME_FROM_API=Y.
- Retry count now can be given a limit to restrict the connection attempts with data source. For example, KVM_DATA_PROVIDER_CONNECTION_RETRY_COUNT=1000, adding this variable in agent environment file would put a lock on connection retry count in case of connection failure with vCenter. 1000 indicates agent would try till 1000 subsequent unsuccessful connection attempts and then would bring down the data provider process with a log message NO MORE ATTEMPTS OF CONNECTION;STOPPING THE DATA COLLECTION, TO RESUME MONITORING PLEASE RESTART THE AGENT. TO HAVE MORE ATTEMPT OF CONNECTIONS, RESET THE VALUE OF THE VARIABLE KVM_DATA_PROVIDER_CONNECTION_RETRY_COUNT. Default value to retry connection attempts is 6, user can set the desired threshold as per the requirement.
- Agent supports configuring of Instance specific heap size to efficiently utilize the allocated memory on the system. For example, KVM_CUSTOM_JVM_ARGS= -Xmx512m, by setting this variable in instance’s environment file would mean that instance is configured to use 512 MB of

heap memory. The size can be changed based on the total count of vCenter objects an instance is monitoring.

September 2019

Expanded platform support for agents

The following agents and platforms are now supported:

Solaris X86-64

- Db2 agent
- SAP agent
- Sybase agent
- UNIX OS agent
- WebSphere Applications agent
- WebSphere MQ agent
- IBM Integration Bus agent

RHEL on x86-64 (64 bit)

- Internet Service Monitoring agent
- Microsoft SQL Server agent
- Sybase agent

RHEL on POWER Little Endian (ppc64le)

- RabbitMQ agent

Agent enhancements

Db2 agent

- The agent now supports Db2 server version 11.5.

Hadoop agent

- A new configuration parameter **`Unique Cluster Name`** which is a unique name for Hadoop Cluster indicating its version and flavor is added in the configuration panel.
- The Hadoop agent now shows display item for the thresholds created on Ambari Services.
- The Hadoop agent now supports monitoring of Streaming Analytics Manager service in Hadoop cluster.
- The Hadoop agent now supports monitoring of Schema Registry service in Hadoop cluster.

HTTP Server agent

- The agent now supports Oracle HTTP server on Solaris Sparc.

Internet Service Monitoring

- The agent now supports IBM Tivoli® Netcool/OMNIbus.
- The agent is enhanced to delete profiles from the existing profiles and rename the existing profiles.

MongoDB agent

- The agent now supports MongoDB database version 4.x.

MySQL agent

- FQDN attribute is added for Application Availability in the Help section.
- Fixed Tooltip display for IP Address Agent Configuration parameter.
- The following new attributes are added for monitoring in IBM Cloud App Management.

    - Database Size Information
    - Error Information
    - Count of Db Instance Lock
    - User's connection details
    - Process list details
    - Events Information

PostgreSQL agent

- Two new situations called `Deadlocks_Count_Crit` and `Deadlocks_Count_Warn` are added to monitor number of deadlocks in a database, which will help you to address the exact issue for the deadlocks.
- One new attribute group called `Deadlocks_Info` is added to check the deadlock details.

Sybase agent

- FQDN attribute is added for Application Availability in the Help section.

Synthetic Playback agent

- Firefox V68.0 ESR is now supported.
- System proxy, PAC proxy, and no proxy configurations are now supported.

Tomcat agent

- The agent is enhanced with metrics and UI views for monitoring Heap/Non-heap memory pool usage for JVM.
- The agent is enhanced with metrics and UI views for monitoring threads, and class loading information for JVM.
- The agent UI now displays FQDN on Server Information view.

UNIX OS agent

- The agent is now updated with the custom scripting feature. Shell scripts, PERL scripts and other types of scripts can be used.

VMware VI agent

- A new configuration field named **`KEY_STORE_PASSWORD`** is added. It allows user to configure the agent with new key-store password set for the agent JRE.

June 2019

Expanded platform support for agents

The following agents and platforms are now supported:

Red Hat Enterprise Linux (RHEL) 8

The following agents and data collectors now support RHEL 8. Before installing agents on RHEL 8, be sure to read the "Specific operating systems" on page 130 section of "Preinstallation on Linux systems" on page 129.

RHEL 8 on x86-64 (64 bit)

- Cassandra agent
- Cisco UCS agent
- DataPower agent
- DataStage® agent
- Db2 agent
- Hadoop agent
- HTTP Server agent

- Integration Agent for Netcool®/OMNIbus
- Internet Service Monitoring agent
- J2SE data collector
- Linux KVM agent
- Linux OS agent
- MongoDB agent
- MQ Appliance agent
- MySQL agent
- NetApp Storage agent
- Node.js data collector
- PHP agent
- Python data collector
- PostgreSQL agent
- RabbitMQ agent
- Response Time Monitoring Agent
- Ruby agent
- SAP agent
- SAP HANA Database agent
- SAP NetWeaver Java Stack agent
- Sterling Connect Direct agent
- Sterling File Gateway agent
- Sybase agent
- Tomcat agent
- VMware VI agent
- WebSphere Applications agent
- WebSphere MQ agent

RHEL 8 on System z

- Db2 agent
- Linux OS agent
- Node.js data collector
- Python data collector
- Response Time Monitoring Agent
- WebSphere Applications agent
- WebSphere MQ agent

RHEL 8 on POWER Little Endian (ppc64le)

- Db2 agent
- Hadoop agent
- J2SE data collector
- Linux OS agent
- MySQL agent
- Node.js data collector
- SAP NetWeaver Java Stack agent
- WebSphere Applications agent

- WebSphere MQ agent

Solaris Sparc 10 and 11

- JBoss agent
- Oracle Database agent
- WebLogic agent

Windows Server 2019

- WebSphere Applications agent

Agent enhancements

Hadoop agent

- The Hadoop agent now supports monitoring of HDF 3.3 (with HDP 3.1.0) Ambari service Big SQL 6.0.
- The Hadoop agent now supports SUSE Linux Enterprise Server (SLES) 15 on x86-64 platform.

HMC Base agent

The HMC Base agent supports HMC V9.1.

Integration Agent for Netcool/OMNIbus

The agent has been updated to support the Red Hat Enterprise Linux (RHEL) 8 and SUSE Linux Enterprise Server (SLES) 15

Internet Service Monitoring agent

The agent now has Service Assurance Agent monitor, which monitors Cisco Service Assurance Agent probes.

Microsoft IIS agent

The agent is enhanced with tolerance provision to the Windows 2019 server. This enhancement displays the FTP site data for agent that is installed on Windows 2019 server.

MongoDB agent

The agent now supports Red Hat Enterprise Linux (RHEL) 8 on x86-64 (64 bit) platform.

Python data collector ifix02

The data collector now supports Django 1.10 and higher.

SAP agent

The SAP agent now supports the following platforms:

- Red Hat Enterprise Linux (RHEL) 8 on x86-64 (64 bit)
- SAP NetWeaver Application Server 7.52 (SAP Basis 752)

SAP HANA Database agent

The SAP HANA Database agent is enhanced with the following features:

- Hostname is added in the :HDB Subnode node of SAP HANA Database agent for its unique identification.
- The agent now supports scale out architecture.
- Red Hat Enterprise Linux (RHEL) 8 on x86-64 (64 bit) and Linux ppc64le platforms.
- SUSE Linux Enterprise Server (SLES) 15 on x86-64 (64 bit) platform.
- A new attribute Trimmed Host is added under System Database attribute group.

SAP NetWeaver Java Stack agent

The SAP NetWeaver Java Stack agent supports the following platforms:

- Red Hat Enterprise Linux (RHEL) 8 on x86-64 (64 bit)
- SUSE Linux Enterprise Server (SLES) 15 on x86-64 (64 bit)
- Windows Server 2019 DE and SE

- Windows Server 2016 DE and SE

Synthetic Playback agent

- Supports .side script recorded by Selenium IDE 3.2.X, 3.3.X, or 3.5.X
- Supports playback by Firefox ESR 60.5.1
- Supports the **`wait`**, **`flow control`**, and **`linkText locator type`** Selenium IDE commands

Skype for Business Server agent

The Skype for Business Server agent is enhanced with the following features:

- The agent now supports Skype for Business Server 2019.
- Two new group widgets such as `Database-Throttled Requests(DBStore)` and `Database-Throttled Requests(SHAREDDBStore)` are added on the Overview page that displays number of requests throttled by the Skype for Business Server due to high latency of the database queue for DBstore and shared DBstore.

Prerequisite scanner

The **`IGNORE_PRECHECK_WARNING`** command is now available as an alternative to the **`SKIP_PRECHECK`** command. For more information, see "Bypassing the prerequisite scanner" on page 146.

Documentation enhancement

A page is created to help you quickly find out the version information and change history for each agent and data collector. See "Change history" on page 54.

March 2019

Expanded platform support for agents

The following agents and platforms are now supported:

Windows Server 2019

- Cassandra agent
- DataStage agent
- Db2 agent
- Hadoop agent
- Internet Service Monitoring
- Microsoft Active Directory agent
- Microsoft Cluster Server agent
- Microsoft IIS agent
- Microsoft Exchange Server agent
- Microsoft SQL Server agent
- MySQL agent
- PostgreSQL agent
- RabbitMQ agent
- SAP agent
- SAP HANA Database agent
- Sybase agent
- Tomcat agent
- Windows OS agent

Solaris SPARC 10 and 11

- Db2 agent
- HTTP Server
- MySQL agent
- SAP agent
- Sybase agent
- UNIX OS agent
- WebSphere Applications agent

Monitoring Agent for Cassandra

The Cassandra agent is enhanced with the following features:

- Added support for Windows Server 2019 Operating System.
- Added detailed logging for troubleshooting.

Monitoring Agent for Db2

The Db2 agent is enhanced with the following features:

- The Db2 agent now supports Windows Server 2019.
- The Db2 agent now supports Solaris SPARC 10/11 platforms.

Monitoring Agent for Hadoop

The Hadoop agent is enhanced with following features:

- Added support to monitor the SSL enabled Hadoop BigInsights®, Hortonworks and Cloudera clusters.
- Added support to test the connection to SSL enabled Hadoop cluster.
- Added support of Windows Server 2019 Operating System (Datacenter and Standard Editions).
- Added support for monitoring Hadoop offering: Cloudera 6.1.1 (CDH 6.1.1).
- Added support for monitoring Hadoop offering: Hortonworks 3.1.0 (HDP 3.1.0).

Monitoring Agent for IBM Integration Bus

The IBM Integration Bus agent is enhanced with the following feature:

- Added tolerance support to monitor IBM App Connect Enterprise V11. For more information, see "Configuring the IBM Integration Bus agent" on page 274.

Monitoring Agent for Microsoft Internet Information Services

The Microsoft IIS agent now supports Windows Server 2019 Operating System.

Monitoring Agent for InfoSphere® DataStage

The DataStage agent is enhanced with the following features:

- Added support for Windows Server 2019 Operating System.
- Added query timeout to data collection queries for better performance of the agent.

Monitoring Agent for Microsoft Active Directory

The Microsoft Active Directory agent is enhanced with the following features:

- Added support for Windows Server 2019.
- Added new attribute group `AD_Services_Status` that provides services state that are related to Active Directory Server.
  Based on the services state, it determines the Server Status of Active Directory.
- Added new situation `AD_Server_Status` that monitors Active Directory Server Status.

- Added new attribute group Root_Directory_server that provides active version and monitored OS name.

Monitoring Agent for Microsoft Cluster Server

The Microsoft Cluster Server agent is enhanced with the following features:

- Added support for Windows Server 2019 Operating System.
- Added the CLUSTER_SERVICE_VERSION attribute.

Monitoring Agent for Microsoft Exchange Server

The Microsoft Exchange Server agent is enhanced with the following features:

- Added support for MS Exchange Server 2019.
- Added new attribute group MSExchange MAPIoverHTTP that provides information about MAPI over HTTP protocol statistics.

Monitoring Agent for Internet Services

The Internet Service Monitoring agent is enhanced with following features:

- Added support for LDAP, NTP, NNTP, SOAP, SNMP, SIP, RTSP, RPING, RADIUS and TFTP monitors.
- Added support for Windows 2008 R2 server OS and Windows server 2019.

Monitoring Agent for Microsoft SQL Server

The Microsoft SQL Server agent now supports Windows Server 2019.

Monitoring Agent for MySQL

The Monitoring Agent for MySQL is enhanced with the following features:

- Added support for Windows Server 2019.
- Added support for Solaris SPARC 10/11 platforms.
- Added capability to set additional properties to the agent initiated JDBC connection with the MySQL server.

Monitoring Agent for PostgreSQL

The PostgreSQL agent now supports Windows Server 2019 Operating System.

Monitoring Agent for RabbitMQ

The RabbitMQ agent now supports Windows Server 2019 Operating System.

Monitoring Agent for Skype for Business Server

The Skype for Business Server agent is enhanced with the following features:

- Added support for Windows Server 2019 Operating System.
- Added a new attribute group called KQL_Server to display the Skype for Business Server product related information.
- Added a new situation called Skype_Server_Down to monitor the Skype for Business Server status based on the server front-end and IM conferencing services status.

Monitoring Agent for SAP Applications

The SAP agent is enhanced with following features:

- Added case sensitive password feature for application user being used between SAP agent and SAP server.
- Added support for Windows Server 2019 Operating System (Datacenter and Standard Editions).
- Added support to see the Long Running Jobs present in SAP System for more than 24 hours.
- Improved performance of /IBMMON/ITM_MAIALRT_INX function Module.
- Added support for Solaris v10 and v11 SPARC Operating Systems.

- Added trim feature to SAP host name in order to match the Managed System Name's maximum 32 characters limit.

Monitoring Agent for SAP HANA Database

The SAP HANA Database agent is enhanced with following features:

- Added support to discover the tenant databases when the tenant db name and HANA system SID are same.
- Added support for Windows Server 2019 Operating System (Datacenter and Standard Editions).

Monitoring Agent for Sybase Server

The Sybase agent is enhanced with the following features:

- Added support for Windows Server 2019.
- Added support for Solaris SPARC 10/11 platforms.
- Enhanced Sybase query for better concurrency and reduced locking.

Monitoring Agent for Tomcat

The Tomcat agent now supports Windows Server 2019 Operating System (Datacenter and Standard Editions).

Monitoring Agent for UNIX OS

The UNIX OS agent is enhanced with the following feature:

- Added support for Solaris SPARC 10 and 11.

Monitoring Agent for VMware VI

The VMware VI agent has been refreshed to ignore the unavailable(-1) values while displaying the trend of average on the graph for all multi-line charts.

Monitoring Agent for WebSphere Applications

The WebSphere Applications agent is enhanced with the following features:

- Added support for Solaris SPARC 10 and 11.
- Added support for monitoring WebSphere® Extreme Scale. You can configure monitoring for an Extreme Scale zone, or several zones, under the node for any server belonging to the zone or zones. You can drill down to view information for different servers, map sets, and partitions within the zone or zones. For more information, see "Configuring the WebSphere Applications agent to monitor WebSphere Extreme Scale" on page 902.

Monitoring Agent for WebSphere MQ

The WebSphere MQ agent is enhanced with the following features:

- Added support for SLES 15 xLinux.
- Added support to collect statistics for the queue manager and display the collected data. For more information, see "Enabling queue statistics monitoring for the queue manager of IBM MQ" on page 922.

Data collector enhancements

J2SE data collector

The J2SE data collector is enhanced with the following features:

- Added support for OpenJDK 9, 10, and 11 versions.
- Added support of Windows Server 2019 Operating System (Datacenter and Standard Editions).
- Added feature to auto-discover the J2SE application specific classes and methods for Transaction Tracking and Diagnostic Data monitoring.

Selenium IDE 3.2.X and 3.3.X for synthetic scripts

If your subscription includes the IBM Website Monitoring on Cloud add-on, Selenium IDE versions 3.2X and 3.3.X are now supported; scripts and test suites are saved in `.side` format rather than the .html format used by older versions of Selenium IDE. If you have existing .html scripts, you can still use them. In some instances, you might want to edit the .html scripts or re-record them in the new .side format.

For more information, see these subtopics of "Managing synthetic transactions and events with Website Monitoring" on page 1006: "Recording synthetic scripts" on page 1007, "Structuring complex scripts" on page 1009, and "Updating scripts from earlier Selenium IDE versions" on page 1010.

December 2018

New agent

Monitoring Agent for IBM Cloud

The Monitoring Agent for IBM Cloud collects virtual machine inventory and metrics from your IBM Cloud (Softlayer) account. Use the IBM Cloud agent to track how many virtual devices you have configured and running in IBM Cloud. You can see what resources are allocated to each virtual device in the detailed dashboard page, which also shows information like the data center a device is located in, the operating system, and the projected public network bandwidth for the month.

Agent enhancements

Monitoring Agent for Cassandra

The Cassandra agent is enhanced with the following features:

- Added a new threshold named `Cassandra_Cluster_Down` that monitors the state of the monitored instance.
- Added support for Ubuntu 18.04 operating system.
- Added support for SUSE Linux Enterprise Server 15 platform.

Monitoring Agent for Db2

The Db2 agent is enhanced with the following features:

- The Db2 agent now supports HADR Monitoring Capabilities for multiple standbys.
- The Db2 agent now supports new value `Stopped` for Database Status attribute. The `Stopped` status indicates that the database is not active and having zero active connections while it is healthy and ready to accept new connections.
- Added the new widget `Db2 Server Information` to display Db2 Server details.
- Added the new page `HADR Status - Local Databases` to display the information about partner databases in the following new widgets:
  - `HADR Databases Details` is the table widget which displays important attribute values for partner database.
  - `Log Gap (History)` is the graph widget which displays log gap trend vs time.
  - `Standby Flag Status` is the table widget which displays standby flag status values.
- Added the new predefined threshold `UDB_HADR_Aux_Standby_Disconnect` to monitor secondary standby databases in HADR environment.
- The Top 5 database memory usage widget is updated to show the correct value.
- The Db2 agent now supports the following platforms:
  - Ubuntu zLinux 18.04
  - SUSE Linux Enterprise Server 15 on x86-64 (64 bit)
  - SUSE Linux Enterprise Server 15 for zLinux
  - SUSE Linux Enterprise Server 15 for Power® Linux Little Endian

Monitoring Agent for InfoSphere DataStage

The InfoSphere DataStage agent is enhanced with the following features:

- Added capability to disable data collection for selected attribute groups.
- Optimized data collection for Job Runs attribute group.

- Added support for SUSE Linux Enterprise Server 15 platform.

Monitoring Agent for Internet Services

Internet Service Monitoring Agent now supports Windows 64-bit and Linux 64-bit platforms.

Monitoring Agent for Microsoft .NET

The .NET agent is enhanced with the following features:

- The .NET agent now tracks the failed requests. The status of these requests is shown as `failed` under the Latest Requests group widget of the Middleware Transaction Details page. Also, the Latest Errors group widget lists the recent failed requests along with the status code and error description.
- The .NET agent also monitors the user data available through `ASP.NET` Identity and `ASP.NET` sessions. The user data is displayed in the Users Top–5 group widget of the Middleware Transaction Details page.

Monitoring Agent for MongoDB

MongoDB agent now supports SUSE Linux Enterprise Server 15 platform.

Monitoring Agent for NetApp Storage

The NetApp Storage agent is enhanced with the following features:

- A new Search box is added on the Event Details page that filters the event data based on the search criteria.
- A new Details page is added for LUNs.
- The user is now able to check the details of related devise mapped to each storage object on the **Details** page.

Monitoring Agent for OpenStack

Support was added for monitoring virtual machine instances, such as the usage of the virtual machine instance CPU, memory, disk, and network interface controller.

Monitoring Agent for PostgreSQL

- Support added for SUSE Linux Enterprise Server 15
- Optimized data collection for CPU and Memory attribute groups

Monitoring Agent for RabbitMQ

Support added for SUSE Linux Enterprise Server 15

Monitoring Agent for SAP Applications

SAP agent now supports the following platforms:

- SUSE Linux Enterprise Server 15 platform
- SAP NW RFC SDK 7.50

Monitoring Agent for Skype for Business Server

The Skype for Business Server is enhanced with the following features:

- Synthetic Transaction Commands in Synthetic Transaction Module are now executable by already configured Test Users. To avail this feature, disable the `Use Agent Configuration Values` in Agent configuration panel and provide value of `Pool FQDN` for which Synthetic Commands are to be executed. Make sure the Test User is configured through NewCsHealthMonitoringConfiguration command for Identity provided in Pool FQDN field of Agent Configuration Panel.
- Users can now disable Synthetic Commands. To disable any particular command from execution, provide `false` against that command name in `LyncSyntheticTrans.exe.config` file present at location `<CANDLE_HOME>\tmaitm6` for 32-bit version and `<CANDLE_HOME>\TMAITM6_x64` 64-bit version.

Monitoring Agent for Tomcat

- New attribute group Cluster is added. It contains information of properties of a cluster.

- New widget Cluster Information is added. This widget displays information of attribute group Cluster. It will not display any data if the agent is monitoring a non-cluster Tomcat setup.
- Configuration panel variable *Tomcat Server Port* is added. This variable represents the port on which Tomcat server is running. Default value of the variable is 8080.

Monitoring Agent for VMware VI

- Component page has been enhanced to show the IP Address or Hostname of configured vCenter and its connectivity with the agent.

WebSphere Infrastructure Manager agent

The WebSphere Infrastructure Manager agent now supports AIX®.

WebSphere MQ agent

The WebSphere MQ agent is now supported on IBM WebSphere MQ 9.1.

Data collector enhancements

J2SE data collector

The J2SE data collector is enhanced with the following features:

- Added support for SUSE Linux Enterprise Server 11 for Power Linux Big Endian (64 bit).
- Added support for Power Linux Big Endian (pLinux BE) (64 bit).
- Added support for Power Linux Little Endian (pLinux LE) (64 bit).
- Added configuration module and Jetty server monitoring.

Expanded platform support for agents

The following agents and platforms are now supported:

SUSE Linux Enterprise Server 15 platform

- Cassandra agent
- DataPower agent
- DataStage agent
- Db2 agent
- HTTP Server
- IBM Integration Bus agent
- Linux OS agent
- MongoDB agent
- OpenStack agent
- PostgreSQL agent
- Monitoring Agent for RabbitMQ
- SAP agent
- WebSphere MQ agent

Power Linux

- J2SE data collector

Ubuntu 18.04

- Cassandra agent
- IBM Integration Bus agent
- OpenStack agent
- Linux OS agent
- RabbitMQ agent
- WebSphere MQ agent

Power 9 Support

Power 9 is now supported for all agents.

What's new for the October 2018 refresh of V8.1.4

Integration with Cloud Event Management

Cloud Event Management provides real-time incident management across your services, applications, and infrastructure. Now, with the integration between Cloud Event Management and IBM Cloud Application Performance Management, all the events that are generated in Cloud APM are sent to Cloud Event Management.

September 2018

New agent available

Monitoring Agent for MQ Appliance

The MQ Appliance agent provides monitoring information that is specific to the MQ appliance level on MQ Appliances, for example, CPU, memory, storage, sensors, and queue managers summary information.

Agent enhancements

Monitoring Agent for Db2

The Db2 agent now supports Power Linux Big Endian operating system.

Monitoring Agent for Hadoop

- The Hadoop agent now monitors the status of two more services: SmartSense and Druid.
- The Hadoop agent now supports Hortonworks Data Platform (HDP) 3.0.0.

Internet Service Monitoring agent Agent

Edit functionality for Internet Service Monitoring agent Agent is enhanced. All the monitors which have configurable parameters can be edited.

Monitoring Agent for MySQL

The MySQL agent now supports monitoring MySQL v8.0.11.

Monitoring Agent for NetApp Storage

The NetApp Storage agent is enhanced with the following features:

- A new widget called Overall Events Summary is added on the NetApp Storage Instance page. It displays the cumulative count of events. You can view all the events that occurred across the environment, irrespective of severity or object by clicking the status bar represented as Total Events.

  Also, an Event Status column is added in every objects table, which shows event status prioritized based on time and then the level of severity.

  The NetApp Storage Instance page now displays Events Summary table rather than a chart.
- The Aggregate Details page is updated to display the related devices that are associated with the selected aggregate.

Monitoring Agent for SAP Applications

Support added for SAP NW RFC SDK 750.

Monitoring Agent for SAP HANA Database

Two new features are added:

- SAP HANA Database can be monitored in stand by mode.
- SAP HANA Database agent supports the Big Endian platform for Power System.

Monitoring Agent for VMware VI

- Monitoring of `HostVFlashManager` is now supported
- The ESX Server Dashboard now shows the count of virtual machines that are in their Critical, Warning and Normal status with respect to CPU utilization.

New Platform: Linux on POWER

There is a new platform available. The following agents are now supported on Linux on POWER:

- Db2 agent
- IBM Integration Bus agent
- Linux OS agent
- SAP HANA Database agent
- WebSphere MQ agent
- WebSphere Applications agent

July 2018

New agents available

Monitoring Agent for Sybase Server

The Sybase agent offers a central point of management for distributed databases. It collects the required information for database and system administrators to examine the performance of the Sybase server system, detect problems early and prevent them.

Agent enhancements

Monitoring Agent for Hadoop

- Support added for monitoring Hadoop services such as, Mahout, Atlas, and Falcon.
- Support added for monitoring Hadoop offering: Cloudera CDH 5.13.
- Support added for monitoring Hadoop offering: Hortonworks HDP 2.6.4.

Monitoring Agent for HMC Base

HMC V8 R8.7.0 is now supported.

Monitoring Agent for HTTP Server

Support for the 64-bit Apache HTTP Server on Windows was added.

Monitoring Agent for Microsoft .NET

The Monitoring Agent for Microsoft .NET is enhanced as follows:

- The IIS Response Time module now monitors the subtransaction and render time breakdown through JavaScript injection for ASP`.`NET web forms (.aspx pages) and ASP`.`NET MVC razor views, which satisfy following conditions:
  - The page meets the W3C HTML standards.
  - Response headers contain `Content-Type: text/html, application/xml,application/json`.
  - Response content includes the *<head>* element.
- The .NET agent uploads the deep dive data to Diagnostic Query Engine (DQE) service on the APM server. The DQE service deep dive dashboard quickly loads and displays the data.
- The new threshold **`NET_Slow_IIS_Request_Crit`** is added that triggers when the Slow Top 10 widget has requests with response time greater than 500 milliseconds.
- The selective filtering tool is updated with the search box to search an application pool from the list of application pools.
- The **`ProcListCaller`** utility is added to provide the list of processes that have loaded the .NET Agent CLR profiler (`CorProfLog.dll`).

Monitoring Agent for Microsoft SQL Server

- The Microsoft SQL Server agent now supports multiple collations in `ERRORLOG` parsing based on collation settings in `koqErrConfig.ini` file. When the `koqErrConfig.ini` file does not contain any valid collation settings, you will be able to see only the default English error message with severity level more than the default severity level, if any. The default severity level is 17. All the collations exist in the `koqErrConfig.ini` file will be considered while parsing the ERRORLOG file. So only the collations that are in use should be added in the `koqErrConfig.ini` file. The ERRORLOG parsing is case sensitive, you must ensure the collation keyword values in the `koqErrConfig.ini` file are exactly the same as the keyword values found in ERRORLOG file or in the reference `koqErrConfigSample.ini` file. Note that the changes made in the `koqErrConfig.ini` file is not preserved during agent upgrade, you must make a backup before agent upgrade.

- The agent also provides the utility tool **`koqVerifyPermissions.exe`** to check if an existing SQL Server user has sufficient permissions to monitor the Microsoft SQL Server. If an existing SQL Server user does not have sufficient permissions, you can use the utility tool **`permissions.cmd`** as an alternative to grant the minimum permissions to an existing SQL Server user for data collection.

Monitoring Agent for NetApp Storage

The Monitoring Agent for NetApp Storage is enhanced as follows:

- The new Component page is added to display the details of agent connection state whether the data provider is up or down, along with IP address of monitored data sources. The individual status bar represents the number of nodes, aggregates, volumes, and disks that are in critical, normal, warning, or unknown state.
- The new NetApp Storage Instance page is added to highlight key properties of clusters, aggregates, volumes, disks, and vServers. It also displays the Events Summary chart with count of events that are occurred on each available storage entity or object across the environment. For example, if there are 12 volumes that are configured and each volume has two events with severity as Critical, then the Event Summary chart depicts the total count of events that are occurred on all the volumes available in an environment. In this case, the chart shows a bar with 24 critical events against the volume as an entity plotted on the X-axis.
- The Node Details page is updated to show network port details.
- The Volume Detail page is updated to display details of the associated snap mirrors and LUNs count for each selected volume.
- The vServers Details page is updated to display information about network logical interfaces.

Monitoring Agent for Tomcat

Tomcat V9.0.5 server is now supported.

Monitoring Agent for WebSphere MQ

Remote monitoring is supported. Two configuration parameters are added for the agent to be able to collect monitoring data for a remote queue manager. However, these configuration parameters do not have any effect on a local queue manager. If you want to configure the agent to monitor a local queue manager, you can press Enter to skip specifying these parameters.

For more information about agent configuration, see "Configuring the WebSphere MQ agent" on page 915.

EIF slot customization enhancement

You can now add multiple attribute values and literal values to the EIF slot. Instead of, say, a `Disk free percent is` **`Disk_Free_Percent`** message for a threshold that tests for low available disk space, you could have `Disk free percent is` **`Disk_Free_Percent`** `and inodes free percent is` **`Inodes_Free_Percent`**. The forwarded message might look like this: `Disk free percent is 13 and inodes free percent is 9`. For more information, see "Customizing an event to forward to an EIF receiver" on page 972.

April 2018

New agent available

Monitoring Agent for AWS Elastic Load Balancer

The Amazon ELB agent provides you with a central point of monitoring for the health, availability, and performance of your AWS Elastic Load Balancers. The agent displays a comprehensive set of metrics for each load balancer type-application, network and classic-to help you make informed decisions about your AWS Elastic Load Balancer environment.

Agent enhancements

Response Time Monitoring Agent

The IBM HTTP Server Response Time module now supports IBM HTTP Server version 7, 8, and 9 on Windows.

Monitoring Agent for Node.js

By default, user sensitive information, such as cookies, HTTP request contexts, and database request contexts are no longer collected by the Node.js data collector. You can change this default behavior by specifying the new environment variable, *SECURITY_OFF*.

Monitoring Agent for Amazon EC2

The component name now reflects the agent name.

Extended data retention support is added.

Monitoring Agent for WebLogic

Transaction tracking and deep-dive diagnostics are enabled on AIX. Previously these features were only enabled on Linux and Windows.

The Request Summary drill-down for servlets that are implemented with annotations for transaction tracking and deep-dive diagnostics is enhanced.

Monitoring Agent for Skype for Business Server

Support for Windows Server 2016.

Monitoring Agent for Sterling File Gateway

Agent fetches events for failed file transfer as a default behavior. You can change this default behavior by specifying the appropriate value for the new environment variable **`KFG_ALL_FGEVENTS`**.

Monitoring Agent for Sterling Connect Direct

The agent logging feature is improved. For more information, see Troubleshooting section.

Data collector enhancements

Node.js data collector

By default, user sensitive information, such as cookies, HTTP request contexts, and database request contexts, are no longer collected by the Node.js data collector. You can change this default behavior by specifying the new environment variable, *SECURITY_OFF*.

Remember: To get this enhancement, you must download and apply the IBM Cloud Application Performance Management Node.js data collector interim fix 1 from IBM Fix Central. For more information, see Interim Fix 1 Readme File.

J2SE data collector

Support was added for auto-discovery of entry point class (main class) and alias name of the J2SE application.

Transaction tracking and deep-dive diagnostics can be enabled and disabled locally by using configuration scripts.

Documentation enhancements

See Agent version in Cloud APM releases.

The Agent and data collector capabilities in each offering table is simplified to improve readability. For more information, see "Capabilities" on page 56.

February 2018

New agent available

Monitoring Agent for Azure Compute

The Azure Compute agent provides you with a central point of monitoring for the health, availability, and performance of your Azure Compute instances. The agent displays a comprehensive set of metrics to help you make informed decisions about your Azure Compute environment. These metrics include CPU usage, network usage, and disk performance.

Monitoring Agent for Sterling Connect Direct

You can use the Sterling Connect Direct agent to monitor health and performance of Sterling Connect Direct server. It monitors Sterling Connect Direct server's features, such as file transfer activities, scheduled processes, hold and wait queue processes. The agent supports remote monitoring and it is multi-instance.

Monitoring Agent for Sterling File Gateway

The Sterling File Gateway agent monitors the Sterling File Gateway application, which is used for transferring files between internal and external partners by using different protocols, different file naming conventions, and different file formats. It also supports the remote monitoring feature.

Agent enhancements

Monitoring Agent for DataPower

Transaction tracking between the WebSphere MQ agent and DataPower agent is supported.

Monitoring Agent for Db2

Support was added for remote monitoring.

Monitoring Agent for Hadoop

Support was added for monitoring the status of Hadoop services, such as HBase, MapReduce2, Tez, and Ranger.

Support was added for monitoring Hadoop offering: Cloudera CDH 5.12.

Monitoring Agent for InfoSphere DataStage

Support was added for MS SQL as metadata repository.

Support was added for Windows operating system.

Monitoring Agent for Tomcat

Transaction tracking and deep-dive support for PLinux by upgrading the agent framework with the 8.1.4.0-IBM-APM-SERVER-IF0001 patch.

Monitoring Agent for SAP Applications

Enhancement to CCMS feature: automation of `idx` file deletion. This automation works only when the SAP system is restarted.

Monitoring Agent for Microsoft .NET

Support was added for end user transactions by using IIS Response Time module.

Monitoring Agent for Skype for Business Server

The name of the agent is changed from Monitoring Agent for Microsoft Lync Server to Monitoring Agent for Skype for Business Server.

Monitoring Agent for Linux KVM

Support was added for RHEV-M 4.x.

Monitoring Agent for Linux OS

Memory upload interval is changed to 1 minute.

The IP address associated with the network interface is displayed on the Linux OS dashboard and System Information widget.

Monitoring Agent for UNIX OS

Memory upload interval is changed to 1 minute.

Data collector enhancements

J2SE data collector

Support was added for Spring Boot Applications.

Availability Monitoring enhancements

With the Availability Monitoring add-on, you can now create whitelists and blacklists to specify URLs that your tests can and cannot access. Your whitelist and blacklist control which dependencies and resources contribute to the response times of your tested web applications, such as third-part metrics. Filter URLs by scheme, domain, or file type by using wildcard characters.

December 2017

New agent available

Monitoring Agent for InfoSphere DataStage

You can use the DataStage agent to monitor the health and performance of the DataStage server resources, such as engine services, engine systems, job activity, job run status, and details of job runs. This agent supports remote monitoring.

Agent enhancements

Monitoring Agent for Hadoop

Support was added for monitoring a Hadoop cluster that is secured with the Kerberos SPNEGO-based authentication, which uses Active Directory Key Distribution Center (KDC).

Support was added for testing connection to hosts of a Hadoop cluster that is secured with Kerberos SPENGO-based authentication, which uses MIT or Active Directory as Key Distribution Center (KDC).

Support was added for monitoring the following Hadoop offerings: Cloudera CDH 5.10 and CDH 5.11.

Support was added for monitoring the status of Hadoop services, such as Flume, Kafka, Titan, Spark, Knox, Pig, Slider, and Solr.

Monitoring Agent for HTTP Server

Support for the Windows 32-bit IBM HTTP Server and Apache HTTP Server was added.

Support for Linux for System z® was added (Transaction tracking is not supported).

Support for Oracle HTTP server on Linux for System x was added.

Monitoring Agent for IBM Integration Bus

Support for Linux for Power Systems (Little Endian) was added.

Monitoring Agent for Microsoft .NET

Monitoring support for ODP.NET was added.

Method trace details were added for HttpWebRequest.GetResponse() method.

Monitoring Agent for Microsoft SQL Server

Tolerance support for SQL Server 2017 was added.

Support for Always On feature for the SQL Server developers edition was added.

Monitoring Agent for MySQL

Tolerance support for the information schema tables being migrated to performance schema was added.

Support was added for deprecated tables of information schema through performance schema.

Monitoring Agent for Microsoft Internet Information Services

Support was added for FTP websites monitoring.

Monitoring Agent for MongoDB

Support was added for remote monitoring.

Support for monitoring In-Memory storage engine was added.

Monitoring Agent for OpenStack

Support for OpenStack authentication V3 API was added.

Monitoring Agent for Oracle Database

Agent version is changed to 8.0.

The configuration parameter **`Oracle JDBC jar file`** was added and the configuration parameters **`Oracle Home Directory`** and **`Oracle Instant Client Installation Directory`** were removed.

Monitoring Agent for PostgreSQL

Support was added for remote monitoring.

Monitoring Agent for SAP Applications

Support for SNC communication was added.

New threshold for SAP system down was added.

Monitoring Agent for SAP NetWeaver Java Stack

The capability to restore the SAP NetWeaver Application Server instance was added.

Monitoring Agent for Tomcat

Support was added for Linux for Power Systems (Little Endian) (Resource monitoring only).

Monitoring Agent for VMware VI

Network Summary and Disk Count were added on the ESX Server overview page.

Events group widget was added on Cluster summary page.

Monitoring Agent for WebSphere Applications

Transaction tracking support for Linux for Power Systems (Little Endian) and for Linux for System z was added.

Remember: To get transaction tracking support on Linux for Power Systems (Little Endian) and Linux for System z, complete the following steps:

1. Download the agent installation image.
2. Install the WebSphere Applications agent.
3. Download the WebSphere Applications agent interim fix 2 from Fix Central.
4. Follow the readme file of the interim fix to apply the fix.

Monitoring Agent for WebSphere MQ

The MQ Service Status group widget was added to provide your MQ service details.

Support for Linux for Power Systems (Little Endian) was added.

Data collector enhancements

Liberty data collector

The managed system name (MSN) registered by the Liberty data collector is changed to reflect the host name and Liberty server name. The new MSN for this data collector is `BI:`*`servername_hostname_md5`*`:`BLP, where *md5* is the local application GUID based on MD5. The length of *servername_hostname_md5* is 25 characters.

Remember: To get this enhancement, you must download and apply the IBM Cloud Application Performance Management Liberty data collector Interim Fix 1 from Fix Central.

J2SE data collector

Transaction tracking support for J2SE applications was added.

Documentation enhancements

The information about default ports that are used by agents and data collectors is provided to facilitate you to prepare the environment. See "Default ports used by agents and data collectors" on page 85.

The information about managed system names (MSN) of Cloud APM agents is provided. Instructions about how to change the hostname string in the MSN are also provided. See "Managed system names" on page 168.

The information about running agent as a non-admin user or permissions that are required to run the agent by non-admin user is provided in the configuration topics for the following agents:

- Microsoft .NET agent
- Microsoft Active Directory agent
- Microsoft Exchange Server agent
- Skype for Business Server agent
- Microsoft SharePoint Server agent
- Microsoft SQL Server agent
- Tomcat agent

August 2017

IBM Cloud Application Performance Management, Availability Monitoring

The Availability Monitoring add-on provides enhanced synthetic monitoring of your web applications from multiple points of presence around the world. Create synthetic tests that mimic user behavior at regular intervals. Run your tests from public points of presence, or download and deploy your own custom points of presence on local or private servers. Use the Availability Monitoring dashboard to monitor application availability, performance, and alerts by using graphs, breakdown tables, and map views. Use waterfall analysis to identify when performance and availability issues occur, and find the reasons for those issues.

IBM API Connect® monitoring

Cloud APM agents and data collectors now support the monitoring of the IBM API Connect environment. You can deploy corresponding agents and data collectors to gain visibility of the health and performance of the components in your environment. Transaction tracking data is also available in addition to resource monitoring and deep-dive diagnostics data, which allows you to view topology information about your IBM API Connect environment. For more information, see "Scenario: Monitoring IBM API Connect" on page 90.

OS support

Linux for System z

Added Linux for System z support for the following monitoring agents: Linux OS, WebSphere Application, Db2, WebSphere MQ, IBM Integration Bus, Tomcat, and Response Time Monitoring.

Linux for Power Systems (Little Endian)

Added Linux for Power Systems (Little Endian) support for the following monitoring agents: Linux OS, WebSphere Application, and Db2.

Linux for System x

Added Linux on System x to support the Liberty data collector.

IBM i OS agent support

Data for the IBM i OS agent can now be displayed in the Cloud APM console. This agent is an IBM Tivoli Monitoring V6 agent and remains as a V6 agent for the V8.1.4 release. You can use the Hybrid Gateway to retrieve the agent data and send it to the Cloud APM server. As a result, you can view monitoring data and events for this agent in the Cloud APM console. For more information about the IBM i OS agent, see "Supported Tivoli Monitoring and OMEGAMON agents" on page 934..

New agents available

Monitoring Agent for OpenStack

You can use the OpenStack agent to monitor the health and performance of your OpenStack applications and view information such as information about API endpoints, SSH server connection, processes, and hypervisors.

New and enhanced data collectors available

You can use the data collectors to monitor the health and performance of the following applications on IBM Cloud, on premises, or both:

J2SE data collector

You can use the J2SE data collector to monitor the health and performance of Java applications and view diagnostics data, such as response time, throughput, request context, and method trace of requests.

Liberty data collector

The Liberty data collector monitors both the Liberty profile in IBM Cloud environment and the local Liberty profile on Linux for System x.

Node.js data collector

The Node.js data collector monitors both IBM Cloud and on-premises applications. You can view both resource and diagnostics monitoring data, such as resource utilization, throughput, and detailed information about requests and methods.

Python data collector

The Python data collector monitors IBM Cloud applications. You can view both resource and diagnostics monitoring data, such as resource utilization, throughput, and detailed information about requests and methods.

The Python agent is removed from the agent installation package in Cloud APM V8.1.4. You can use only the Python data collector to monitor your Python applications.

Ruby data collector

The Ruby data collector monitors only IBM Cloud applications. You can view both resource and diagnostics monitoring data, such as resource utilization, throughput, and detailed information about requests and methods.

Agent enhancements

Monitoring Agent for Amazon EC2

- The agent can handle null end dates for scheduled events correctly.
- Support for a forward proxy between the Amazon EC2 agent and Amazon Web Services was added.

Monitoring Agent for Citrix Virtual Desktop Infrastructure

- Monitoring of Windows events and PowerShell metrics even when the agent is installed on a Linux system was added.
- The **VDA Sessions** page, which is accessible through the **VDA Machine Details** page was added.
- The Machine Metrics widget was added to the **VDA Machine Details** page.
- The Desktop Delivery Controller (DDC) configuration was enhanced to enable the agent to handle DDC fail-over in a distributed environment.

Monitoring Agent for Db2

- Support for Linux for System z was added.

Monitoring Agent for Hadoop

- Support for monitoring the following Hadoop offerings was added: Hortonworks HDP 2.6 and Cloudera CDH 5.9, 5.10, and 5.11
- Support for monitoring the status of Hadoop services, such as are ZooKeeper, Sqoop, Hive, HDFS, YARN, Ambari Metrics, and Oozie was added.
- Support was added for monitoring a Hadoop cluster that is secured with the Kerberos SPNEGO-based authentication, which uses only MIT Kerberos V5 Key Distribution Center (KDC).

Monitoring Agent for IBM Integration Bus

Support was added for Linux for System z (Transaction tracking is not supported).

Monitoring Agent for JBoss

- The transaction tracking and deep-dive diagnostics configuration process was simplified for the JBoss agent in the Advanced Agents offering.
- Two dashboard widgets were added to the **Garbage Collection Detail** page. One widget shows the amount of heap memory that is freed since the last garbage collection, and the other widget shows the historical Eden/Survivor/Tenured (Old Gen) heap memory pool sizes.

Monitoring Agent for Linux OS

Support for Linux for Power Systems (Little Endian) was added.

Monitoring Agent for Skype for Business Server

- The Lync Usage Summary group widget was added to the Lync Server Overview dashboard to view the front-end registration status and the quality of poor calls.
- A dashboard to display the details of the Microsoft Lync Server usage was added.

Monitoring Agent for SAP NetWeaver Java Stack

The following enhancements were added to the dashboard of the SAP NetWeaver Java Stack agent:

- Data sets, group widgets, and pages were added to collect and view the transaction tracking and diagnostics data.
- Support for installing and configuring the agent on Windows 2016 systems was added.
- The Top 5 Slowest Requests by Response Time group widget is added to the SAP NW Java Instance dashboard to provide information about the top 5 requests that are made by the user to the application with the high response time.
- Diagnostic information about the requests that are displayed in the Top 5 Slowest Requests by Response Time group widget can be seen in the Request Instances page by clicking the Request. Support for displaying the diagnostic information about the requests in the Top 5 Slowest Requests by Response Time group widget was added.
- The Top 5 User Sessions by Response Time group widget is removed.

Monitoring Agent for MongoDB

Support was added for monitoring the MongoDB cluster or replication setup when the primary node fails.

Monitoring Agent for MySQL

Data sets and a configuration parameter were added to remotely monitor MySQL resources.

Monitoring Agent for NetApp Storage

- The Component page is updated to display the summarized information of clusters and Vservers.
- The NetApp Storage Instance page is updated to display information about clusters.

Monitoring Agent for Node.js

The following enhancements were added to the Node.js agent to leverage Node Application Metrics (Appmetrics):

- New dashboard and group widgets to view garbage collection details
- New dashboard and group widgets to view event loop details

Monitoring Agent for PostgreSQL

- Support for installing and configuring the agent on Windows systems.
- Support for monitoring PostgreSQL V9.6.
- The **Status Overview** page was updated so status is not critical when the buffer hit rate is zero.

Monitoring Agent for SAP HANA Database

The License Expiry Days attribute and the HANA_License_Expiry_Crit_SYS and HANA_License_Expiry_Warn_SYS thresholds were added to monitor the number of days that are remaining before license expiration.

Monitoring Agent for Tomcat

- Support was added for Linux for System z
- Data sets, dashboards, and group widgets for transaction tracking and deep-dive diagnostics were added.

Monitoring Agent for VMware VI

The ESX Server group widget in the Server Summary dashboard was updated to show the SSH status.

Monitoring Agent for WebLogic

Transaction tracking and deep-dive diagnostics were added to the agent in the Advanced Agents offering.

Monitoring Agent for WebSphere Applications

Support for Linux for System z was added (Transaction tracking is not supported).

Monitoring Agent for WebSphere MQ

Support for Linux for System z was added (Transaction tracking is not supported).

Channel and queue long-term history data is supported. After the queue manager is configured to collect channel or queue statistics data, you can configure the agent to enable channel or queue long-term history data collection. Although there are no predefined dashboards or widgets to display the collected long-term history data, you can the **Attribute Details** tab to query the collected data in your custom tables.

Response Time Monitoring Agent

- Support for the Windows 32-bit IBM HTTP Server and Apache HTTP Server is added.
- Support for configuring user tracking for applications on the **Agent Configuration** page was added.
- Support for configuring session tracking for applications on the **Agent Configuration** page was added.

Enhanced visualization

Custom Views

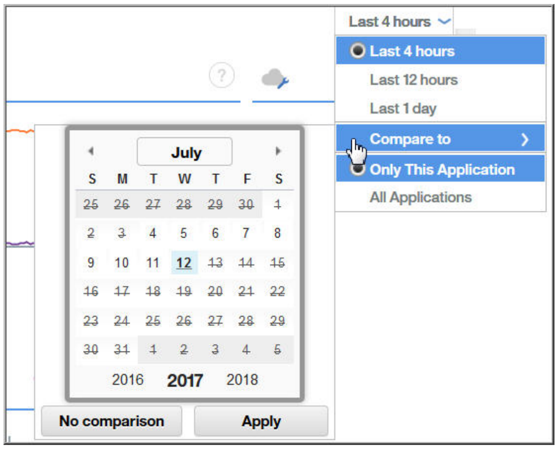
You can use the IBM Cloud Application Business Insights Universal View to create customized pages for the applications that you are monitoring. In the Custom Views tab, you can use an existing template or create customized templates for your page. You can choose from different chart and metric options to create widgets to monitor data according to your requirement.

By using the Universal View, you can create dashboards to monitor data from various agents. You can export the customized page data to a Raw Data file.

For more information, see “Custom views” on page 1086.

Calendar for comparing a previous day's data

When you are viewing line charts that show historical data, a calendar opens after you choose the time selector option to compare the time range from a previous day. The days that are unavailable for comparison are crossed out. For more information, see “Adjusting and comparing metrics over time” on page 1067.



Agent Builder enhancements

When you create an agent to monitor data from a Java Database Connectivity (JDBC) database, you can modify the enumeration values that are set for Error, Missing data, and No value to avoid overlap with legitimate values in the database.

You can set the Time Stamping Authority for JAR files in the Agent Builder **Preferences** window. If the default Time Stamping Authority signing certificate expires, by setting a new authority, you can continue to verify JAR files.

Enhanced integration

Customizable EIF Slots for events

When you have event forwarding configured, you can customize the base EIF slot message and create custom EIF slots for events sent to a receiver such as Netcool/OMNIbus. The Threshold Editor has a new **Forward EIF Event?** field and **EIF Slot Customization** button for customizing how events are mapped to forwarded events. For more information, see “Customizing an event to forward to an EIF receiver” on page 972.

Multiple Hybrid Gateways

In previous releases, you were able to install the Hybrid Gateway on a single IBM Tivoli Monitoring domain, which has one hub Tivoli Enterprise Monitoring Server. You can now install a Hybrid Gateway on multiple Tivoli Monitoring domains. The Hybrid Gateway category in the Cloud APM console **Advanced Configuration** page has been moved to its own **Hybrid Gateway Manager** page. Here you can create and edit Hybrid Gateway profiles for monitoring managed systems from multiple Tivoli Monitoring domains, one profile for each domain. For more information, see “Hybrid Gateway” on page 933.

Enhanced scalability

An increase in the maximum number of managed systems that you can monitor from Cloud APM from 4,000 managed systems to 10,000 managed systems.

Previous releases

For information about new features, capabilities, and coverage in previous releases, see the following *What's new* topics:

- “What's new: April 2017” on page 32
- “What's new: September 2016” on page 38
- What's new: April 2016

What's new: April 2017

New features, capabilities, and coverage became available in the April 2017 release of Cloud APM.

New Application Performance Management Developer Center

The APM Developer Center is a central location from which you can access resources for the APM products: blogs, videos, documentation, support, events, IBM Marketplace, and other resources. The Cloud APM console ? **Help** menu has a convenient link to the Application Performance Management Developer Center.

Product rebranding and simplification

IBM Performance Management on Cloud was rebranded to IBM Cloud Application Performance Management. The component names have also changed. For example, Cloud APM console and Cloud APM server were called the Performance Management console and Performance Management server in earlier releases.

IBM Performance Management on Cloud subscription offerings have been consolidated and renamed:

Offering name October 2016 release and earlier	Offering name March 2017 and later
Monitoring on Cloud	Cloud APM, Base
Application Performance Management Advanced on Cloud	Cloud APM, Advanced

Some product extensions have been consolidated and renamed:

Extension name October 2016 release and earlier	Extension name March 2017 and later
Base Extension Pack (Hadoop agent)	Base Extension Pack (adds the new Cassandra agent and Microsoft Office 365 agent)
Advanced Extension Pack (SAP HANA Database agent and SAP NetWeaver Java Stack agent)	Advanced Extension Pack (adds the new RabbitMQ agent)

OS support

Windows 2016 operating systems

Added support for Windows 2016 operating systems. For more information, see the Software Product Compatibility Report (SPCR) for all agents: http://ibm.biz/agents-pm-systemreqs
Find your operating system in the Windows section of the report and click the component icon for a list of supported agents.

New extension pack available

IBM Cloud Application Performance Management z Systems® Extension Pack

The z Systems Extension Pack enables support for your IBM OMEGAMON® agents in your Cloud APM offering. OMEGAMON agent data is sent to the Cloud APM server by the Hybrid Gateway. The Hybrid Gateway retrieves the OMEGAMON agent data and events from the IBM Tivoli Monitoring infrastructure that the OMEGAMON agents are connected to. As a result, you can view monitoring data and events for your OMEGAMON agents in the Cloud APM console.

The Cloud APM z Systems Extension Pack is available if you have either of the Cloud APM offerings.

To integrate this extension pack with Cloud APM, complete the steps in "Integrating with OMEGAMON" on page 944.

New agents and data collectors available

Monitoring Agent for Cassandra

You can use the Cassandra agent to monitor the health and performance of the Cassandra cluster resources, such as the nodes, keyspaces, and column families.

Monitoring Agent for Microsoft Office 365

You can use the Microsoft Office 365 agent to monitor the health and performance of the Office 365 resources, such as the Office 365 subscribed services, Office 365 portal, mailbox users, SharePoint sites, and OneDrive storage.

Monitoring Agent for NetApp Storage

You can use the NetApp Storage agent to monitor the health, availability, and performance of the NetApp storage systems by using the NetApp OnCommand Unified Manager (OCUM). The monitoring agent performs the following tasks:

- Identifies poorly performing storage system objects
- Performs discovery and monitoring by using the OCUM server at the focal point

Monitoring Agent for RabbitMQ

You can use the RabbitMQ agent to monitor the health and performance of the RabbitMQ cluster resources, such as the nodes, queues, and channels of the cluster.

Data collectors for Bluemix® applications

You can use the data collectors for Bluemix applications to monitor the health and performance of the following types of your applications on Bluemix:

- Liberty applications
- Node.js applications
- Python applications
- Ruby applications

You can view both resource and diagnostics monitoring data, such as resource utilization, throughput, and detailed information about requests and methods.

Monitoring Agent for Siebel

You can use the Siebel agent to monitor the health and performance of Siebel resources including Siebel statistics, user sessions, components, tasks, application server, Siebel Gateway Name Server, process CPU and memory usage, and log event monitoring.

Agent enhancements

Monitoring Agent for Amazon EC2

The following enhancements are added to the Amazon EC2 agent:

- Replace Instance ID with tag name when a tag name is available
- Allow data to be filtered and grouped based on tag name

Monitoring Agent for Db2

The following enhancements are added in the Hadoop agent:

- Linux on Power Little Endian (pLinux LE) is supported
- Added a script file to grant privileges to a Db2 user to view data for all the attributes of the Db2 agent for a monitored instance

Monitoring Agent for Hadoop

The following enhancements are added in the Hadoop agent:

- Added support to install and configure the agent on Windows 2016 and AIX 7.2 systems

- Added support for monitoring the following Hadoop offerings: Hortonworks HDP 2.5, Cloudera CDH 5.6, 5.7, and 5.8, and IBM BigInsights 4.2
- Added the test connection button to verify connection to the Hadoop daemons that you specify when you configure the agent
- Improved the agent configuration process to reduce the configuration time and complexity. The configuration is simplified because the following prerequisite and configuration tasks are not required:
  - Installing the plug-in on each node of the Hadoop cluster
  - Configuring and updating the `hadoop-metrics2.properties` file
  - Restarting the Hadoop daemons after configuring the `hadoop-metrics2.properties` file
  - Configuring all the DataNodes and NodeManagers in the cluster
  - Restarting the agent when additional nodes are added to the cluster

Monitoring Agent for JBoss

The following enhancements are added to the JBoss agent:

- Added transaction tracking and deep dive monitoring in the Advanced Agents offering
- Added a dashboard page to monitor datasource metrics
- Added support for monitoring the following JBoss offerings: WildFly 8.x/9.x/10.x, JBoss EAP 7.x, JBoss AS 7.x
- Added support for running the agent on the Windows operating system

Monitoring Agent for Linux KVM

The following enhancements are added to the dashboard of the Linux KVM agent:

- Updated the Hosts group widget in the Hosts, Clusters, and Storage page to display the Max Scheduling Memory (GB) and the Live Snapshot KPIs
- Added the Storage Details page to display details about the disks and disk snapshots in the storage pool
- Add the Network Transmitted/Received Data (GB) group widget in the Host Detail page to display historical information of the total data (in GB) that is transmitted and received over the network

Monitoring Agent for Linux OS

The following enhancement is added to the Linux OS agent:

- Linux on Power Little Endian (pLinux LE) is supported

Monitoring Agent for Microsoft Exchange Server

The following enhancements are added in the dashboard of the Microsoft Exchange Server agent:

- Added the inbound time and outbound time attributes in the Reachability data set
- Added pages and group widgets to display reachability details
- Added an eventing threshold for reachability
- Added support to install and configure the agent on Exchange Server 2016 and Windows Server 2016 system

Monitoring Agent for Microsoft Internet Information Services

The following enhancement is added to the Microsoft IIS agent :

- Added support to install and configure the agent on the Microsoft Windows Server 2016 system

Monitoring Agent for Microsoft Active Directory

The following enhancements are added in the Microsoft Active Directory agent:

- Added the group widgets and pages to display the details of Group Policy Object, Netlogon, Local Security Authority, and LDAP details
- Added the following data sets that you can view in the **Attribute details** tab:

  - Services Data Set
  - Replication
  - File Replication Service
  - Moved or Deleted Org. Unit
  - LDAP Attributes
  - Security Accounts Manager
  - DFS
  - Address Book
  - Event Log
  - Password Setting Objects
- Added the data sets for ADFS, ADFS Proxy, and Asynchronous Thread Queue
- Added the group widgets and pages to display details of ADFS and ADFS Proxy
- Added support to install and configure the agent on Windows Server 2016 systems

Monitoring Agent for Microsoft .NET

The following enhancements are added in the dashboard of the Microsoft .NET agent:

- Updated the MS .NET Status group widget on the Component page to display the response times of database calls, status of .NET processes with high thread count, and Just in Time (JIT) compilation failures
- Added data sets, pages, and group widgets to show JIT compilation details, database call details, GC handles and pinned objects collection for a selected .NET process, thread contention rate, and thread queue length
- Added eventing thresholds for JIT failures, .NET request failures, slow commands, garbage collection, and the active threads in .NET processes

Monitoring Agent for Microsoft SQL Server

The following enhancements are added to the dashboard of the Microsoft SQL Server agent:

- Added the Expensive Queries group widget in the Server Performance - Detail page to view the top 10 cached query plans according to the performance statistics of the Microsoft SQL Server
- Added support for monitoring the Microsoft SQL Server 2016
- Added support to install and configure the Microsoft SQL Server agent on the Microsoft Windows Server 2016 system
- Added the new *COLL_ERRORLOG_RECYCLE_WAIT* environment variable to set the time interval (in seconds) for which the agent waits before collecting data of the MS SQL Error Event Detail attribute group

Monitoring Agent for MongoDB

The following enhancements are added to the dashboard of the MongoDB agent:

- Updated the Component page to display the number of MongoDB instances and their status
- Added pages to display details of the MMAPv1 and the WiredTiger storage engines
- Added the Input Output Information page to display cursor details and historical data for the queued operations, active connections, data flow, and the data access of the selected host
- Added pages to display details of the locks of version 2.x and version 3.x, or later
- Added the Replication Details page to display details of the replication member, oplog, and historical data of the replication lag and the space that is used by the oplog

Monitoring Agent for Node.js

The following enhancements are added to the Node.js agent to leverage Node Application Metrics (Appmetrics):

- Added new dashboard and group widgets to view garbage collection details

- Added new dashboard and group widgets to view event loop details

Monitoring Agent for PostgreSQL

The following enhancements are added to the PostgreSQL agent:

- Added support to install and configure the agent on Windows systems
- Added support for monitoring the PostgreSQL V9.6
- Updated the Status Overview page so that status is not critical when the buffer hit rate is zero

Monitoring Agent for SAP NetWeaver Java Stack

The following enhancements are added to the SAP NetWeaver Java Stack agent:

- Added data sets, group widgets, and pages to collect and view the transaction tracking and diagnostics data
- Added support to install and configure the agent on Windows 2016 systems

Monitoring Agent for Synthetic Playback

The following enhancement is added to the Synthetic Playback agent:

- The Synthetic Playback agent includes a new filtering feature for synthetic transactions. In the Synthetic Script Manager, configure blacklists and whitelists for your synthetic transactions that exclude or include requests to specified URLs and domains. Use blacklists and whitelists to filter out or include dependencies that affect the response times for your application, such as third-party metrics.

Monitoring Agent for Tomcat

The following enhancement is added to the Tomcat agent:

- Added support to install and configure the Tomcat agent on Windows and SUSE Linux Enterprise 12 systems

Monitoring Agent for WebSphere Applications

The following enhancements are added to the WebSphere Applications agent:

- Linux on Power Little Endian (pLinux LE) is supported. (Transaction tracking is not supported on pLinux LE systems.)
- Added support for IBM WebSphere Application Server traditional V9.
- Added the Memory Analysis dashboard to help you diagnose possible memory leaks by checking the heap usage information for each heap dump. The diagnostics mode must be enabled for this dashboard to contain data.
- Added support to use the Application Health Status data set to create event thresholds for application status monitoring. The data collection for this usage is disabled by default. You must modify the data collector properties file to enable it before you create event thresholds.
- Simplified the manual configuration of data collector. For WebSphere Applications Server, you only need to add some JVM arguments and variables for the application server on the WebSphere administrative console. For Liberty, you only need to modify three files for the server.

Response Time Monitoring Agent

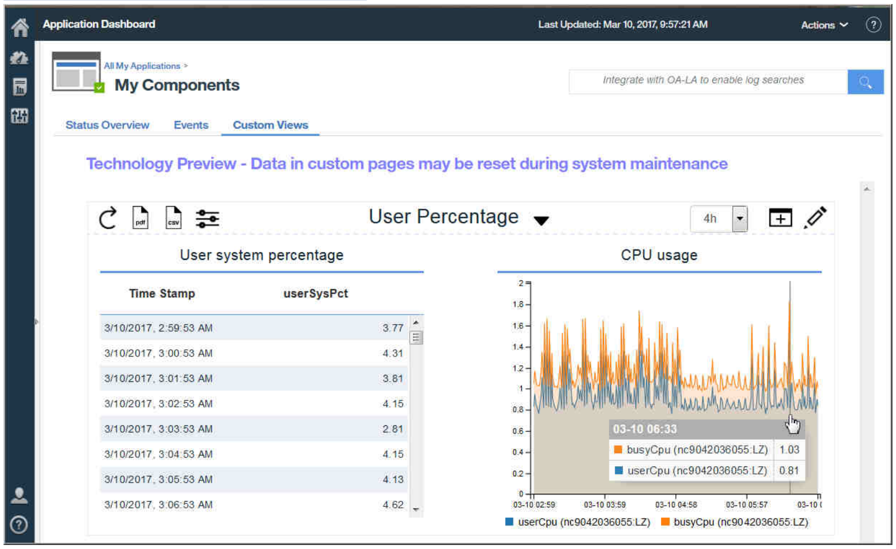
The following enhancements are added to the Response Time Monitoring agent:

- Added support for configuring user tracking for applications in the **Agent Configuration** page.
- Added support for configuring session tracking for applications in the **Agent Configuration** page.

Cloud APM console enhancements

Various improvements were made to the agent installation and configuration interfaces, as well as the following console enhancements:

- Technology preview: A new **Custom Views** tab is available for your Application Dashboard pages. You can create a variety of views for reporting metrics from a managed resource and apply functions such as average and count. After you open a saved page, you can refresh the page with data from a different resource, and download the page metrics as a PDF or CSV file. For more information, see "Custom views" on page 1086.



Technology Preview - Data in custom pages may be reset during system maintenance

User Percentage

User system percentage

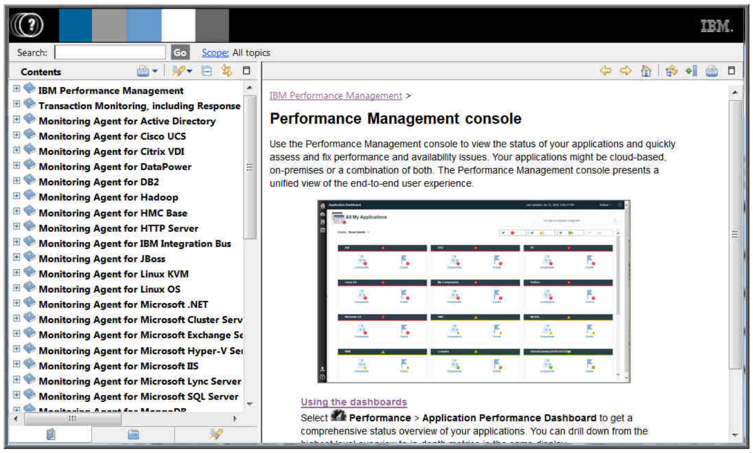
Time Stamp	userSysPct
3/10/2017, 2:59:53 AM	3.77
3/10/2017, 3:00:53 AM	4.31
3/10/2017, 3:01:53 AM	3.81
3/10/2017, 3:02:53 AM	4.15
3/10/2017, 3:03:53 AM	2.81
3/10/2017, 3:04:53 AM	4.15
3/10/2017, 3:05:53 AM	4.13
3/10/2017, 3:06:53 AM	4.62

CPU usage
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
03-10 02:59
03-10 03:59
03-10 04:58
03-10 05:57
03-10 0
03-10 06:33
busyCpu (nc9042036055:LZ) 1.03
userCpu (nc9042036055:LZ) 0.81
userCpu (nc9042036055:LZ)
busyCpu (nc9042036055:LZ)

If you do not see **Custom Views** in your Cloud APM subscription and you want to try out this new feature, please open a service request with IBM Support to enable the **Custom Views** technology preview. Be aware that your custom dashboard pages and the historical data that populates them are not saved during system maintenance.

- When you open the Cloud APM console help system, notice that it is hosted by IBM Knowledge Center. You have the ☰ **Hide table of contents** tool, search and print capabilities, and links to support information and feedback options.

Before



IBM Performance Management
Transaction Monitoring, including Response
Monitoring Agent for Active Directory
Monitoring Agent for Cisco UCS
Monitoring Agent for Citrix VDI
Monitoring Agent for DataPower
Monitoring Agent for DB2
Monitoring Agent for Hadoop
Monitoring Agent for HMC Base
Monitoring Agent for HTTP Server
Monitoring Agent for IBM Integration Bus
Monitoring Agent for JBoss
Monitoring Agent for Linux KVM
Monitoring Agent for Linux OS
Monitoring Agent for Microsoft .NET
Monitoring Agent for Microsoft Cluster Serv
Monitoring Agent for Microsoft Exchange Se
Monitoring Agent for Microsoft Hyper-V Se
Monitoring Agent for Microsoft IIS
Monitoring Agent for Microsoft Lync Server
Monitoring Agent for Microsoft SQL Server

IBM Performance Management >

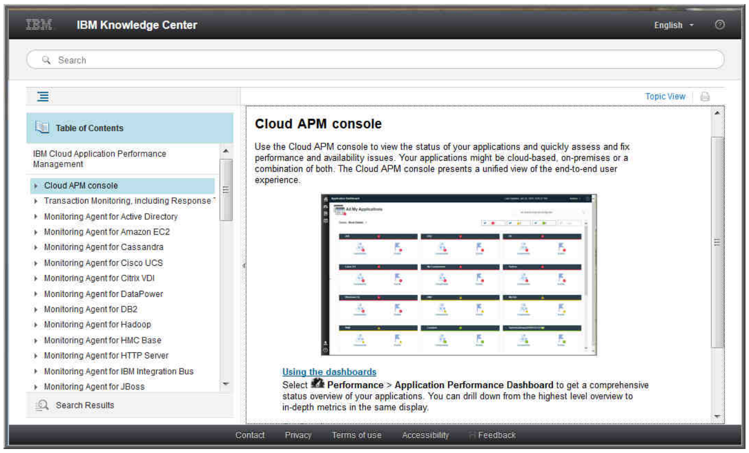
Performance Management console

Use the Performance Management console to view the status of your applications and quickly assess and fix performance and availability issues. Your applications might be cloud-based, on-premises or a combination of both. The Performance Management console presents a unified view of the end-to-end user experience.

Using the dashboards

Select **Performance** > **Application Performance Dashboard** to get a comprehensive status overview of your applications. You can drill down from the

After



IBM Cloud Application Performance Management
Cloud APM console
Transaction Monitoring, including Response
Monitoring Agent for Active Directory
Monitoring Agent for Amazon EC2
Monitoring Agent for Cassandra
Monitoring Agent for Cisco UCS
Monitoring Agent for Citrix VDI
Monitoring Agent for DataPower
Monitoring Agent for DB2
Monitoring Agent for Hadoop
Monitoring Agent for HMC Base
Monitoring Agent for HTTP Server
Monitoring Agent for IBM Integration Bus
Monitoring Agent for JBoss

Cloud APM console

Use the Cloud APM console to view the status of your applications and quickly assess and fix performance and availability issues. Your applications might be cloud-based, on-premises or a combination of both. The Cloud APM console presents a unified view of the end-to-end user experience.

Using the dashboards

Select **Performance** > **Application Performance Dashboard** to get a comprehensive status overview of your applications. You can drill down from the highest level overview to in-depth metrics in the same display.

Agent Builder enhancements

Support is improved for building Cloud APM summary dashboards for Agent Builder agents. You must use single-row data sets to provide data for summary dashboards. You can provide such data sets from entire log files and from any sets of data that can be filtered to a single row.

What's new: September 2016

New features, capabilities, and coverage became available in the September 2016 release of Performance Management on Cloud.

New agents available

Monitoring Agent for Amazon EC2

You can use the Amazon EC2 agent to monitor the health, availability, and performance of your Amazon Elastic Compute Cloud (EC2) Instance resources. You can monitor the following resources:

- CPU utilization
- Elastic Block Store (EBS) utilization
- Network utilization
- Amazon Web Services (AWS) maintenance updates
- Disk performance

This agent is in the Infrastructure Extension Pack and is available for the following offerings: IBM Monitoring, IBM Application Performance Management, and IBM Application Performance Management Advanced.

Monitoring Agent for SAP NetWeaver Java Stack

You can use the SAP NetWeaver Java Stack agent to monitor the health, availability, and performance of your SAP NetWeaver Java Stack Cluster and Instance resources. You can use the agent to monitor the cluster resources, such as heap dumps, JVM Instance, response time of the user sessions, transaction details, system information, and license details. You can use the agent to monitor the instance resources, such as CPU utilization, disk utilization, memory utilization, database collection, garbage collection, heap dumps, failed application, web container, and session information. This agent is in the Advanced Extension Pack and available if you have one of the following offerings: IBM Application Performance Management and IBM Application Performance Management Advanced.

Agent enhancements

Monitoring Agent for Citrix Virtual Desktop Infrastructure

Added the ability to retrieve Windows Event Log Events for Virtual Delivery Agent (VDA) and Desktop Delivery Controller (DDC) machines.

Monitoring Agent for Linux KVM

Dashboards are available for the agent to monitor the deployment of your Linux Kernel-based virtual machines. The dashboards provide the following monitoring capabilities:

- The summary dashboard shows the overall status of the hosts based on the CPU and memory utilization of your Linux Kernel-based virtual machines environment or application.
- The Host Detail dashboard shows details about the selected host.
- The Hosts, Clusters, and Storage dashboard shows details about the monitored virtual machines.
- The Virtual Machine Details dashboard shows details about the virtual machine that you select on the Host Detail page.

Monitoring Agent for Linux OS

Docker V1.8.0 or later is supported. New attribute groups and widgets were added to enable the Linux OS agent to deliver docker monitoring capabilities.

Monitoring Agent for Oracle Database

The Oracle Database agent dashboard includes the following new features on the Instance Details page:

- A table that displays information about the lock contention on the selected instance.
- A table that displays information about the Oracle Real Application Clusters GCS and GES.
- A table that displays details of the Automatic Storage Management (ASM) disk groups that are attached to the selected instance.
- A view that shows detailed information per tablespace, which is visible if you click **Bottom 5 Free Table Space**.
- A table that displays the historical details of the foreground and background processes that are attached to the selected instance. You can click the entity in the table and view a detailed table of all processes for that instance.
- A table that displays the Top 5 Worst SQL queries (by run time) on the selected instance. You can click in the table and view a detailed table of the top 50 worst SQL queries for that instance.

Monitoring Agent for Synthetic Playback

The Synthetic Playback agent includes a new security feature. You can prevent passwords that are stored in synthetic scripts from displaying in the Synthetic Script Manager.

Monitoring Agent for VMware VI

With the addition of the agent decoupling feature, you can view and select the agent node and its subnodes in the same view.

When you select the VMware Virtual Infrastructure component in the Select Component window, the Component Editor displays a tree structure of the agent node with all its subnodes.

- If you expand the tree and select the agent node, all subnodes are automatically selected. You can also expand the tree and individually select the subnodes that you want to monitor.
- If you select the agent node when the tree is collapsed, all subnodes are automatically excluded.

When you select the ESX Server component in the Select Component window, along with subnodes, stand-alone ESX Servers are also displayed in the Component Editor. With the subnodes, you can select stand-alone ESX Servers for monitoring.

After the application is created, the APM UI dashboard displays a tree structure of the agent instance as parent and its nodes as children.

Response Time Monitoring Agent

You can customize the locations that are applied to specific IP addresses or address ranges in the End User Transaction dashboards for your particular environment. Use the **Geolocation** tab in the **Agent Configuration** to customize location values.

Cloud APM console enhancements

- Various improvements were made to the agent installation and configuration interfaces.
- A **Dashboard Log** option was added to the **Actions** menu to review the list of agent dashboards that were updated since the last server restart. For more information, see “All My Applications - Application Performance Dashboard” on page 1055.
- The Application Performance Dashboard page for the selected application is streamlined for improved viewing. A count of critical and warning severity events is displayed on the Events tab title and replaces the **Event Severity Summary** bar chart. For applications with topology views enabled, the **Aggregate Application Topology** view has a toggle button for switching to the Current Component Status bar chart. For more information, see “Status Overview” on page 1058.
- In earlier releases, the Application Performance Dashboard **Attribute Details** tab was available only for component instances. The **Attribute Details** tab is available for creating historical tables of Response Time Monitoring agent and Synthetic Playback agent transaction instances. For visually impaired users, the ability to create historical tables provides an alternative to line charts, which

assistive technologies such as screen-reader software cannot interpret. For more information, see "Viewing and managing custom charts and tables" on page 1068.

API

You can use APIs to create scripts for automating the onboarding of your Performance Management environment. For more information, see "Exploring the APIs" on page 1049.

What's new: April 2016

New features, capabilities, and coverage became available in the April 2016 release of Performance Management on Cloud.

IBM Marketplace

IBM Performance Management on Cloud offerings are available from IBM Marketplace. Sign up for a free trial or subscription account. For more information, see "Downloading your agents and data collectors" on page 105.

New agents available

Monitoring Agent for Citrix Virtual Desktop Infrastructure

You can use the Citrix VDI agent to monitor the health, availability, and performance of Citrix XenDesktop or XenApp resources such as sites, machines, applications, desktops, sessions, and users. This agent is in the Infrastructure Extension Pack and is available for the following offerings: IBM Monitoring, IBM Application Performance Management, and IBM Application Performance Management Advanced.

Monitoring Agent for Skype for Business Server

You can use the Skype for Business Server agent to monitor the health, availability, and performance of the Microsoft Lync Server resources such as database, mediation server, synthetic transactions, instant messaging, CDR service write operations, and SIP peers.

Monitoring Agent for WebLogic

You can use the WebLogic agent to monitor the health, availability, and performance of WebLogic server resources such as Java virtual machines (JVMs), Java messaging service (JMS), and Java Database Connectivity (JDBC).

Integration enhancements

Agent coexistence

Agent coexistence is supported. You can install IBM Performance Management agents on the same computer where IBM Tivoli Monitoring agents are installed. However, both agents cannot be installed in the same directory. See "Cloud APM agent and Tivoli Monitoring agent coexistence" on page 930.

IBM Alert Notification

Alert Notification includes a mobile app that offers a subset of Alert Notification functions on iOS and Android devices.

IBM integration stack monitoring

You can monitor the IBM integration stack to see transaction tracking information for the IBM MQ, IBM Integration Bus, and DataPower appliance middleware products and the services they expose and troubleshoot if any problems arise. See "Scenario: Monitoring the IBM integration stack" on page 99.

Agent enhancements

Monitoring Agent for Db2

Commands were added for granting privileges to the default user (for Windows systems) and instance owner user (for Linux and AIX systems) for viewing the data for some of the Db2 agent attributes.

Monitoring Agent for Hadoop

The Hadoop agent is supported on Linux, Windows, and AIX operating systems.

Monitoring Agent for HMC Base

Monitoring capabilities are provided for virtual I/O and for hardware events.

Monitoring Agent for IBM Integration Bus

The library path of the latest version of IBM MQ (WebSphere MQ) can be automatically discovered during agent configuration on Linux and AIX systems.

Monitoring Agent for Microsoft Cluster Server

The Microsoft Cluster Server agent gets automatically configured after it is installed.

Monitoring Agent for Microsoft Exchange Server

Some additional services were added in the **Exchange Services** tab of the agent configuration window for determining the Exchange Server status.

Monitoring Agent for Microsoft Hyper-V Server

Removed the agent configuration panel. The agent configuration is not required.

Monitoring Agent for SAP HANA Database

The Cache Information Details group widget was added in the **SAP HANA Database Details** dashboard to provide information about the percentage of used memory, percentage of available memory, and hit ratio of the cache for the monitored database.

Monitoring Agent for Synthetic Playback

The Synthetic Playback agent includes the following features:

- You can install and configure the Synthetic Playback agent to monitor the performance and availability of private, internal-facing applications on the Application Performance Dashboard, in addition to public, external-facing applications.
- Use the Synthetic Script Manager to generate a simple script to test the availability and performance of your applications.
- Configure simultaneous or staggered playback of synthetic transactions at different locations.
- Monitor your monthly playback usage in the Synthetic Script Manager.
- View HTTP metrics and availability ratios in Synthetic Playback agent reports.
- View two new reports: Trend of Transactions and Trend of Subtransactions.
- Organize your synthetic transactions into a resource group and apply thresholds to all transactions in that resource group.
- View synthetic transaction data in the **My Transactions** window in the Application Performance Dashboard without needing to create an application that contains associated synthetic transactions.
- Download synthetic scripts from Synthetic Script Manager.

Monitoring Agent for VMware VI

The VMware VI agent dashboard is enhanced to include the following new features:

- The number of triggered alarms in the critical or warning state are also displayed on the **Component** page.
- A new table on the **Cluster Summary** page provides information about the proactive and failure alarms. You can click the triggered entity in the table and view the detail page of that triggered entity.
- A new table on the **Cluster Detail** page displays details of the ESX servers that belong to the selected cluster. You can click the ESX server and view the detail page of that ESX server.
- The Datastore table on the **Cluster Detail** page shows the over-commitment metric of the datastore.
- The Virtual Machines table on VM Detail page displays more performance metrics such as memory size, NICs, and disks. You can click these metrics and view their detail pages.
- New widgets and pages are added to display important performance metrics of the memory, disks, and network for the selected virtual machine.
- A new table on the **ESX Server Detail** page displays the network performance of the server network.

- The Datastore table on the **VM Detail** page and the **ESX Server Detail** page displays the latency metric of the datastore.
- A new table on the **Datastore Detail** page displays information about the virtual machine that is associated with the datastore. You can click the virtual machine in the table and view the **Virtual Machine Detail** page.
- The title of the % Memory (History) chart on the **VM detail** page was changed to Guest Memory(History).

Monitoring Agent for WebSphere Applications

The In-flight Requests Summary dashboard provides the capability to identify the request instances that are currently slow or hung. You can perform a soft cancel operation on an in-flight request by selecting the request and then clicking **Cancel Thread** in the In-flight Request widget on this dashboard.

All predefined eventing thresholds were refined to provide a better user experience. The enhancements and updates involve the condition that triggers an alert, the sampling interval, and the severity of the threshold.

The user interface of the Monitoring Agent for WebSphere Applications is accessible to users with physical disabilities.

The configuration process was refined based on customer feedback and technical review to provide a better user experience.

Monitoring Agent for WebSphere MQ

Some changes were made to the predefined eventing thresholds:

- All predefined thresholds have a prefix of MQ_ instead of MQSeries_ as in previous versions.
- Two thresholds, MQ_Channel_Initiator_Crit and MQ_Queue_Manager_Crit, were added to trigger critical alerts for the channel initiator server status and queue manager status.
- The trigger condition of the MQ_Queue_Depth_High event was changed from static 80% to the high depth value of the queue.

The name of the **Queue not being Read - Top 5** widget was changed to **Queue in Use not being Read - Top 5**. This widget provides a list of top five queues that have messages and are connected by one or more applications to put messages on the queue, but are not being read by any application.

The library path of the latest version of IBM MQ (WebSphere MQ) can be automatically discovered during agent configuration. You can keep the `WMQLIBPATH` parameter empty in the silent response file or accept the default value when you configure the agent interactively.

OS agents

The OS agents contain a new functionality to monitor application log files. The functionality includes the capability to configure log file monitoring based on regular expressions.

For compatibility, the OS agent consumes the following information and formats:

- Configuration information and the format file that was used by the IBM Tivoli Monitoring 6.x Log File Agent
- Configuration information and format strings that were used by the Tivoli Event Console Log File Adapter

These format strings allow the agent to filter the log data according to patterns in the format file and submit only the relevant data to an event consumer. The OS Agent sends data to the Performance Management server or through the Event Integration Facility (EIF) to any EIF receiver, such as the OMNIbus EIF probe.

Response Time Monitoring Agent

End User Transactions dashboards include user and device information, which was previously displayed in the Authenticated Users and Mobile Devices Users dashboards in the Users group. User, session, and device information are sorted by location (country, state, and city) based on the

IP address of the user. Use the new and updated dashboards to understand user volumes and whether issues are isolated to specific sets of users.

Customize the locations that are applied to specific IP addresses or address ranges in the End User Transaction dashboards for your particular environment. Use the **Geolocation** tab in the **Agent Configuration** to customize location values.

Transaction Tracking

The Transaction Summary page includes a Service Dependencies topology, which shows the selected resource node, such as an IBM Integration Bus, and the services that it depends on. The Transaction Details page includes a Transaction Dependencies topology that shows a transaction node for each component instance, and an uninstrumented node for each dependent service at the transaction level, for example, IBM Integration Bus and its service transactions. The Transaction Details page also highlights the users of the selected application who are experiencing the slowest response times, and the hosts with the highest volume of transactions.

General agent enhancements

The following general agent installation and configuration enhancements were made:

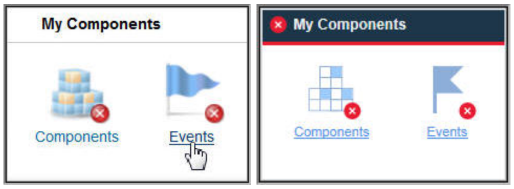
- The agent installation script performs a permissions check before the installation starts. If you do not have adequate permission, a message is displayed.
- The agent status command checks the status between the agent and the Performance Management console.
- Agents that are supported on Windows systems have a GUI utility that you can use to perform agent configuration and check the connection status.
- You can use a new command to remove an agent instance without uninstalling the agent.

Performance Management server enhancements

Performance Management user authentication is managed through an IBMid OpenID Connect provider.

Performance Management console enhancements

- The Performance Management console appearance was updated to align with the IBM Bluemix user interface. For example, view the differences between a summary box in the **All My Applications** dashboard from V8.1.2 and now:



- A new option was added to the Advanced Configuration page so advanced users can easily enable or disable all predefined thresholds across all system groups. See “Background information” on page 958.
- A new option was added to the Advanced Configuration page to control the Application Performance Dashboard automatic refresh rate. For more information, see “UI Integration” on page 1050.
- Various improvements were made to the agent installation and configuration interfaces.
- Improvements to the accessibility of the Performance Management console. For information about the accessibility features of the user interface, see “Accessibility features” on page 1473.

API

You can use APIs to create scripts for automating the onboarding of your Performance Management environment. For more information, see “Exploring the APIs” on page 1049.

Agent Builder enhancement

Agent Builder includes enhanced data set filtering. You can use filtering to create data sets that return a single row based on multi-row data sets including the Availability data set. Use this feature to provide information in summary dashboards.

Chapter 2. PDF documentation

The PDF documents are available for topics in this IBM Knowledge Center collection and for agent references.

IBM Knowledge Center in PDF format

In addition to this User's Guide, you can download the IBM Agent Builder User's Guide.

Agent Reference PDFs

The References provide information about dashboards, eventing thresholds, and data sets. Data sets contain attributes, which are the metrics that are reported by the agent and that make up the key performance indicators (KPIs). For the Reference PDF for each agent, see Agent metrics/Reference PDFs.

Chapter 3. Product overview

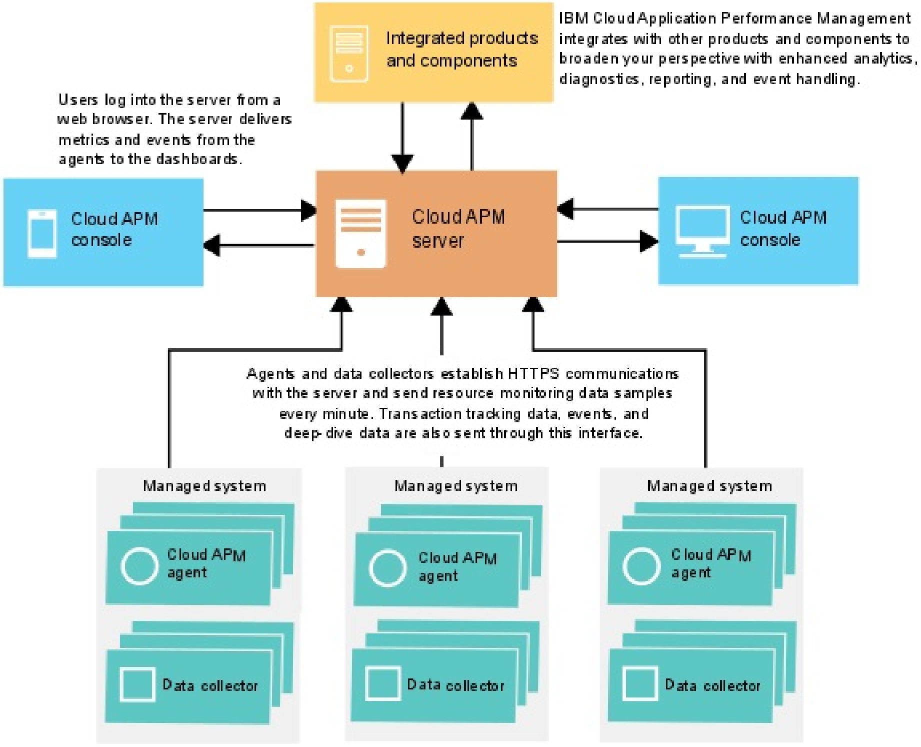
IBM Cloud Application Performance Management (Cloud APM) is a comprehensive solution that helps you manage the performance and availability of applications that are deployed on premises (private), in a public cloud, or as a hybrid combination. This solution provides you with visibility, control, and automation of your applications, ensuring optimal performance and efficient use of resources.

By using this solution, you manage your data center, cloud infrastructure, and workloads with cognitive intelligence. You can reduce and prevent outages and slowdowns around the clock in a hybrid application world as Cloud APM assists you in moving from identifying performance issues to isolating where the problem is occurring and diagnosing issues before your business is impacted.

Use the key features, which vary by offering, to work with data that is collected by the Cloud APM agents and data collectors. More features are available through integration with other products and components.

Architecture overview

IBM Cloud Application Performance Management uses *agents* and *data collectors* to collect data on the monitored hosts. Agents and data collectors pass the data to the Cloud APM server, which collates it into the Cloud APM console. The Cloud APM server is hosted in the IBM cloud.


Integrated products and components
IBM Cloud Application Performance Management integrates with other products and components to broaden your perspective with enhanced analytics, diagnostics, reporting, and event handling.
Users log into the server from a web browser. The server delivers metrics and events from the agents to the dashboards.
Cloud APM console
Cloud APM server
Cloud APM console
Agents and data collectors establish HTTPS communications with the server and send resource monitoring data samples every minute. Transaction tracking data, events, and deep-dive data are also sent through this interface.
Managed system
Cloud APM agent
Data collector
Managed system
Cloud APM agent
Data collector
Managed system
Cloud APM agent
Data collector

Data collection

Agents and data collectors monitor systems, subsystems, or applications and collect data. An agent or a data collector interacts with a single resource (for example, a system or application) and, in most cases, is on the same computer or virtual machine where the system or application is running. For example, the Linux OS agent collects performance indicators for the operating system on the Linux host and the WebSphere Applications agent monitors the performance indicators of WebSphere application servers. Also, some agents track transactions between different resources.

You can set up thresholds on key performance indicators (KPIs). If an indicator changes to go over or under the threshold, the agent or data collector generates an alert, which the server processes. You can also configure forwarding of events to a target such as the Netcool/OMNIbus Probe for Tivoli EIF or an SMTP server and use Alert Notification to configure email notifications for events.

Agents and data collectors are preconfigured to communicate with the Cloud APM server.

Communication between the server and agents or data collectors

The agents and data collectors on every monitored host establish HTTPS communication with the Cloud APM server, which is in the IBM cloud. The agent or data collector is the client side of the connection.

Agents and data collectors require internet connectivity to send data to the server and, if they cannot send data directly over the internet, a forward proxy might be required. For more information, see "Network connectivity" on page 159.

Data stored by the server

Agents and data collectors push data to the Cloud APM server at intervals ranging from 1 minute to 8 minutes, depending on the type of data. The server stores all values that are sent by the agents and data collectors for 8 days by default. Summarized transaction data is stored for longer periods.

Saved monitoring data is called *historical* data. The server uses historical data to display tables and graphs that you can use to analyze the trends in your environment.

Historical reports are also available for certain agents. For more information, see "Reports" on page 1097.

Scalability

You can monitor up to 10,000 managed systems from Cloud APM. A managed system is a single operating system, subsystem, or application in your enterprise that an agent is monitoring.

Cloud APM supports between 150 and 400 monitored user transactions per second.

Integration

IBM Cloud Application Performance Management integrates with other products and components when they are configured for communication with the Cloud APM server.

Products that can be integrated include IBM Control Desk, Netcool/OMNIbus, Tivoli Monitoring, OMEGAMON, Operations Analytics - Log Analysis, Operations Analytics - Predictive Insights, IBM Alert Notification, and IBM Cloud.

Agent Builder is a component that can be used to create custom agents.

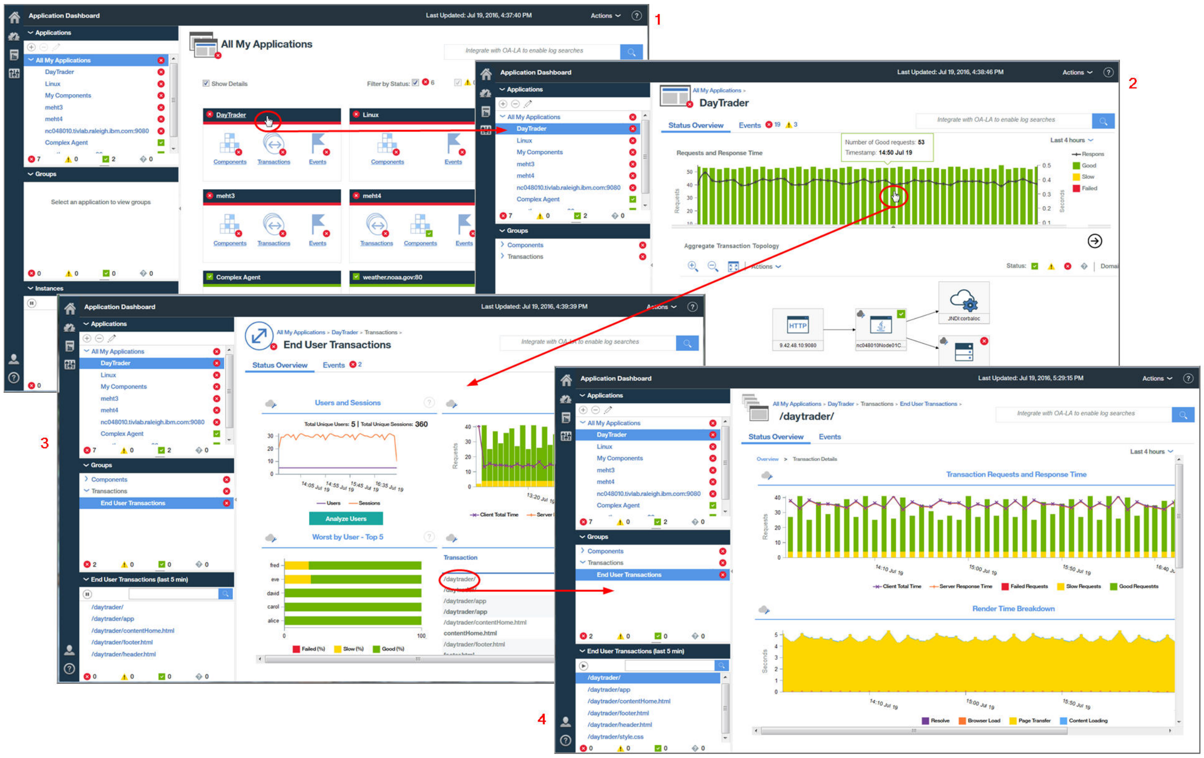
User interface

The Cloud APM console is the user interface for Cloud APM. This unified user interface provides a single view across hybrid applications. You use the console to view the status of your applications and quickly assess and fix performance and availability issues.

The dashboards in the console simplify problem identification so you can isolate bottlenecks that affect application performance. With simple dashboard navigation, you move from a view of application status to code level detail. You have visibility into source code problems at the exact moment of an issue. You can search and diagnose problems by using integrated search analytics.

The Application Performance Dashboard navigator in the console is hierarchical, giving a status overview of your applications, the health of their components, and the quality of the user experience. For more details about your monitored resource, you can click a navigator item or a link in the dashboard views. Consider, for example, that your application has a slow response time. The issue is revealed in the dashboard. Starting from your dashboard, you can follow the problem to the source by clicking links to discover the cause: high CPU usage on a system due to an out-of-control process.

For more information about using the dashboards in the Cloud APM console, see Chapter 10, "Using the dashboards," on page 1055.



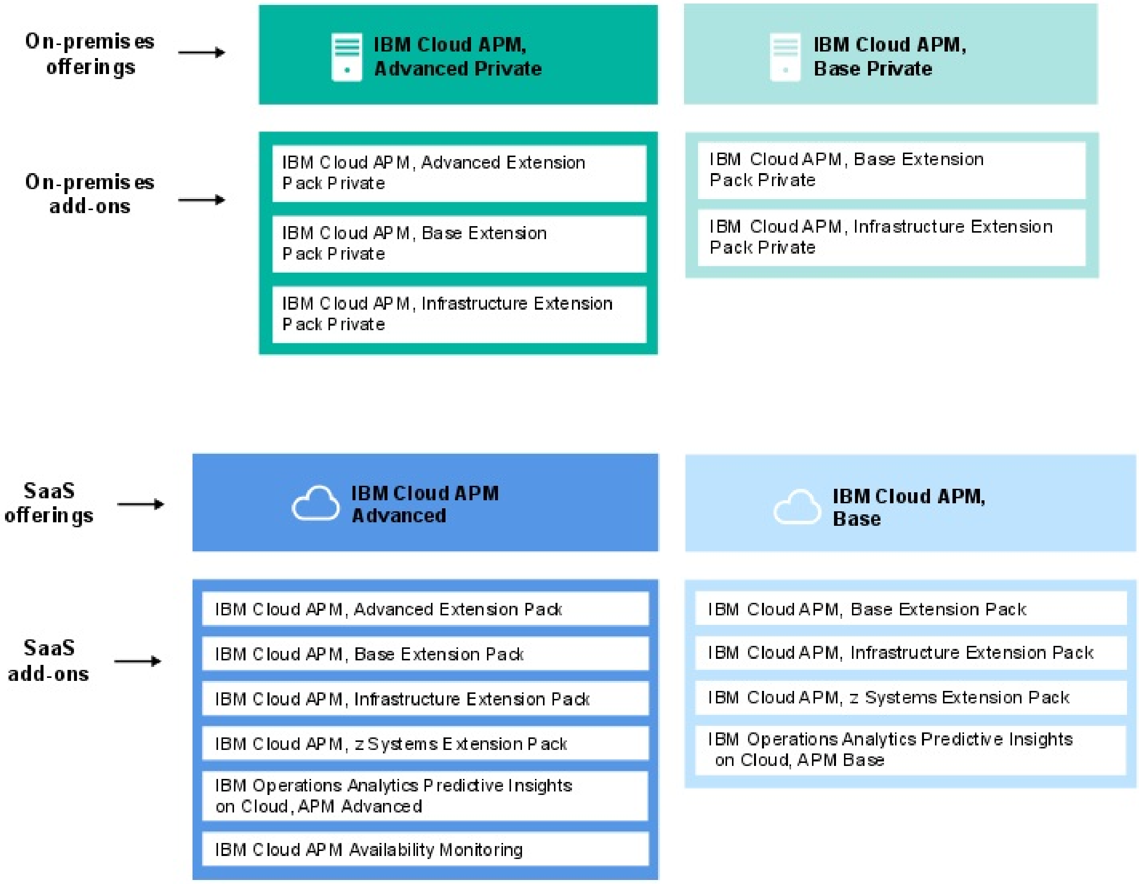
Offerings and add-ons

IBM Cloud Application Performance Management contains two offerings and multiple add-ons. The offerings and add-ons contain agents and data collectors. Specific add-ons can be used with each offering.

To see which agents are included in an offering or add-on and the agent and data collector capabilities, see “Capabilities” on page 56.

For each offering, add-ons are available in IBM Marketplace. IBM Cloud Application Performance Management, Advanced is the most comprehensive offering, the one that includes all agents, data collectors, and dashboard pages. IBM Cloud Application Performance Management, Base is a subset of Cloud APM, Advanced. You can replace Cloud APM, Base with Cloud APM, Advanced at any time. The final installed offering after this replacement is Cloud APM, Advanced. The diagram shows which add-ons are available for each offering.

The add-ons are the same for all offerings, except for Availability Monitoring, which is an add-on only for the Cloud APM, Advanced offering.


On-premises offerings
IBM Cloud APM, Advanced Private
IBM Cloud APM, Base Private
On-premises add-ons
IBM Cloud APM, Advanced Extension Pack Private
IBM Cloud APM, Base Extension Pack Private
IBM Cloud APM, Infrastructure Extension Pack Private
IBM Cloud APM, Base Extension Pack Private
IBM Cloud APM, Infrastructure Extension Pack Private
SaaS offerings
IBM Cloud APM Advanced
IBM Cloud APM, Base
SaaS add-ons
IBM Cloud APM, Advanced Extension Pack
IBM Cloud APM, Base Extension Pack
IBM Cloud APM, Infrastructure Extension Pack
IBM Cloud APM, z Systems Extension Pack
IBM Operations Analytics Predictive Insights on Cloud, APM Advanced
IBM Cloud APM Availability Monitoring
IBM Cloud APM, Base Extension Pack
IBM Cloud APM, Infrastructure Extension Pack
IBM Cloud APM, z Systems Extension Pack
IBM Operations Analytics Predictive Insights on Cloud, APM Base

Offerings

IBM Cloud Application Performance Management, Advanced

This offering is for end user experience, transaction tracking, and resource monitoring of all your application components. You have code level visibility into your applications and the health of your application servers. Use the diagnostics dashboards to find performance bottlenecks in the application code and for managing your critical applications in production.

The offering includes IBM Cloud Application Performance Management, Base, and contains agents and data collectors that you use to monitor applications, transactions, and other resources that are installed in your enterprise. For a list of agents and data collectors in this offering, see "Capabilities" on page 56.

With this offering, DevOps has a complete solution that provides full visibility and control over your applications and infrastructure. Line of business owners can manage critical applications and end user experience in production. Application developers can view transaction details and diagnose application problems.

IBM Cloud Application Performance Management, Base

This offering is for resource monitoring of infrastructure, application components, and cloud workloads. Resource monitoring helps you identify and address slow transactions, capacity issues, and outages. The offering contains agents and data collectors that you use to monitor applications and other resources that are installed in your enterprise. For a list of agents and data collectors in this offering, see "Capabilities" on page 56.

With this offering, IT operators can deal with slow transactions, capacity issues, and outages.

Add-ons

Advanced Extension Pack

This extension pack contains the Monitoring Agent for SAP HANA Database, the SAP NetWeaver Java Stack agent, and the Monitoring Agent for RabbitMQ.

Use the SAP HANA Database agent to monitor the SAP HANA database. Use the SAP NetWeaver Java Stack agent to monitor the SAP NetWeaver Java Stack. Use the RabbitMQ agent to monitor RabbitMQ messaging. This extension pack is available if you have the IBM Cloud Application Performance Management, Advanced offering.

Base Extension Pack

This extension pack contains the following agents:

- Monitoring Agent for Cassandra
- Monitoring Agent for InfoSphere DataStage
- Monitoring Agent for Hadoop
- Monitoring Agent for Microsoft Office 365
- Monitoring Agent for Sterling Connect Direct
- Monitoring Agent for Sterling File Gateway

Use these agents to monitor a Cassandra database, Hadoop cluster, DataStage server resources, Microsoft Office 365 applications, Connect Direct servers, and Sterling File Gateway application. This extension pack is available if you have either of the Cloud APM offerings.

Infrastructure Extension Pack

This extension pack contains the following agents:

- Monitoring Agent for Amazon EC2
- Monitoring Agent for AWS Elastic Load Balancer
- Monitoring Agent for Azure Compute
- Monitoring Agent for Citrix Virtual Desktop Infrastructure
- Monitoring Agent for IBM Cloud

Use the Amazon EC2 agent to monitor your Amazon EC2 instances. Use the Amazon ELB agent to monitor your AWS Elastic Load Balancers. Use the Azure Compute agent to monitor your Azure Compute virtual machines. Use the Citrix VDI agent to monitor your Citrix virtual desktop infrastructure.

This extension pack is available if you have either of the Cloud APM offerings.

z Systems Extension Pack

You can use the z Systems Extension Pack to view monitoring data and events for your OMEGAMON application components in the Cloud APM console. This extension pack is available if you have either of the Cloud APM offerings.

Operations Analytics - Predictive Insights

This add-on is for analyzing the metric data that is collected by Cloud APM, and generating alarms when anomalies are detected. The add-on is available if you have either of the Cloud APM offerings.

Availability Monitoring

This add-on is for monitoring the availability and performance of your web applications from multiple, geographically distributed points of presence. This add-on does not function as a stand-alone offering, but is available if you have the IBM Cloud Application Performance Management, Advanced offering.

For an overview of the features in each offering, see “Offering details” on page 52.

For a description of each agent and data collector and links to information that is specific to each one, see “Descriptions” on page 60.

Offering details

Some features are available for all offerings and others are available only for certain offerings.

Table 1 on page 52 shows key features that are available for each offering at-a-glance.

Table 1. Features in each offering

Feature	Cloud APM, Advanced (For DevOps, Developers, and Line of Business)	Cloud APM, Base (For Operations)
Application resource monitoring: Languages, middleware (coverage varies by offering).	✓	✓
Operating system monitoring: Linux, UNIX, Windows systems	✓	✓
Log file monitoring: Use the OS Agents to monitor application log files.	✓	✓
Dashboards: • View Tivoli Monitoring and Cloud APM KPIs in the same dashboards • Historical metrics • Customizable dashboards	✓	✓
APIs: Manage your environment by using APIs.	✓	✓
Role-based access control: Manage the access and privilege of your IBM Cloud Application Performance Management users.	✓	✓
Historical reporting: Generate reports for the performance and response time of your applications that are broken down by transaction, device, browser, and others (coverage varies by offering).	✓	✓
IBM Agent Builder: Build custom agents to monitor any platform or technology.	✓	✓
Database resource monitoring: (coverage varies by offering)	✓	✓
Infrastructure resource monitoring: Hypervisors, storage, and network (coverage varies by offering).	✓	✓
Commercial applications resource monitoring: Business and collaboration applications (coverage varies by offering).	✓	✓
Response time monitoring: See how your application performance is affecting your users.	✓	✓
Integration with search analytics: Find the insights to quickly isolate, diagnose, and resolve problems.	✓	✓

Table 1. Features in each offering (continued)

Feature	Cloud APM, Advanced (For DevOps, Developers, and Line of Business)	Cloud APM, Base (For Operations)
Operations Analytics - Predictive Insights (add-on): Determine application performance anomalies before they impact your users.	✓	✓
Real end user experience monitoring: See what your users experience from your infrastructure to their device.	✓	–
Transaction tracking: Track end-to-end transactions through your application environment. • Application topology: See how all components are connected in your application environment. • Transaction instance topology: See the path that is followed through your environment for each instance of a transaction.	✓	–
Deep-dive diagnostics: • Drill down from summary dashboards to view code-level, stack trace, and SQL query detail for specific agents. • Detect, diagnose, and kill hung or slow transactions that are still in progress.	✓	–
Thresholds: Detect specific application behaviors and conditions based on actively monitored definitions.	✓	✓
Resource groups: Categorize managed systems in your monitored enterprise by their purpose.	✓	✓

Extra features such as the following are available for all offerings through integration with other products and components. See "Integration " on page 80, and for more details, see Chapter 8, "Integrating with other products and components," on page 929).

- Tivoli Monitoring and OMEGAMON agents: Use the Hybrid Gateway to retrieve monitoring data and events so this information is displayed in the Cloud APM console.
- Agent coexistence: Install Cloud APM agents on the same computer where Tivoli Monitoring agents are installed.
- Netcool/OMNIbus and other EIF receivers: Forward events to IBM Tivoli Netcool/OMNIbus.
- Alert Notification: Receive notification when application performance exceeds thresholds.
- IBM Control Desk: Automatically open tickets in Control Desk.
- IBM Cloud: Monitor IBM Cloud applications.

Agents and data collectors

IBM Cloud Application Performance Management agents and data collectors are available in both the offerings and the add-ons.

Many resources in your environment can be monitored by agents. Some resources on IBM Cloud and on premises can be monitored by data collectors. Corresponding agents exist for all data collectors, except the J2SE and Python data collectors. For a list of agents and data collectors and their descriptions, see "Descriptions" on page 60. To figure out the capabilities that the agent or data collector can provide in each offering, see "Capabilities" on page 56. To find out the change history of each agent and data collector, see "Change history" on page 54.

You can install these agents or data collectors, depending on your environment and requirements. Data collectors send data directly to the Cloud APM server. When an agent is configured, data collectors send data to the agent, which forwards it to the server. Data collectors operate within the application process space, whereas agents run as a separate process outside the application process space.

Install data collectors in the following situations:

- You want a simpler installation process.
- You use containers.

Install agents in the following situations:

- You want greater scalability.
- You want to limit sockets from end points to the server.
- When you add a threshold in the threshold editor, you want a clear list, which contains only the attributes for the environment you want to monitor. If you use a data collector, you must choose from the attributes of several data collectors.
- You want to turn on or off some of the data collection functions on the UI, such as diagnostics, transaction tracking, or method trace.
- You want to view on-demand diagnostics data, such as in-flight requests and heap dump at the current time.

Change history

Find out the information about versions and change history for each agent and data collector.

The following table lists the agent and data collector names with change history technote links. Click the links to view change history details.

Table 2. Agent and data collector change history

Agents and data collectors	Links
Amazon EC2 agent	Change history
Amazon ELB agent	Change history
Azure Compute agent	Change history
Cassandra agent	Change history
Cisco UCS agent	Change history
Citrix VDI agent	Change history
DataPower agent	Change history
DataStage agent	Change history
Db2 agent	Change history
Hadoop agent	Change history

Table 2. Agent and data collector change history (continued)

Agents and data collectors	**Links**
HMC Base agent	Change history
HTTP Server agent	Change history
IBM Cloud agent	Change history
IBM Integration Bus agent	Change history
IBM i OS agent	Change history
Internet Service Monitoring	Change history
Integration Agent for Netcool/OMNIbus	Change history
J2SE data collector	Change history
JBoss agent	Change history
Liberty data collector	Change history
Linux KVM agent	Change history
Linux OS agent	Change history
MariaDB agent	Change history
Microsoft Active Directory agent	Change history
Microsoft Cluster Server agent	Change history
Microsoft Exchange Server agent	Change history
Microsoft Hyper-V Server agent	Change history
Microsoft IIS agent	Change history
Microsoft .NET agent	Change history
Microsoft Office 365 agent	Change history
Microsoft SharePoint Server agent	Change history
Microsoft SQL Server agent	Change history
MongoDB agent	Change history
MQ Appliance agent	Change history
MySQL agent	Change history
NetApp Storage agent	Change history
Node.js agent	Change history
Node.js data collector	Change history
OpenStack agent	Change history
Oracle Database agent	Change history
PHP agent	Change history
PostgreSQL agent	Change history
Python data collector	Change history
RabbitMQ agent	Change history

Table 2. Agent and data collector change history (continued)

Agents and data collectors	Links
Response Time Monitoring Agent	Change history
Ruby agent	Change history
Ruby data collector	Change history
SAP agent	Change history
SAP HANA Database agent	Change history
SAP NetWeaver Java Stack agent	Change history
Siebel agent	Change history
Skype for Business Server agent	Change history
Sterling Connect Direct agent	Change history
Sterling File Gateway agent	Change history
Sybase agent	Change history
Synthetic Playback agent	Change history
Tomcat agent	Change history
UNIX OS agent	Change history
VMware VI agent	Change history
WebLogic agent	Change history
WebSphere Applications agent	Change history
WebSphere Infrastructure Manager agent	Change history
WebSphere MQ agent	Change history
Windows OS agent	Change history

Capabilities

Agent and data collector capabilities vary depending on your offering. The key agent and data collector capabilities are resource monitoring, transaction tracking, and diagnostics. You can subscribe to any of the offerings and add-ons in IBM Cloud Application Performance Management. Specific offerings are required for add-ons.

Each agent and data collector monitors the resources for which the agent or the data collector is named, for example, the Monitoring Agent for Cisco UCS monitors Cisco UCS resources.

Depending on whether you are a developer, in operations, or a line-of-business owner, you use different Cloud APM capabilities.

- Resource monitoring capability includes response time monitoring, application resource monitoring, and infrastructure resource monitoring. All agents and data collectors can provide resource monitoring capability.
- Transaction tracking capability provides transaction instance and topology information.
- Diagnostics capability includes tracing and analyzing individual requests, and when necessary, method calls.

Remember: The resource monitoring capability is common to all offerings and add-ons. The diagnostics and transaction tracking capabilities are available only in the Cloud APM, Advanced offering and add-ons.

The agents and data collectors for the applications that you want to monitor are available for download from **Products and services**. The agents take minutes to install. The data collectors require no installation, and you need to only configure them after the download completes. For instructions about installing the agents, see Chapter 6, "Installing your agents," on page 121.

Table 3 on page 57 provides a comprehensive list of the agents and data collectors, shows which offering or add-on contains the agent or data collector, and shows the capabilities of the agent or data collector. When add-ons (such as Infrastructure Extension Pack) are noted for an agent or a data collector, they are required. Agents and data collectors that support transaction tracking and/or diagnostics capabilities are also noted in the Cloud APM, Advanced column.

✓ indicates the agent or data collector is available in the offering and can provide resource monitoring capability.

— indicates the data or capability is not available in this offering, or the add-on is not required for the agent or data collector.

TT indicates transaction tracking.

DD indicates diagnostics.

Table 3. Agent and data collector capabilities in each offering

Agents and data collectors	Cloud APM, BaseCloud APM, Base Private	Cloud APM, Advanced	Add-on (if required)
Amazon EC2 agent	✓	✓	Infrastructure Extension Pack
Amazon ELB agent	✓	✓	Infrastructure Extension Pack
Azure Compute agent	✓	✓	Infrastructure Extension Pack
Cassandra agent	✓	✓	Base Extension Pack
Cisco UCS agent	✓	✓	—
Citrix VDI agent	✓	✓	Infrastructure Extension Pack
Db2 agent	✓	✓	—
DataPower agent	✓	✓ TT	—
DataStage agent	✓	✓	Base Extension Pack
Hadoop agent	✓	✓	Base Extension Pack
HMC Base agent	✓	✓	—
HTTP Server agent	✓	✓ TT	—

Table 3. Agent and data collector capabilities in each offering (continued)

Agents and data collectors	**Cloud APM, BaseCloud APM, Base Private**	**Cloud APM, Advanced**	**Add-on (if required)**
IBM Cloud agent	✓	✓	Infrastructure Extension Pack
IBM Integration Bus agent	—	✓ TT	—
IBM i OS agent	✓	✓	—
Internet Service Monitoring	—	—	Base Extension Pack
J2SE data collector for on-premises applications	—	✓ TT DD	—
JBoss agent	✓	✓ TT DD	—
Liberty data collector for IBM Cloud and on-premises applications	—	✓ TT DD	—
Linux KVM agent	✓	✓	—
Linux OS agent	✓	✓	—
Microsoft Active Directory agent	✓	✓	—
Microsoft Cluster Server agent	✓	✓	—
Microsoft Exchange Server agent	✓	✓	—
Microsoft Hyper-V Server agent	✓	✓	—
Microsoft IIS agent	✓	✓	—
Microsoft .NET agent	✓	✓ TT DD	—
Microsoft Office 365 agent	✓	✓	Base Extension Pack
Microsoft SharePoint Server agent	✓	✓	—
Microsoft SQL Server agent	✓	✓	—
MongoDB agent	✓	✓	—
MQ Appliance agent	—	✓	—
MySQL agent	✓	✓	—
NetApp Storage agent	✓	✓	—
Node.js agent	✓	✓ DD	—

Table 3. Agent and data collector capabilities in each offering (continued)

Agents and data collectors	**Cloud APM, BaseCloud APM, Base Private**	**Cloud APM, Advanced**	**Add-on (if required)**
Node.js data collector for IBM Cloud and on-premises applications	–	✓ TT DD	–
OpenStack agent	✓	✓	–
Oracle Database agent	✓	✓	–
PHP agent	✓	✓	–
PostgreSQL agent	✓	✓	–
Python data collector for IBM Cloud and on-premises applications	✓	✓ DD	–
RabbitMQ agent	–	✓	Advanced Extension Pack
Response Time Monitoring Agent	✓	✓ TT	–
Ruby agent	✓	✓ DD	–
Ruby data collector for IBM Cloud applications	–	✓ DD	–
SAP agent	–	✓	–
SAP HANA Database agent	–	✓	Advanced Extension Pack
SAP NetWeaver Java Stack agent	–	✓ TT DD	Advanced Extension Pack
Siebel agent	✓	✓	–
Skype for Business Server agent (formerly known as Microsoft Lync Server agent)	✓	✓	–
Sterling Connect Direct agent	✓	✓	Base Extension Pack
Sterling File Gateway agent	✓	✓	Base Extension Pack
Sybase agent	✓	✓	–
Tomcat agent	✓	✓ TT	–
UNIX OS agent	✓	✓	–
VMware VI agent	✓	✓	–

Table 3. Agent and data collector capabilities in each offering (continued)

Agents and data collectors	Cloud APM, BaseCloud APM, Base Private	Cloud APM, Advanced	Add-on (if required)
WebLogic agent	✓	✓ TT DD	–
WebSphere Applications agent	✓	✓ TT DD	–
WebSphere Infrastructure Manager agent	✓	✓	–
WebSphere MQ agent	–	✓ TT	–
Windows OS agent	✓	✓	–

For more information about whether transaction tracking or diagnostics is enabled by default for the agent or data collector, see Transaction tracking enablement for agents and data collectors table. For information about the predefined diagnostics dashboards, see Diagnostics dashboards of agents and data collectors.

Descriptions

The descriptions of the agents and data collectors provide information about what each of these components monitors and links to more information about each component.

Each agent and data collector has a version number, which changes each time the agent or data collector is updated. In any release, new agents and data collectors might be added, and existing agents and data collectors might be updated. If you do not have the latest version of an agent or data collector, consider updating it. For information about how to check the version of an agent or data collector in your environment, see Agent version command.

Each agent and data collector description contains links to the following types of details about these components:

- Agent or data collector configuration and other information about specific agent or data collector capabilities
- Reference PDF that contains descriptions of the Cloud APM agent or data collector dashboards, group widgets, thresholds, data sets, and attributes (metrics and KPIs)

For links to documentation for IBM Tivoli Monitoring V6 and V7 agents that can coexist with Cloud APM V8 agents and data collector, see Table 238 on page 931.

Amazon EC2 monitoring

The Monitoring Agent for Amazon EC2 provides you with a central point of monitoring for the health, availability, and performance of your Amazon Elastic Compute Cloud (EC2) instances. The agent displays a comprehensive set of metrics to help you make informed decisions about your EC2 environment, including CPU utilization, Elastic Block Store (EBS) utilization, network utilization, Amazon Web Services (AWS) maintenance updates, and disk performance.

- For information about configuring the agent after installation, see “Configuring Amazon EC2 monitoring” on page 189.
- For information about the dashboards, thresholds, and attributes, see the Amazon EC2 agent Reference.

AWS Elastic Load Balancer monitoring

The Amazon ELB agent provides you with a central point of monitoring for the health, availability, and performance of your AWS Elastic Load Balancers. The agent displays a comprehensive set of metrics for each load balancer type-application, network and classic-to help you make informed decisions about your AWS Elastic Load Balancer environment.

- For information about configuring the agent after installation, see “Configuring AWS Elastic Load Balancer monitoring” on page 196.
- For information about the dashboards, thresholds, and attributes, see the Amazon ELB agent Reference.

Azure Compute monitoring

The Azure Compute agent provides you with a central point of monitoring for the health, availability, and performance of your Azure Compute instances. The agent displays a comprehensive set of metrics to help you make informed decisions about your Azure Compute environment. These metrics include CPU usage, network usage, and disk performance.

- For information about configuring the agent after installation, see “Configuring Azure Compute monitoring” on page 201.
- For information about the dashboards, thresholds, and attributes, see the Azure Compute agent Reference.

Cassandra monitoring

The Monitoring Agent for Cassandra provides you with the capability to monitor the Cassandra cluster. You can collect and analyze information about the nodes, keyspaces, and column families of the Cassandra cluster.

- For information about configuring the agent after installation, see “Configuring Cassandra monitoring” on page 211.
- For information about the dashboards, thresholds, and attributes, see the Cassandra agent Reference.

Cisco UCS monitoring

The Monitoring Agent for Cisco UCS provides you with an environment to monitor the health, network, and performance of Cisco UCS. The Cisco UCS agent provides a comprehensive way for collecting and analyzing information that is specific to Cisco UCS and required to detect problems early and prevent them.

- For information about configuring the agent after installation, see “Configuring Cisco UCS monitoring” on page 214.
- For information about the dashboards, thresholds, and attributes, see the Cisco UCS agent Reference.

Citrix Virtual Desktop Infrastructure monitoring

The Monitoring Agent for Citrix Virtual Desktop Infrastructure provides you with a central point of monitoring for the health, availability, and performance of your Citrix virtual desktop infrastructure. The agent displays a comprehensive set of metrics to help you make informed decisions about your XenDesktop or XenApp resources, including sites, machines, applications, desktops, sessions, users, and more.

- For information about configuring the agent after installation, see “Configuring Citrix Virtual Desktop Infrastructure monitoring” on page 220.
- For information about the dashboards, thresholds, and attributes, see the Citrix VDI agent Reference.

DataPower monitoring

The Monitoring Agent for DataPower provides a central point of monitoring for the DataPower Appliances in your enterprise environment. You can identify and receive notifications about common problems with the appliances. The agent also provides information about performance, resource, and workload for the appliances.

- For information about configuring the agent after installation, see “Configuring the DataPower agent” on page 238.
- For information about the dashboards, thresholds, and attributes, see the DataPower agent Reference.

- For information about monitoring DataPower appliances as part of the IBM integration stack, see "Monitoring the IBM integration stack" on page 99.

Db2 monitoring

The Monitoring Agent for Db2 offers a central point of monitoring for your Db2 environment. You can monitor a multitude of servers from a single IBM Performance Management console, with each server monitored by a Db2 agent. You can collect and analyze information in relation to applications, databases, and system resources.

- For information before you upgrade to a new version of the agent, see "Agents on AIX: Stopping the agent and running slibclean before you upgrade" on page 1114
- For information about configuring the agent after installation, see "Configuring Db2 monitoring " on page 241.
- For information about the dashboards, thresholds, and attributes, see the Db2 agent Reference.
- For information about monitoring database transactions as part of the IBM Java application stack, see "Monitoring the IBM Java application stack" on page 92.

Hadoop monitoring

The Monitoring Agent for Hadoop provides capabilities to monitor the Hadoop cluster in your organization. You can use the agent to collect and analyze information about the Hadoop cluster, such as status of data nodes and Java virtual machine, memory heap and non-heap information, and information about Hadoop nodes, file systems, and queues.

- For information about configuring the agent after installation, see "Configuring Hadoop monitoring" on page 251.
- For information about the dashboards, thresholds, and attributes, see the Hadoop agent Reference.

HMC Base monitoring

The Monitoring Agent for HMC Base provides you with the capability to monitor the Hardware Management Console (HMC). The agent monitors the availability and health of the HMC resources: CPU, memory, storage, and network. The agent also reports on the HMC inventory and configuration of Power servers, CPU pools, and LPARs. The CPU utilization of the Power servers, LPARs, and pools are monitored by using HMC performance sample data.

- For information about configuring the agent after installation, see "Configuring HMC Base monitoring" on page 260.
- For information about the dashboards, thresholds, and attributes, see the HMC Base agent Reference.

HTTP Server monitoring

The Monitoring Agent for HTTP Server collects performance data about the IBM HTTP Server. For example, server information, such as the status and type of server, the number of server errors, and the number of successful and failed logins to the server are shown. A data collector gathers the data that is sent to the HTTP Server agent. The agent runs on the same system with the IBM HTTP Server that it monitors. Each monitored server is registered as a subnode. The IBM HTTP Server Response Time module is installed with the HTTP Server agent. When you use the HTTP Server agent with the Response Time Monitoring agent, the WebSphere Application agent, and a database agent, you can see transaction monitoring information from the browser to the database for the IBM Java application stack.

- Before you begin the agent installation, see Preinstallation on AIX systems - HTTP Server agent and Preinstallation on Linux systems HTTP Server agent.
- For instructions on how to review the data collector settings and activate the data collector after agent installation, see "Configuring HTTP Server monitoring" on page 265.
- For information about the dashboards, thresholds, and attributes, see the HTTP Server agent Reference.
- For information about monitoring HTTP server transactions as part of the IBM Java application stack, see "Monitoring the IBM Java application stack" on page 92.

IBM Cloud monitoring

The Monitoring Agent for IBM Cloud collects virtual machine inventory and metrics from your IBM Cloud (Softlayer) account. Use the IBM Cloud agent to track how many virtual devices you have configured and running in IBM Cloud. You can see what resources are allocated to each virtual device in the detailed dashboard page, which also shows information like the data center a device is located in, the operating system, and the projected public network bandwidth for the month.

- For information about configuring the agent after installation, see Configuring IBM Cloud monitoring.
- For information about the dashboards, thresholds, and attributes, see the IBM Cloud agent Reference.

IBM Integration Bus monitoring

The Monitoring Agent for IBM Integration Bus is a monitoring and management tool that provides you with the means to verify, analyze, and tune message broker topologies that are associated with the IBM WebSphere Message Broker and IBM Integration Bus products.

- For information about configuring the agent after installation, see “Configuring IBM Integration Bus monitoring” on page 273.
- For information about the dashboards, thresholds, and attributes, see the IBM Integration Bus agent Reference.
- For information about monitoring IBM Integration Bus brokers as part of the IBM integration stack, see “Monitoring the IBM integration stack” on page 99.

InfoSphere DataStage monitoring

The monitoring agent for InfoSphere DataStage monitors the availability, resource usage, and performance of the DataStage Server. The agent monitors health status of the engine nodes and jobs. You can analyze the information that the agent collects and take appropriate actions to resolve issues in the DataStage Server.

- For information about configuring the agent after installation, see Configuring InfoSphere DataStage monitoring.
- For information about the dashboards, thresholds, and attributes, see the DataStage agent Reference.

Internet Service Monitoring

The Internet Service Monitoring offers to determine whether a particular service is performing adequately, identify problem areas and report service performance measured against Service Level Agreements. Internet Service Monitoring agent works by emulating the actions of a real user. It regularly poll or test Internet services to check their status and performance.

- For information about configuring the agent after installation see “Configuring the agent on Windows systems” on page 438
- For information about the dashboards, thresholds, and attributes, see the Internet Service Monitoring agent Reference

J2SE data collector monitoring

The J2SE data collector collects resource monitoring and deep-dive diagnostics data for Java applications. The deep-dive diagnostics data is shown in the dashboards based on requests and aggregated information to support various drill-down views. Both resource monitoring and deep-dive diagnostics are supported, which helps detect, isolate, and diagnose issues with Java applications. You can configure the data collector to diagnose slow requests.

- For information about configuring the data collector, see Configuring the J2SE data collector.
- For information about the dashboards, thresholds, and attributes, see the J2SE data collector Reference.

JBoss monitoring

The Monitoring Agent for JBoss monitors the resources of JBoss application servers and the JBoss Enterprise Application platform. Use the dashboards that are provided with the JBoss agent to identify the slowest applications, slowest requests, thread pool bottlenecks, JVM heap memory and garbage collection issues, busiest sessions, and other bottlenecks on the JBoss application server.

- For information about configuring the agent after installation, see "Configuring JBoss monitoring" on page 448.
- For information about the dashboards, thresholds, and attributes, see the JBoss agent Reference.

Linux KVM monitoring

The Monitoring Agent for Linux KVM is a multi-instance and multi-connection agent and supports connections to the Enterprise Linux based KVM hypervisor and Red Hat Enterprise Virtualization Manager (RHEV-M) environments. You can create multiple instances of this agent to monitor multiple hypervisors in an RHEV-M or KVM hypervisor environment. You can monitor virtualized workloads and analyze the resource capacity across different virtual machines. To connect the agent to a virtual machine in the KVM hypervisor environment, you must install the prerequisites: `libvirt*.rpm` and Korn Shell Interpreter (pdksh). The agent collects metrics by connecting remotely to a libvirt hypervisor that manages the virtual machines.

- For information about configuring the agent after installation, see "Configuring Linux KVM monitoring" on page 474.
- For information about the dashboards, thresholds, and attributes, see the Linux KVM agent Reference.

Linux OS monitoring

The Monitoring Agent for Linux OS provides monitoring capabilities for the availability, performance, and resource usage of the Linux OS environment. This agent supports Docker container monitoring. For example, detailed information such as the CPU usage, memory, network and I/O usage information that relates to the docker container is shown. General information about the docker containers running on the server, such as the docker ID and instance name is also shown. Also, you can configure log file monitoring to monitor application log files. You can collect and analyze server-specific information, such as operating system and CPU performance, Linux disk information and performance analysis, process status analysis, and network performance.

- For information about configuring log file monitoring after installation, see "Configuring OS agent log file monitoring" on page 626.
- For information about the dashboards, thresholds, and attributes, see the Linux OS agent Reference.

MariaDB monitoring

The Monitoring Agent for MariaDB offers a central point of management for your MariaDB environment or application. The software provides a comprehensive means for gathering the information required to detect problems early and to prevent them. Information is standardized across the system. You can monitor multiple servers from a single console. By using the Monitoring Agent for MariaDB you can easily collect and analyze MariaDB specific information.

- For information about configuring the agent after installation, see "Configuring MariaDB monitoring" on page 485.
- For information about the dashboards, thresholds, and attributes, see the MariaDB agent Reference.

Microsoft Active Directory monitoring

The Monitoring Agent for Microsoft Active Directory provides capabilities to monitor the Active Directory in your organization. You can use the agent to collect and analyze information that is specific to Active Directory, such as network status, Sysvol replication, address book performance, and directory system usage.

- For information about configuring the agent after installation, see "Configuring Microsoft Active Directory monitoring" on page 488.
- For information about the dashboards, thresholds, and attributes, see the Microsoft Active Directory agent Reference.

Microsoft Cluster Server monitoring

The Monitoring Agent for Microsoft Cluster Server provides capabilities to monitor the Microsoft Cluster Server in your organization. You can use the Microsoft Cluster Server agent to collect information that is related to cluster resource availability, such as cluster level, cluster nodes, cluster resource groups, cluster resources, and cluster networks. The agent also provides statistics for cluster resources usage, such as processor usage, memory usage, disk usage, and network usage.

- For information about configuring the agent after installation, see "Configuring Microsoft Cluster Server monitoring" on page 494 .
- For information about the dashboards, thresholds, and attributes, see the Microsoft Cluster Server agent Reference.

Microsoft Exchange Server monitoring

The Monitoring Agent for Microsoft Exchange Server provides capabilities to monitor the health, availability, and performance of the Exchange Servers in your organization. You can use the Microsoft Exchange Server agent to collect server-specific information, such as mail traffic, state of mailbox databases, and activities of clients. Additionally, the agent provides statistics of cache usage, mail usage, database usage, and client activities to help you analyze the performance of Exchange Servers.

- For information about configuring the agent after installation, see "Configuring Microsoft Exchange monitoring" on page 496.
- For information about the dashboards, thresholds, and attributes, see the Microsoft Exchange Server agent Reference.

Microsoft Hyper-V Server monitoring

The Monitoring Agent for Microsoft Hyper-V Server provides capability to monitor the availability and performance of all the Hyper-V systems in your organization. The Microsoft Hyper-V Server agent provides configuration information such as the number of virtual machines, the state of the virtual machines, the number of allocated virtual disks, the allocated virtual memory, and the number of allocated virtual processors. Additionally, the agent provides statistics of physical processor usage, memory usage, network usage, logical processor usage, and virtual processor usage.

- For information about configuring the agent after installation, see "Configuring Microsoft Hyper-V monitoring" on page 509.
- For information about the dashboards, thresholds, and attributes, see the Microsoft Hyper-V Server agent Reference.

Microsoft Internet Information Services monitoring

The Monitoring Agent for Microsoft Internet Information Services provides you with the capability to monitor the availability and performance of Microsoft Internet Information Server. You can use the Microsoft Internet Information Server agent to monitor website details such as request rate, data transfer rate, error statistics, and connections statistics.

- For information about configuring the agent after installation, see "Configuring Microsoft IIS monitoring" on page 513.
- For information about the dashboards, thresholds, and attributes, see the Microsoft IIS agent Reference.

Microsoft .NET monitoring

The Monitoring Agent for Microsoft .NET monitors Microsoft .NET applications that are based on Internet Information Services (IIS) and Microsoft .NET Framework resources. The data collector component collects data from incoming HTTP requests. The data collector collects method calls and constructs a call tree, and collects request context and stack trace data. Use the dashboards that are provided with the Microsoft .NET agent to identify the problems that are associated with Microsoft .NET Framework, and also to identify the slowest HTTP requests from where you can drill down to stack trace information to isolate problems.

- For information about configuring the agent after installation, see "Registering the data collector" on page 522.

- For information about the dashboards, thresholds, and attributes, see the Microsoft .NET agent Reference.

Microsoft Office 365 monitoring

The Monitoring Agent for Microsoft Office 365 provides you with the capability to monitor the Microsoft Office 365. You can collect and analyze information about Microsoft Exchange Online, SharePoint Online, Skype for Business, and OneDrive for Business.

- For information about configuring the agent after installation, see "Configuring Microsoft Office 365 monitoring" on page 530.
- For information about the dashboards, thresholds, and attributes, see the Microsoft Office 365 agent Reference.

Microsoft SharePoint Server monitoring

The Monitoring Agent for Microsoft SharePoint Server provides you with the environment to monitor the availability, events, and performance of the Microsoft SharePoint Server. Use this agent to gather data from the Microsoft SharePoint Server and manage operations.

- For information about configuring the agent after installation, see "Configuring Microsoft SharePoint Server monitoring " on page 537.
- For information about the dashboards, thresholds, and attributes, see the Microsoft SharePoint Server agent Reference.

Microsoft SQL Server monitoring

The Monitoring Agent for Microsoft SQL Server provides you with the capability to monitor the Microsoft SQL Server. The Microsoft SQL Server agent offers a central point of management for distributed databases. Use the Microsoft SQL Server agent dashboards to monitor the availability, performance, resource usage, and the overall status of all the SQL Server instances that are being monitored.

- For information about configuring the agent after installation, see "Configuring Microsoft SQL Server monitoring " on page 540.
- For information about the dashboards, thresholds, and attributes, see the Microsoft SQL Server agent Reference.

MongoDB monitoring

The Monitoring Agent for MongoDB provides monitoring capabilities for the usage, status, and performance of the MongoDB deployment. You can collect and analyze information such as database capacity usage, percentage of connections open, memory usage, instance status, and response time in visualized dashboards.

- For information about configuring the agent after installation, see "Configuring MongoDB monitoring" on page 570.
- For information about the dashboards, thresholds, and attributes, see the MongoDB agent Reference.

MQ Appliances monitoring

The Monitoring Agent for MQ Appliance provides monitoring information that focuses on the MQ appliance level on MQ Appliances, for example, CPU, memory, storage, sensors, and queue managers summary information.

- For information about configuring the agent after installation, see "Configuring IBM MQ Appliances monitoring" on page 287.
- For information about the dashboards, thresholds, and attributes, see the MQ Appliance agent Reference.

MySQL monitoring

The Monitoring Agent for MySQL provides monitoring capabilities for the status, usage, and performance of the MySQL deployment. You can collect and analyze information such as Bytes Received vs Sent, InnoDB Buffer Pool Pages, and Historical Performance.

- Before you begin the agent installation, see Preinstallation on Linux systems - MySQL agent or Preinstallation on Windows systems - MySQL agent.
- For information about configuring the agent after installation, see "Configuring MySQL monitoring" on page 575.
- For information about the dashboards, thresholds, and attributes, see the MySQL agent Reference.

NetApp storage monitoring

The Monitoring Agent for NetApp Storage provides you with the capability to monitor the NetApp storage systems by using the NetApp OnCommand Unified Manager (OCUM). You can collect and analyze information about the aggregates, nodes, disks, and volumes of the NetApp storage systems.

- For information about configuring the agent after installation, see "Configuring NetApp Storage monitoring" on page 577.
- For information about the dashboards, thresholds, and attributes, see the NetApp Storage agent Reference.

Node.js monitoring

The Monitoring Agent for Node.js or the stand-alone Node.js data collector can be used to measure and collect data about the performance of Node.js applications. For example, throughput and response times for HTTP requests, and other measurements that relate to resource usage, are monitored and stored for display and analysis. To choose between the Node.js agent and the Node.js data collector, see "Configuring Node.js monitoring" on page 584 for instructions.

Node.js agent

- Before you begin the installation, see Preinstallation on Linux systems - Node.js agent.
- For information about configuring the agent after installation, see "Configuring the Node.js agent" on page 585.
- For information about the dashboards, thresholds, and attributes, see the Node.js agent Reference.

Node.js data collector (stand-alone)

The Node.js data collector monitors IBM Cloud and on-premises applications. Resource monitoring and deep-dive diagnostics are supported, which helps detect, isolate, and diagnose issues of your applications. You can configure the data collector to track the performance of individual request and method calls, and use the information to diagnose slow requests and take actions accordingly.

IBM Cloud applications

- For information about configuring the data collector, see "Configuring the stand-alone Node.js data collector for IBM Cloud(formerly Bluemix) applications" on page 590.
- For information about the dashboards, thresholds, and attributes, see the Data collectors Reference.

On-premises applications

- For information about configuring the data collector, see "Configuring the stand-alone Node.js data collector for on-premises applications" on page 595.
- For information about the dashboards, thresholds, and attributes, see the Data collectors Reference.

OpenStack monitoring

The Monitoring Agent for OpenStack provides with you the capabilities to monitor your OpenStack applications. Use the dashboards to view the performance of your OpenStack applications, such as information about API endpoints, SSH sever connection, processes, and hypervisors.

- For information about configuring the agent after installation, see "Configuring the OpenStack agent" on page 605.
- For information about the dashboards, thresholds, and attributes, see the OpenStack agent Reference.

Oracle Database monitoring

The Monitoring Agent for Oracle Database provides monitoring capabilities for the availability, performance, and resource usage of the Oracle database. You can configure more than one Oracle Database agent instance to monitor different Oracle databases. Remote monitoring capability is also provided by this agent.

- Before you begin the agent installation, see Preinstallation on AIX systems - Oracle Database agent, Preinstallation on Linux systems - Oracle Database agent, or Preinstallation on Windows systems - Oracle Database agent (Windows).
- For instructions on configuring the agent after installation, see "Configuring Oracle Database monitoring" on page 610.
- For information about the dashboards, thresholds, and attributes, see the Oracle Database agent Reference.
- For information about monitoring database transactions as part of the IBM Java application stack, see "Monitoring the IBM Java application stack" on page 92.

PHP monitoring

The Monitoring Agent for PHP monitors PHP web applications by collecting web access metrics through an Apache web server and performance statistics data from MySQL. The agent discovers all WordPress applications on an Apache server and provides WordPress application statistics information. Use the PHP agent to monitor web server availability, Apache server status, and GET/POST requests. The agent evaluates only the performance of PHP requests in WordPress applications. CSS and JS loading are not evaluated. The agent does not use URL arguments to identify URLs.

- For information about configuring the agent after installation, see "Configuring PHP monitoring" on page 658.
- For information about the dashboards, thresholds, and attributes, see the PHP agent Reference.

PostgreSQL monitoring

The Monitoring Agent for PostgreSQL monitors the PostgreSQL database by collecting PostgreSQL metrics through a JDBC driver. The agent provides data about system resource usage, database capacity, connections that are used, individual status of running instances, statistics for operations, response time for SQL query statements, database size details, and lock information.

- For information about configuring the agent after installation, see "Configuring PostgreSQL monitoring" on page 660.
- For information about the dashboards, thresholds, and attributes, see the PostgreSQL agent Reference.

Python monitoring

The Python data collector monitors both on-prem and IBM Cloud Python applications. Both resource monitoring and deep-dive diagnostics are supported, which provides monitoring data such as CPU and memory usage, garbage collection, and threads. You can configure the data collector to track the performance of individual request and method calls, and use the information to diagnose slow requests and take actions accordingly.

IBM Cloud applications

- For information about configuring the data collector, see "Configuring the Python data collector for IBM Cloud applications" on page 664.
- For information about the dashboards, thresholds, and attributes, see the Data collectors Reference.

On-premises applications

- For information about configuring the data collector, see "Configuring the Python data collector for on-premises applications" on page 669.
- For information about the dashboards, thresholds, and attributes, see the Data collectors Reference.

RabbitMQ monitoring

The Monitoring Agent for RabbitMQ provides you with the capability to monitor the RabbitMQ cluster. You can collect and analyze information about the nodes, queues, and channels of the RabbitMQ cluster.

- For information about configuring the agent after installation, see "Configuring RabbitMQ monitoring " on page 674.
- For information about the dashboards, thresholds, and attributes, see the RabbitMQ agent Reference.

Response Time monitoring

The Response Time Monitoring Agent uses network monitoring to capture HTTP and HTTPS transaction data such as response times and status codes. Use the Response Time Monitoring agent to monitor the performance and availability of web applications for users, including transaction request, application, and server information. Also, use this agent to monitor devices and session information.

- Before you begin the Response Time Monitoring agent installation, see Preinstallation on AIX systems - Response Time Monitoring agent, Preinstallation on Linux systems - Response Time Monitoring agent, or Preinstallation on Windows systems - Response Time Monitoring agent.
- For information about configuring the agent after installation, see "JavaScript Injection" on page 681.
- For information about the dashboards, thresholds, and attributes, see the Transaction Monitoring Reference.
- For information about using Response Time Monitoring as part of the IBM Java application stack, see "Monitoring the IBM Java application stack" on page 92.

Ruby monitoring

The Monitoring Agent for Ruby or the stand-alone Ruby data collectors monitor the performance of your Ruby on Rails applications, including request traffic and configuration statistics. You can also use the diagnostic function to get a deeper view into each application.

The standalone Ruby data collector monitors only IBM Cloud applications.

Ruby agent

- For information about configuring the agent after installation, see "Configuring Ruby monitoring" on page 708.
- For information about the dashboards, thresholds, and attributes, see the Ruby agent Reference.

Ruby data collector (stand-alone)

You can use the Ruby data collector to monitor IBM Cloud applications. Both resource monitoring and deep-dive diagnostics are supported, which helps detect, isolate, and diagnose issues of your applications. You can configure the data collector to track the performance of individual request and method calls, and use the information to diagnose slow requests and take actions accordingly.

IBM Cloud applications

- For information about configuring the data collector, see Configuring the Ruby data collector.
- For information about the dashboards, thresholds, and attributes, see the Data collectors Reference.

SAP applications monitoring

The Monitoring Agent for SAP Applications provides you the capability to monitor your SAP applications that run on the Advanced Business Application Programming (ABAP) stack. The agent also monitors the SAP Solution Manager, which is an SAP lifecycle management tool, and the SAP NetWeaver Process Integration (SAP PI), which is an enterprise integration software for SAP. It offers a central point of management for gathering the information that you need to detect problems early, and to take steps to prevent them from recurring. It enables effective systems management across

SAP releases, applications, and components; and the underlying databases, operating systems, and external interfaces.

- For information about configuring the agent after installation, see "Configuring SAP monitoring" on page 719.
- For information about the dashboards, thresholds, and attributes, see the SAP agent Reference.

SAP HANA Database monitoring

The Monitoring Agent for SAP HANA Database monitors the availability, resource usage, and performance of the SAP HANA database. The agent can monitor HANA deployment scenarios such as single host - single database, single host - multiple tenant databases, multiple hosts - single database, and multiple hosts - multiple tenant databases. You can analyze the information that the agent collects and take appropriate actions to resolve issues in the SAP HANA database.

- Before you begin the agent installation, see Preinstallation on AIX systems - SAP HANA Database agent or Preinstallation on Linux systems - SAP HANA Database agent or Preinstallation on Windows systems - SAP HANA Database agent.
- For information about configuring the agent after installation, see "Configuring SAP HANA Database monitoring" on page 756.
- For information about the dashboards, thresholds, and attributes, see the SAP HANA Database agent Reference.

SAP NetWeaver Java Stack monitoring

The Monitoring Agent for SAP NetWeaver Java Stack monitors the availability, resource usage, and performance of the SAP NetWeaver Java Stack. The agent can monitor SAP NetWeaver Java Stack deployment scenarios such as single host - single instance, single host - multiple instances, multiple hosts - single instances, and multiple hosts - multiple instances. You can analyze the information that the agent collects and take appropriate actions to resolve issues in the SAP NetWeaver Java Stack.

- For information about configuring the agent after installation, see "Configuring SAP NetWeaver Java Stack monitoring" on page 759.
- For information about the dashboards, thresholds, and attributes, see the SAP NetWeaver Java Stack agent Reference.

Siebel monitoring

The Monitoring Agent for Siebel provides a central point of monitoring for your Siebel resources, which includes Siebel statistics, user sessions, components, tasks, application server, Siebel Gateway Name Server, process CPU and memory usage, and log event monitoring.

- For information about configuring the agent after installation, see "Configuring Siebel monitoring" on page 766.
- For information about the dashboards, thresholds, and attributes, see the Siebel agent Reference.

Skype for Business Server (formerly known as Microsoft Lync Server) monitoring

The Monitoring Agent for Skype for Business Server provides you with the capability to monitor the health, availability, and performance of the Skype for Business Server. You can use the Skype for Business Server agent to collect server-specific information, such as latency, synthetic transactions, call details recording (CDR) service write operations, state of throttled requests, and session initiation protocol (SIP) peers. Additionally, the agent provides historical usage statistics of instant messaging and mediation server to help you analyze the performance of Lync or Skype for Business Servers.

- For information about configuring the agent after installation, see "Configuring Skype for Business Server monitoring" on page 515.
- For information about the dashboards, thresholds, and attributes, see the Skype for Business Server agent Reference.

Sterling Connect Direct monitoring

The Monitoring Agent for Sterling Connect Direct provides monitoring of Connect Direct nodes. It provides you with health and performance of the servers. Also, it gives analysis of file transfer activity.

- For information about configuring the agent after installation, see “Configuring Sterling Connect Direct monitoring” on page 777.
- For information about the dashboards, thresholds, and attributes, see the Sterling Connect Direct agent Reference.

Sterling File Gateway monitoring

The Monitoring Agent for Sterling File Gateway monitors the Sterling File Gateway application, which is used for transferring files between internal and external partners by using different protocols, different file naming conventions, and different file formats. It also supports the remote monitoring feature.

- For information about configuring the agent after installation, see “Configuring Sterling File Gateway monitoring” on page 779.
- For information about the dashboards, thresholds, and attributes, see the Sterling File Gateway agent Reference.

Sybase Server monitoring

The Monitoring Agent for Sybase Server offers a central point of management for distributed databases. It collects the required information for database and system administrators to examine the performance of the Sybase server system, detect problems early and prevent them.

- For information about configuring the agent after installation, see “Configuring Sybase Server monitoring” on page 785.
- For information about the dashboards, thresholds, and attributes, see the Sybase agent Reference.

Tomcat monitoring

The Monitoring Agent for Tomcat monitors the resources of Tomcat application servers. Use the dashboards that are provided with the Tomcat agent to identify the slowest applications, slowest requests, thread pool bottlenecks, JVM heap memory and garbage collection issues, the busiest sessions, and other bottlenecks on the Tomcat application server.

- For information about configuring the agent after installation, see “Configuring Tomcat monitoring” on page 794.
- For information about the dashboards, thresholds, and attributes, see the Tomcat agent Reference.

UNIX OS monitoring

The Monitoring Agent for UNIX OS provides monitoring capabilities for the availability, performance, and resource usage of the UNIX OS environment . Also, you can configure log file monitoring to monitor application log files. You can collect and analyze server-specific information, such as operating system and CPU performance, UNIX disk information and performance analysis, process status analysis, and network performance.

- For information about configuring log file monitoring after installation, see “Configuring OS agent log file monitoring” on page 626.
- For information about the dashboards, thresholds, and attributes, see the UNIX OS agent Reference.

VMware VI monitoring

The Monitoring Agent for VMware VI monitors the VMware Virtual Infrastructure by connecting to the VMware Virtual Center. You can use the VMware VI agent to view the status summary for clusters and monitor multiple components, such as clusters, virtual machines, data stores, and ESX servers from a single console.

- For information about configuring the agent after installation, see “Configuring VMware VI monitoring” on page 804.
- For information about the dashboards, thresholds, and attributes, see the VMware VI agent Reference.

WebLogic monitoring

The Monitoring Agent for WebLogic provides you with a central point of monitoring for the health, availability, and performance of your WebLogic server environment. The agent displays a comprehensive set of metrics to help you make informed decisions about your WebLogic resources, including Java virtual machines (JVMs), Java messaging service (JMS), Java Database Connectivity (JDBC).

- For information about configuring the agent after installation, see "Configuring WebLogic monitoring" on page 813.
- For information about the dashboards, thresholds, and attributes, see the WebLogic agent Reference.

WebSphere Applications monitoring

The Monitoring Agent for WebSphere Applications with the embedded data collector, or the stand-alone Liberty data collector monitor the resources of WebSphere application servers. These monitoring components can be configured to do the following things:

- Gather PMI metrics for resource monitoring through a JMX interface on the application server.
- Gather aggregated request performance metrics.
- Track the performance of individual request and method calls.

The monitoring data is displayed in the dashboards. You can use the provided dashboards to isolate specific problem areas of your application server. Drill down to determine whether a problem lies with an underlying resource or if it relates to the application's code.

For information about whether to use the agent or one of the data collectors, see "Configuring WebSphere Applications monitoring" on page 828.

WebSphere Applications agent and embedded data collector

- For information about configuring the agent after installation, see "Configuring the data collector for WebSphere Applications agent" on page 828.
- For information about the dashboards, thresholds, and attributes, see the WebSphere Applications agent Reference.
- For information about monitoring WebSphere application server transactions as part of the IBM Java application stack, see "Monitoring the IBM Java application stack" on page 92.

Liberty data collector (stand-alone)

You can use the Liberty data collector to monitor WebSphere Liberty profile on IBM Cloud or to monitor WebSphere Application Server Liberty on Linux for System x. Resource monitoring, diagnostics, and transaction tracking are all supported, which helps detect, isolate, and diagnose issues of your applications. You can configure the stand-alone data collector to track the performance of individual request and method calls, and use the information to diagnose slow requests and take actions accordingly.

IBM Cloud applications

- For information about configuring the data collector, see "Configuring the Liberty data collector in IBM Cloud environment (Liberty V18.* and older versions)" on page 468.
- For information about the dashboards, thresholds, and attributes, see the Data collectors Reference.

On-premises applications (Linux for System x only)

- For information about configuring the data collector, see "Configuring the Liberty data collector in on-premises environments (Liberty V18.* and older versions)" on page 464.
- For information about the dashboards, thresholds, and attributes, see the Data collectors Reference.

WebSphere Infrastructure Manager monitoring

The Monitoring Agent for WebSphere Infrastructure Manager provides the monitoring capabilities for the WebSphere Application Server Deployment Manager and Node Agent, including server status, resources, and transactions. You can use the data that is collected by the WebSphere Infrastructure Manager agent to analyze the performance of your Deployment Manager and Node Agent, and whether a problem occurred.

- For information about configuring the agent after installation, see "Configuring WebSphere Infrastructure Manager monitoring" on page 910.
- For information about the dashboards, thresholds, and attributes, see the WebSphere Infrastructure Manager agent Reference.

WebSphere MQ monitoring

With the Monitoring Agent for WebSphere MQ, you can easily collect and analyze data that is specific to WebSphere MQ for your queue managers from a single vantage point. You can then track trends in the data that is collected and troubleshoot system problems by using the predefined dashboards.

- For information about configuring the agent after installation, see "Configuring WebSphere MQ monitoring" on page 911.
- For information about the dashboards, thresholds, and attributes, see the WebSphere MQ agent Reference.
- For information about monitoring message queues as part of the IBM integration stack, see "Monitoring the IBM integration stack" on page 99.

Windows OS monitoring

The Monitoring Agent for Windows OS provides monitoring capabilities for the availability, performance, and resource usage of the Windows OS environment. Also, you can configure log file monitoring to monitor application log files. You can collect and analyze server-specific information, such as operating system and CPU performance, disk information and performance analysis, process status analysis, Internet session data, monitored logs information, Internet server statistics, message queuing statistics, printer and job status data, Remote Access Services statistics, and services information. The KNTCMA_FCProvider service is installed with the agent.

- For information about configuring log file monitoring after installation, see "Configuring OS agent log file monitoring" on page 626.
- For information about the dashboards, thresholds, and attributes, see the Windows OS agent Reference.

Features

The key features vary by offering. Some features are available in one or both offerings, in an add-on, or through integration with other products and components.

Application resource monitoring

Use resource monitoring agents to monitor languages and middleware. Coverage varies by offering. See "Capabilities" on page 56.

Operating system monitoring

Use resource monitoring agents to monitor Linux, UNIX, and Windows operating systems. See "Capabilities" on page 56.

Log file monitoring

The OS agents contain a feature to monitor application log files. This feature includes the capability to configure log file monitoring based on regular expressions.

For compatibility, the OS agent consumes the following information and formats:

- Configuration information and the format file that was used by the IBM Tivoli Monitoring Log File Agent V6.x
- Configuration information and format strings that were used by the Tivoli Event Console Log File Adapter

These format strings allow the agent to filter the log data according to patterns in the format file, and submit only the relevant data to an event consumer. The OS agent sends data to the Cloud APM server or through the Event Integration Facility (EIF) to any EIF receiver, such as the Netcool/OMNIbus Probe for Tivoli EIF.

Dashboards

The **Application Performance Dashboard** gives you a high-level status of the applications in your environment. View areas of interest either by selecting from the navigator or by clicking in a summary box to drill down to the next level.

To learn about the features that are available at each dashboard level, see "All My Applications - Application Performance Dashboard" on page 1055, "Application - Application Performance Dashboard" on page 1058, and "Group and Instance - Application Performance Dashboard" on page 1063.

View KPIs from the Tivoli Monitoring and Cloud APM domains in the same dashboards

In an environment that includes both IBM Tivoli Monitoring and IBM Cloud Application Performance Management products, you can install the IBM Cloud Application Performance Management Hybrid Gateway to provide a consolidated view of managed systems from both domains. To view your hybrid environment in the Cloud APM console, you must create a managed system group, install the Hybrid Gateway in your Tivoli Monitoring environment, and configure communications with the Hybrid Gateway.

For more information, see "Integrating with IBM Tivoli Monitoring V6.3 " on page 929.

Historical metrics

Get visualizations of up to 24 hours of historical data on the Application Performance Dashboard. When a time selector is displayed in a dashboard's **Status Overview** tab, you can adjust the time range for the charts and tables whose values are derived from historical data samples. For line charts, you can also compare the current data, up to the past 24 hours, with up to 8 days of historical data to spot abnormalities.

For more information, see "Adjusting and comparing metrics over time" on page 1067.

IBM Cloud Application Business Insights Universal View

You can use the Universal View to create customized pages for the applications you are monitoring. Choose from different chart and metric options to create widgets to monitor data according to your requirements. With Universal View, you can customize a dashboard to view consolidated data from multiple agents.

When you are viewing data on the dashboard, you can change the chart type dynamically. On the grid widget, you can filter data dynamically.

You can export the customized page data to a Raw Data file.

For more information, see "Custom views" on page 1086.

Application Details

After you drill down from the **All My Applications** dashboard to a detailed dashboard for a managed system instance, the Attribute Details tab is displayed for you to create and manage custom historical line charts and tables that can be saved. You can save more chart or table pages for your viewing only or to be shared with all users in the same environment.

For more information, see "Creating a custom chart or table page" on page 1068.

APIs

Cloud APM APIs are available for managing your environment such as to assign users roles and to create thresholds. For more information, see "Exploring the APIs" on page 1049.

Role-based access control

In Cloud APM, a role is a group of permissions that control the actions you can take. Use the Role Based Access Control feature to create customized roles, which are the basis of security. The following four predefined roles are also available: Role Administrator, Monitoring Administrator, System Administrator, and Monitoring User. You can assign users to both customized roles or predefined roles, and users can be assigned to multiple roles. You can assign permissions to customized roles, or you can assign more permissions to existing default roles. Permissions are cumulative. A user is assigned all the permissions for all the roles they are assigned to.

You can assign the View permission and the Modify permission to individual applications, system resource groups, and custom resource groups. For example, if you are a member of a role that has View permission for an application, you can view all the supporting components within that application.

You can assign the View permission and Modify permission to system administration tasks. For example, if you are a member of a role that has View permission for Advanced Configuration, you can make and save changes in the **Advanced Configuration** window.

For more information, see "Roles and permissions" on page 982.

Historical Reporting

Reports are available for data that is collected by the WebSphere Applications agent, the Response Time Monitoring Agent, and the Synthetic Playback agent. Transaction tracking is required for Response Time Monitoring agent reports (Not available with Cloud APM, Base) For report descriptions, see "Reports" on page 1097.

Agent Builder

Build custom agents to monitor any platform or technology. See https://www.ibm.com/support/knowledgecenter/SSMKFH/com.ibm.apmaas.doc/install/agent_builder_guide.htm.

Database resource monitoring

Coverage varies by offering. See "Capabilities" on page 56 for the names of the databases that can be monitored,

Infrastructure resource monitoring

Use resource monitoring agents to monitor hypervisors, storage, and networks. Coverage varies by offering. See "Capabilities" on page 56.

Commercial applications resource monitoring

Use resource monitoring agents to monitor business and collaboration applications. Coverage varies by offering. See "Capabilities" on page 56.

Response time and end user experience monitoring

See what your users experience from your infrastructure to their device. Use response time monitoring to monitor the performance and availability of websites and web applications from the browser through to the database, and to monitor mobile devices. After you install the Response Time Monitoring agent on any web servers that you want to monitor, data that is collected by these agents is displayed in the Application Performance Dashboard with little or no further configuration required. Data from the Response Time Monitoring agent is used for the **End User Transactions** dashboards. In Cloud APM, Advanced you can measure response time from the Browser, and data from the Response Time Monitoring agent is also used in the **Aggregate Transaction Topology**. For more information, see "Scenario: Monitoring the IBM Java application stack " on page 91.

Transaction tracking

This feature is available with Cloud APM, Advanced. The transaction tracking feature enables topology views and instance level transaction monitoring. Transaction tracking is installed as part of the Cloud APM server. Transaction tracking is automatically enabled for some agents but must be manually enabled for others. Table 4 on page 75 provides more information about agents that support transaction tracking.

Table 4. Transaction tracking enablement for agents and data collectors

Agent or data collector	Enabled by default	How to enable
DataPower agent	✓	"Configuring transaction tracking for the DataPower agent" on page 240
IBM Integration Bus agent	—	"Configuring transaction tracking for the IBM Integration Bus agent" on page 285 **Note:** TT is not supported if you deploy this agent on Solaris X86.

Table 4. Transaction tracking enablement for agents and data collectors (continued)

Agent or data collector	Enabled by default	How to enable
J2SE data collector	✓	"Configuring J2SE monitoring" on page 442
JBoss agent	—	"Setup the JBoss agent transaction tracking or diagnostics data collector" on page 460
Liberty data collector	✓	"Configuring the Liberty data collector in on-premises environments (Liberty V18.* and older versions)" on page 464"Configuring the Liberty data collector in IBM Cloud environment (Liberty V18.* and older versions)" on page 468
Microsoft .NET agent	—	"Enabling collection of transaction tracking and diagnostics data" on page 525
Node.js data collector	—	"Customizing the stand-alone Node.js data collector for IBM Cloud applications" on page 592 "Customizing the Node.js data collector for on-premises applications" on page 597
Response Time Monitoring agent + HTTP Server agent	—	"Planning the installation " on page 679
SAP NetWeaver Java Stack agent	—	"Enabling the collection of transaction tracking and diagnostics data" on page 763
Tomcat agent	—	"Enabling the collection of transaction tracking and diagnostics data" on page 799
WebLogic agent	—	"Configuring WebLogic monitoring" on page 813
WebSphere Applications agent	—	"Configuring the data collector interactively" on page 835 **Note:** TT is not supported if you deploy this agent on Solaris X86.
WebSphere MQ agent	—	"Configuring transaction tracking for the WebSphere MQ agent" on page 920 **Note:** TT is not supported if you deploy this agent on Solaris X86.

Data is shown in both the **Aggregate Transaction Topology** and **Transaction Instance Topology** views for all agents that support transaction tracking.

Application topology
See how all components are connected in your application environment. For more information, see "Application - Application Performance Dashboard" on page 1058.

Transaction instance topology
Visualize the path followed through your environment for each instance of a transaction. For more information, see "Transaction Instance Topology " on page 97

Availability Monitoring
IBM Cloud Availability Monitoring provides enhanced synthetic monitoring of your web applications from multiple points of presence around the world. Create synthetic tests that mimic user behavior at regular intervals. Run your tests from public points of presence, or download and deploy your own custom points of presence on local or private servers. Use the Availability Monitoring dashboard to monitor application availability, performance, and alerts by using graphs, breakdown tables, and map

views. Use waterfall analysis to identify when performance and availability issues occur, and find the reasons for those issues..

For more information about using synthetic tests, see “Availability Monitoring” on page 1024.

Deep-dive diagnostics

For specific agents, you can drill-down from summary dashboards to deep-dive diagnostics dashboards and view information about individual requests. Drill down from summary dashboards to view code-level, stack trace, and SQL query detail. Use the diagnostics dashboards to identify which requests have a problem and to debug the problematic transaction. You can also detect, diagnose, and kill hung or slow transactions that are still in progress (see the WebSphere Applications agent Reference). Table 5 on page 77 provides more information about the diagnostics agents.

Table 5. Diagnostics dashboards of agents and data collectors

Agent or data collector	Diagnostic data configured by default	Available diagnostics dashboards	How to access diagnostics dashboards	How to configure the agent or data collector to collect diagnostic data
J2SE data collector	✓	Detail, Web Modules, Request Instances, Request Summary, Request Traces	Click **Diagnose** in the **Overview** dashboard or **Web Modules** dashboard.	“Configuring J2SE monitoring” on page 442
JBoss agent	–	Diagnostic Dashboard, In-flight Requests Summary, In-flight Request Stack Trace Dashboard, JVM Garbage Collection, Heap dump, Heap Dump Comparison	Click **Diagnose**, **Inflight Requests**, **Details**, or **Heap Dump** in the **Overview** dashboard.	“Setup the JBoss agent transaction tracking or diagnostics data collector” on page 460
Liberty data collector	✓	Detail, Heap dump, Heap Dump Comparison, Memory Analysis	Click **Diagnose**, **View Heap Dump**, or **View Memory Analysis** in the **Overview** dashboard.	• “Configuring the Liberty data collector in IBM Cloud environment (Liberty V18.* and older versions)” on page 468 • “Configuring the Liberty data collector in on-premises environments (Liberty V18.* and older versions)” on page 464
Microsoft .NET agent	–	Request Instances, Request Summary, Request Traces	Click **Diagnose** in the **Overview** dashboard.	“Enabling the collection of diagnostics data by using the configdc command” on page 525

Table 5. Diagnostics dashboards of agents and data collectors (continued)

Agent or data collector	Diagnostic data configured by default	Available diagnostics dashboards	How to access diagnostics dashboards	How to configure the agent or data collector to collect diagnostic data
Node.js agent	✓	GC Details, Request Instances, Request Summary, Request Traces	Click **Diagnose** in the **Overview** dashboard.	"Configuring the Node.js agent" on page 585
Node.js data collector	✓	GC Details, Slowest Requests Detail, Request Instances, Request Traces	Click **Diagnose** or **GC Details** in the Overview dashboard.	• "Configuring the stand-alone Node.js data collector for IBM Cloud(formerly Bluemix) applications" on page 590 • "Configuring the stand-alone Node.js data collector for on-premises applications" on page 595
Python data collector	✓	Slowest Requests Details, Request Instances Detail, Request Traces Detail, Python Thread Details, Python Garbage Collection, Python Heap Details	Click **Diagnose**, **Threads Detail**, or **Memory Detail** in the **Overview** dashboard.	• "Configuring the Python data collector for IBM Cloud applications" on page 664 • "Configuring the Python data collector for on-premises applications" on page 669
Ruby agent	–	Request Summary Detail, Sampled Request Instances, Request Traces	Click **Diagnose** in the **Overview** dashboard.	"Configuring Ruby monitoring" on page 708
Ruby data collector	✓	Request Instances, Request Summary, Request Traces	Click **Diagnose** in the **Overview** dashboard.	"Configuring the Ruby data collector for IBM Cloud applications" on page 715
SAP NetWeaver Java Stack agent	✓	Request Instances, Request Summary, Request Traces	Click **Diagnose** in the **Overview** dashboard.	"Enabling the collection of transaction tracking and diagnostics data" on page 763

Table 5. Diagnostics dashboards of agents and data collectors (continued)

Agent or data collector	Diagnostic data configured by default	Available diagnostics dashboards	How to access diagnostics dashboards	How to configure the agent or data collector to collect diagnostic data
Tomcat agent	—	Request Instances, Request Summary, Request Traces	Click **Diagnose** in the **Overview** dashboard.	“Enabling the collection of transaction tracking and diagnostics data” on page 799
WebLogic agent	—	Diagnostic Dashboard, In-flight Requests Summary, In-flight Request Stack Trace Dashboard, JVM GC Detail, Heap Dump, Heap Dump Comparison	Click **Diagnose**, **View Requests**, **Details**, or **Heap Dump** in the **Overview** dashboard.	“Configuring WebLogic monitoring” on page 813
WebSphere Applications agent	—	Diagnostics, Request Instance, Request Sequence, In-flight Requests Summary, In-flight Request Stack Trace, Heap dump, Heap Dump Comparison, Memory Analysis	Click **Diagnose**, **View Requests**, **View Heap Dump**, or **View Memory Analysis** in the **Overview** dashboard. The **View Memory Analysis** button works only after memory leak monitoring is enabled.	• “Configuring the data collector with the simple configuration utility” on page 832 • “Enabling memory leak monitoring” on page 870

The **Diagnose** button is enabled only when deep-dive diagnostics is configured for your agent and you are a member of the Role Administrator role, Monitoring Administrator role, or some other custom role that has view permission for Diagnostics Dashboards.

Thresholds

With thresholds, you can detect specific application behaviors and conditions based on actively monitored definitions. Predefined thresholds are available for each agent and you can define new thresholds for monitoring. For more information, see “Threshold Manager” on page 967.

When you have event forwarding configured, events are sent to the EIF receiver. You can use the default mapping between thresholds and events forwarded to the event server or customize how thresholds are mapped. For more information, see “Customizing an event to forward to an EIF receiver” on page 972.

In the **Application Performance Dashboard**, after you select an application, the **Events** tab is displayed. The **Events** tab shows the open events for the current application. You can drill down to detailed dashboards with performance metrics to help you determine the cause of the event. For more information, see “Event Status” on page 1083.

Resource groups

Managed systems in your monitored enterprise can be categorized by their purpose. Such managed systems often have the same threshold requirements. Use the Resource Group Manager to organize monitored systems into groups that you can assign eventing thresholds to. For more information, see “Resource Group Manager” on page 962.

Getting started page

After you log in to the Cloud APM console, you are presented with a Getting Started page. Click any of the **User Tasks** or **Administrator Tasks** to link to a scenario-based tour or video demonstration. "Start now" links take you directly to the feature, such as the Threshold Manager. **Community Resources** links go to **Frequently Asked Questions**, the Cloud APM forum, and more.

Extra features are available through integration with other products and components. For more information, see “Integration ” on page 80 and more details in Chapter 8, “Integrating with other products and components,” on page 929.

Integration

Extra features are provided through integration with other products and components: Tivoli Monitoring, OMEGAMON, Netcool/OMNIbus, Operations Analytics - Log Analysis, Operations Analytics - Predictive Insights, Alert Notification, Control Desk, IBM Cloud, and Agent Builder.

IBM Tivoli Monitoring

Agent coexistence is supported. You can install IBM Cloud Application Performance Management agents on the same computer where IBM Tivoli Monitoring agents are installed. However, both agent types cannot be installed in the same directory. For more information, see “Cloud APM agent and Tivoli Monitoring agent coexistence” on page 930.

If your environment has both IBM Tivoli Monitoring and Cloud APM products (cloud, on premises, or both), you can install the IBM Cloud Application Performance Management Hybrid Gateway to provide a consolidated view of managed systems from both environments. For more information, see “Hybrid Gateway” on page 933. For the list of supported Tivoli Monitoring agents, see “Supported Tivoli Monitoring and OMEGAMON agents” on page 934..

IBM OMEGAMON

The z Systems Extension Pack connects one or more OMEGAMON agents that are running on your z Systems mainframe to Cloud APM. By using the z Systems Extension Pack and the Hybrid Gateway to connect your deployed OMEGAMON agents to Cloud APM, you can view monitoring data and events for your OMEGAMON application components in the Cloud APM console.

For more information, see “Integrating with OMEGAMON” on page 944.

IBM Netcool/OMNIbus

You can forward your events from Cloud APM into your on-premises Netcool/OMNIbus event manager. For more information, see “Integrating with Netcool/OMNIbus” on page 945.

IBM Operations Analytics - Log Analysis

When your environment includes IBM Operations Analytics - Log Analysis, you can bring together application log data and performance data to help find the root cause of problems that are experienced by your applications. You can search through log data that is associated with your applications to find the cause of a problem, such as slowness or a failure. For more information, see “Integrating with Operations Analytics - Log Analysis” on page 952.

IBM Operations Analytics - Predictive Insights

Operations Analytics - Predictive Insights analyzes data and learns the normal behavior of a system. It creates a performance model and uses it to detect or forecast behavior outside the modeled range, and generates alarms when anomalous behavior occurs. You can add Operations Analytics - Predictive Insights to your Cloud APM subscription. You can then view anomalies in the Application Performance Dashboard and drill down to the Operations Analytics - Predictive Insights user interface to view more details. For more information, see “Integrating with Operations Analytics - Predictive Insights” on page 952.

IBM Cloud

You can view monitoring information for your applications within the IBM Cloud environment by using the stand-alone data collectors. The data collectors enable the integration of monitoring capabilities with IBM Cloud by transferring resource and deep-dive diagnostics monitoring data about your IBM Cloud applications to the Cloud APM server. The Cloud APM server receives and processes monitoring information that is gathered by the data collectors. The following types of IBM Cloud applications can be monitored:

- Liberty applications
- Node.js applications
- Python applications
- Ruby applications

After configuring a data collector, you can view monitoring data on the Cloud APM console. For more information, see "General procedure for configuring data collectors" on page 184.

IBM Alert Notification

If you are using IBM Cloud Application Performance Management, IBM Alert Notification is automatically integrated for you. Alert Notification is an easy to use, simple notification system that gives IT staff instant notification of alerts for issues in your IT operations environment. Data that is received from agents provides the source of the alerts. After you activate Alert Notification, connect it to a Cloud APM instance. As a stand-alone package, you can integrate Alert Notification with any on-premises monitoring tool that can implement and start a REST API. Supported tools include IBM Tivoli Netcool/OMNIbus. For more information, see Integrating with Alert Notification.

IBM Control Desk

Integration with IBM Control Desk is available by submitting a support ticket to IBM Support. You can configure Cloud APM events to automatically open tickets in IBM Control Desk. Go to IBM Support and select **Subscription**. For more information about the details that are required by support to enable them to configure this integration, see "Integrating with Control Desk" on page 954.

IBM Agent Builder

You can use Agent Builder to build custom agents for any technology. For more information, see https://www.ibm.com/support/knowledgecenter/SSMKFH/com.ibm.apmaas.doc/install/agent_builder_guide.htm.

Documentation

You can find information for IBM Cloud Application Performance Management in the IBM Knowledge Center and Cloud APM console.

IBM Knowledge Center

Cloud APM in the IBM Knowledge Center is the official source of technical information for the product.

User interface help

When you are logged in to the Cloud APM console or exploring the Guided Demo, you can access the help system:

- Click **Help Contents** from the navigation bar. ? Help menu.
- Click ? in the Application Performance Dashboard banner.
- Click the **Learn more** link in the System Configuration pages.
- Click ? in a dashboard widget.

IBM Cloud Application Performance Management Forum and dwAnswers

The Cloud Application Performance Management Forum and dwAnswers contain technical discussions of product issues, including troubleshooting problems and solutions.

Information is also available at the following websites:

Software Product Compatibility Reports (SPCR) tool

You can use the SPCR tool to generate various types of reports that are related to offering and component requirements. Search for one of the Cloud Application Performance Management offering names or for IBM Cloud Application Performance Management - Agents.

IBM Marketplace

Resources such as video demonstrations and FAQs are available in IBM Marketplace.

IBM API Explorer

For documentation about the Cloud APM APIs, see "Exploring the APIs" on page 1049.

IBM Terminology

The IBM Terminology website contains terminology that is relevant to IBM products and consolidated in one convenient location.

IBM Redbooks®

The IBM Redbooks website contains Redbooks publications, Redpapers, and Redbooks technotes that provide information about products from platform and solution perspectives.

Conventions used in the documentation

Several conventions are used in the documentation for special terms, actions, commands, paths that are dependent on your operating system, and for platform-specific and product-specific information.

Typeface conventions

The following typeface conventions are used in the documentation:

Bold

- Lowercase commands, mixed-case commands, parameters, and environment variables that are otherwise difficult to distinguish from the surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip:**)
- Keywords and parameters in text

Italic

- Citations (examples: titles of publications, diskettes, and CDs)
- Words and phrases defined in text (example: a nonswitched line is called a *point-to-point line*)
- Emphasis of words and letters (example: The LUN address must start with the letter *L*.)
- New terms in text, except in a definition list (example: a *view* is a frame in a workspace that contains data.)
- Variables and values you must provide (example: where *myname* represents...)

`Monospace`

- Examples and code examples
- File names, directory names, path names, programming keywords, properties, and other elements that are difficult to distinguish from the surrounding text
- Message text and prompts
- Text that you must type
- Values for arguments or command options

`Bold monospace`

- Command names, and names of macros and utilities that you can type as commands
- Environment variable names in text
- Keywords
- Parameter names in text: API structure parameters, command parameters and arguments, and configuration parameters
- Process names
- Registry variable names in text
- Script names

Operating system-dependent variables and paths

The direction of the slash for directory paths might vary in the documentation. Regardless of what you see in the documentation, follow these guidelines:

- Linux AIX Use a forward slash (**/**).
- Windows Use a backslash (****).

The names of environment variables are not always the same in Windows and AIX. For example, %TEMP % in Windows is equivalent to $TMPDIR in AIX or Linux.

For environment variables, follow these guidelines:

- Linux AIX Use **$***variable*.
- Windows Use **%***variable***%**.

Windows If you are using the bash shell on a Windows system, you can use the AIX conventions.

Installation directory variable and paths for agents

install_dir is the installation directory for the agents. The default location depends on the operating system:

- Windows `C:\IBM\APM`
- Linux `/opt/ibm/apm/agent`
- AIX `/opt/ibm/apm/agent`

Chapter 4. Planning your deployment

To ensure that your IBM Cloud Application Performance Management deployment is successful, planning is critical.

System requirements

For the IBM Cloud Application Performance Management agents and data collectors, various operating systems are supported and each of these components has specific requirements.

Time zone

Use Network Time Protocol (NTP) on managed systems to ensure that the time is accurate. Setting the time to match the physical location of the servers (such as UTC-03:00 for Brasilia and UTC +06:30 for Yangon) helps ensure accurate time stamps for events and transactions. Agents report data in the APM UI in the local time of the user.

Cloud APM agent and data collector requirements

Obtain information about the requirements for each monitoring agent and stand-alone data collector that you plan to install.

Cloud APM agent and data collectors in general are hypervisor transparent, which means they can be installed and deployed on any supported operating systems regardless of the hypervisors that the operating systems are hosted on, such as Hyper-V, IBM PowerVM, KVM, VMWare ESX, and so on.

For the agent and data collector requirements, generate a report from for the selected agent or data collector:

1. Open the IBM Software Product Compatibility Reports for IBM Cloud Application Performance Management - Agents V8.1.
2. Expand the **Report filters** twisty and click the **Edit** button.
3. Edit the report filters for the agent or data collector that you want to review and click **Apply** to generate the report.

For example, if you want to see the requirements for the Cisco UCS agent , clear all the **Agent or Client** check boxes and select the Monitoring Agent for Cisco UCS check box before you click **Apply**.

For information about supported browsers, see the IBM Cloud Application Performance Management Detailed System Requirements report.

The local computer system where the agent is installed must support UTF-8 encoding if the agent sends globalized data to the Cloud APM server.

Default ports used by agents and data collectors

Various ports are used for communication between the Cloud APM component and the application or system (either local or remote) that is being monitored. In most cases, default ports are provided to facilitate configuration. Most defaults can be customized by using configuration parameters.

Table 6 on page 86 lists the default ports that are used by the Cloud APM agents and data collectors to communicate with the applications or systems that they are monitoring. N/A in the table indicates one of the following situations:

- The agent or data collector does not use any port to communicate with the monitored application or system.
- The port used for communication is determined by configuration of the monitored application.
- Ports used by the agent or data collector are dynamically assigned and no static defaults are provided.
- All ports to be used must be specified by the user and no defaults are provided.

Table 6. Default ports used by agents and data collectors

Agents and data collectors	Default ports	Configurable	Local	Remote
Amazon EC2 agent	• TCP port 80 (for HTTP) • TCP port 443 (for HTTPS)	N/A	Yes	No
Amazon ELB agent	• TCP port 80 (for HTTP) • TCP port 443 (for HTTPS)	N/A	No	No
Azure Compute agent	• TCP port 80 (for HTTP) • TCP port 443 (for HTTPS)	No	No	No
Cassandra agent	7199 (for JMX server, local and remote)	Yes	Yes	Yes
Cisco UCS agent	• TCP port 80 (for HTTP) • TCP port 443 (for HTTPS)	No	Yes	No
Citrix VDI agent	For PowerShell calls: • 5985 (for HTTP) • 5986 (for HTTPS)	Yes	Yes	Yes
Db2 agent	• 50000 (default port of Db2 server) • Remote Monitoring supported: Uses port number given by the user while cataloging the remote server instance.	Yes	Yes	Yes
DataPower agent	5550 (for connecting to remote DataPower appliance)	Yes	No	Yes
Hadoop agent	• Local monitoring: **CP_PORT** environment variable value • Remote monitoring: – 50070 (Standby Namenode) – 50090 (Secondary Namenode) – 8088 (ResourceManager) – 19888 (JobHistory Server) – 8080 (Ambari)	Yes	Yes	Yes
HMC Base agent	12443 (for downloading SDK from HMC)	No	Yes	No
HTTP Server agent	HTTP server might be configured to different port, but the agent itself has no default port.	N/A	Yes	No
IBM Cloud agent	Outgoing connection to api.softlayer.com port 443.	N/A	No	Yes
IBM Integration Bus agent	N/A	N/A	Yes	No

Table 6. Default ports used by agents and data collectors (continued)				
Agents and data collectors	**Default ports**	**Configurable**	**Local**	**Remote**
Internet Service Monitoring	For databridge: • 9510 • 9520	Yes	No	Yes
DataStage agent	• 9443 (WAS HTTPS port) • 50000 (Database JDBC port) • 1433 (Microsoft SQL) • 1521 (Oracle)	Yes	Yes	Yes
J2SE data collector	N/A	N/A	No	No
JBoss agent	Varies according to the version of the JBoss server: • 9990 • 9994 • 9999	No	Yes	No
Liberty data collector	N/A	N/A	No	No
Linux KVM agent	• 8080 (for HTTP) • 8443 (for HTTPS)	Yes	Yes	No
Linux OS agent	22 (for remote log monitoring with SSH)	Yes	Yes	No
MariaDB agent	3306	Yes	Yes	Yes
Microsoft Active Directory agent	The port number depends on the listener setting for monitoring usage.	N/A	Yes	Yes
Microsoft Cluster Server agent	N/A	N/A	No	No
Microsoft Exchange Server agent	N/A	N/A	No	No
Microsoft Hyper-V Server agent	N/A	N/A	No	No
Microsoft IIS agent	N/A	N/A	No	No
Microsoft .NET agent	To send transaction tracking data, port 5456 is used by default.	Yes	Yes	No
Microsoft Office 365 agent	7799 (for Skype synthetic transaction)	Yes	Yes	No
Microsoft SharePoint Server agent	1433 (for SQL server)	No	Yes	Yes
Microsoft SQL Server agent	1433 (default of SQL server)	Yes (by **COLL_PORT**)	Yes	No

Table 6. Default ports used by agents and data collectors (continued)

Agents and data collectors	Default ports	Configurable	Local	Remote
MQ Appliance agent	• 162 (for receiving SNMP events) • 5554 (for connecting to MQ Appliances)	Yes	Yes	Yes
MongoDB agent	• 27017 (for single instance) • 27019 (for cluster)	Yes	Yes	No
MySQL agent	3306 (for JDBC connection)	Yes	Yes	Yes
NetApp Storage agent	For remote monitoring: • 8088 • 8488 • 443 • 8443	No	No	Yes
Node.js agent	63336	Yes	Yes	No
Node.js data collector	N/A	N/A	No	No
OpenStack agent	5000 (for connecting OpenStack identity service)	Yes	No	Yes
Oracle Database agent	1521 (for SQL connection)	Yes	Yes	No
PHP agent	• Apache connection • Port number is based on the Apache configuration	Yes	Yes	No
PostgreSQL agent	5432 (for JDBC connection)	Yes	Yes	Yes
Python data collector	N/A	N/A	No	No
RabbitMQ agent	Port number where the RabbitMQ management plug-in is enabled (local and remote): 15672	Yes	Yes	Yes
Response Time Monitoring Agent	• Package analyzer model monitors HTTP transactions at port 80. • HTTP server model monitors all ports.	Yes	Yes	No
Ruby agent	Dynamically generated	N/A	Yes	No
Ruby data collector	N/A	N/A	No	No
SAP agent	33*nn* (where *nn* is the SAP instance number)	No	Yes	No
SAP HANA Database agent	Default: 30013. Range: 30013-39913.	Yes	Yes	No
SAP NetWeaver Java Stack agent	Default: 50004. Range: 50004-59904.	Yes	Yes	No
Siebel agent	N/A	N/A	Yes	No

Table 6. Default ports used by agents and data collectors (continued)

Agents and data collectors	Default ports	Configurabl e	Local	Remote
Skype for Business Server agent (formerly known as Microsoft Lync Server agent)	• Business server default port 5061 • SQL server port 1433 (local or remote depending on environment).	No	Yes	No
Sterling Connect Direct agent	1363	Yes	No	Yes
Sterling File Gateway agent	50000 The IBM B2B Integrator REST API port number and Database port number are both required and are configurable.	Yes	Yes	Yes
Sybase agent	5000	N/A	Yes	No
Synthetic Playback agent	• 4444 (for connecting internal selenium server) • Remote ports are specified in the http URL of monitored websites, typically HTTP 80 and HTTPS 443	No	Yes	No
Tomcat agent	8686 (for Tomcat MBean server)	Yes (by JMX port)	Yes	No
UNIX OS agent	22 (for remote log monitoring with SSH)	Yes	Yes	No
VMware VI agent	• 443 (for remote monitoring) • 80 (for local monitoring)	No	Yes	Yes
WebLogic agent	7003 (JMX Management HTTP traffic)	Yes	Yes	No
WebSphere Applications agent	• 63335 (for V8 monitoring agent) • 63336 (for V6 monitoring agent) • 63355 (for resource monitoring) • 5457 (for Transaction Framework Extension)	Yes	Yes	No
WebSphere Infrastructure Manager agent	N/A	N/A	Yes	No
WebSphere MQ agent	The port number depends on the listener setting for monitoring usage.	N/A	No	Yes
Windows OS agent	22 (for remote log monitoring with SSH)	Yes	Yes	No

Scenarios

Depending on the complexity of your environment, you must install different agents to monitor different components. Use these deployment scenarios to help you understand what you must install where to get the best results from IBM Cloud Application Performance Management.

Scenario: Monitoring IBM API Connect

You can monitor and troubleshoot your IBM API Connect environment by using APM agents and data collectors.

The Cloud APM product helps you manage the performance and availability of your API Connect environment. By using Cloud APM agents and data collectors, you are provided with visibility and control of both the API Connect infrastructure and the application APIs, ensuring optimal performance and efficient use of resources. When you encounter performance issues within the API Connect environment, the Cloud APM product can assist you in detecting, diagnosing, and isolating them.

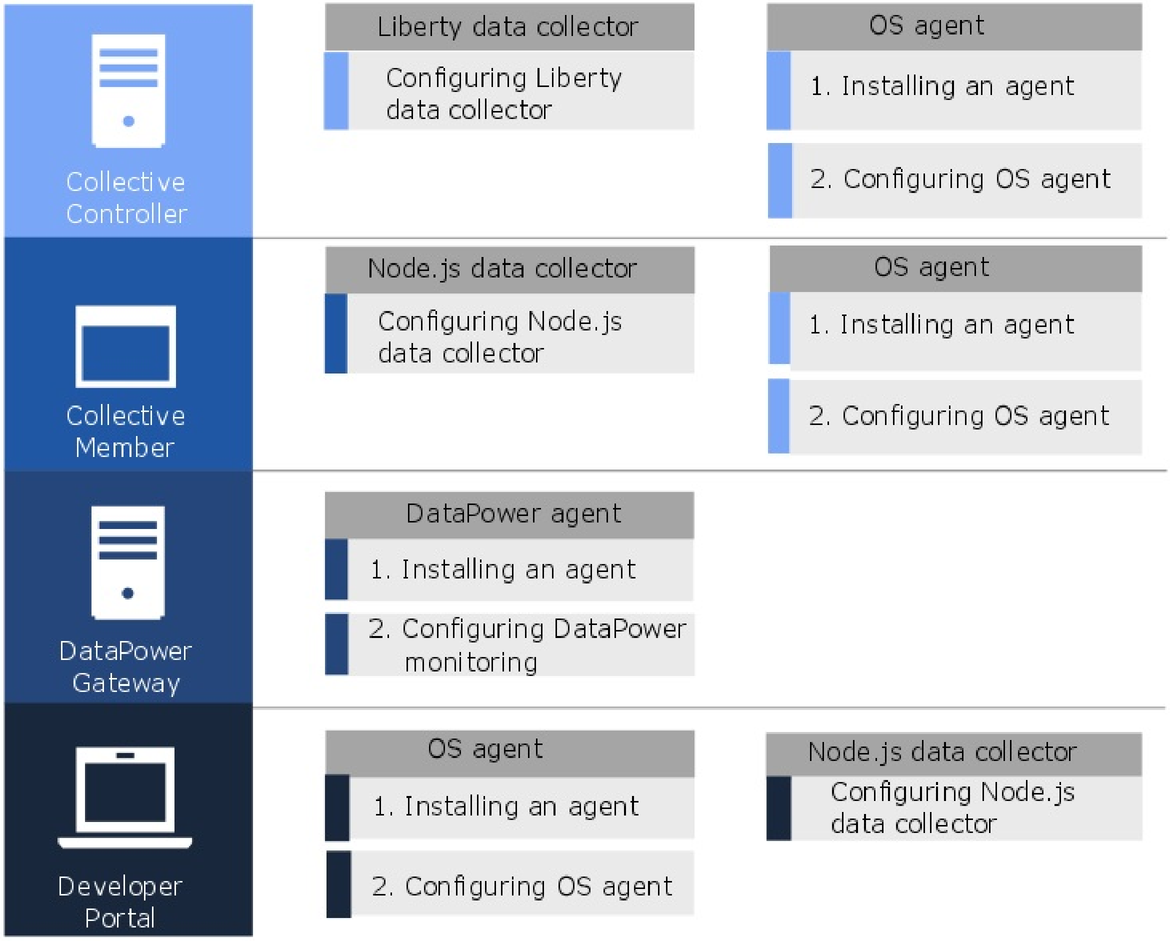
For example, you can install the OS agents on all applicable systems. Use the OS agents to collect and analyze server-specific performance, including CPU performance, disk I/O and utilization, process availability and performance, and network performance. In addition, the OS agents can be configured to monitor the key API Connect logs and system logs.

If you have other middleware products deployed, the transaction tracking feature, which is installed as part of the Cloud APM server, can provide you with topology views to see transaction tracking information for the middleware products and the services they expose and troubleshoot when problems arise.

The following picture shows API Connect components and the corresponding Cloud APM agents and data collectors that can monitor them. To enable these agents and data collectors, complete installation and configuration tasks that are listed under the agent and data collector name. Click the rectangular boxes in the picture that contains the task name to go to the installation or configuration tasks.

Note:

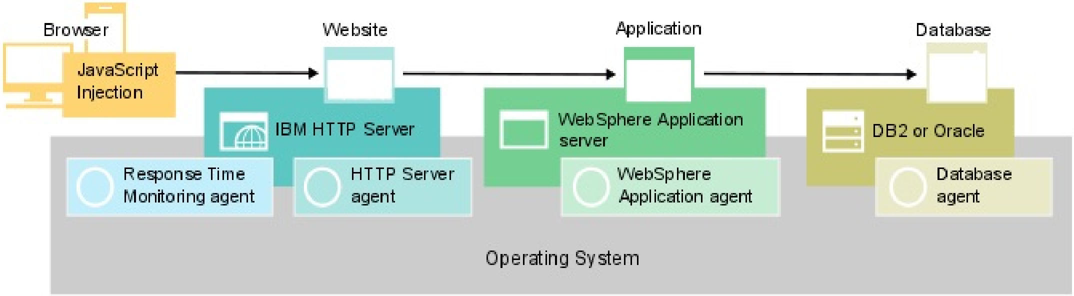
- Install a Node.js data collector to each published IBM API Connect application on the collective member.
- When monitoring the DataPower Gateway, the DataPower agent runs remotely from the DataPower appliance.



1. "Configuring the Liberty data collector in on-premises environments (Liberty V18.* and older versions)" on page 464
2. Chapter 6, "Installing your agents," on page 121
3. "Configuring OS monitoring" on page 624
4. "Configuring the stand-alone Node.js data collector for on-premises applications" on page 595
5. Chapter 6, "Installing your agents," on page 121
6. "Configuring OS monitoring" on page 624
7. Chapter 6, "Installing your agents," on page 121
8. "Configuring DataPower monitoring" on page 230
9. Chapter 6, "Installing your agents," on page 121
10. "Configuring OS monitoring" on page 624
11. "Configuring the stand-alone Node.js data collector for on-premises applications" on page 595

Scenario: Monitoring the IBM Java application stack

You can monitor and troubleshoot the IBM Java application stack to see transaction monitoring information from the browser through to the database, including resource monitoring from individual components. The IBM Java application stack includes the IBM HTTP Server, the WebSphere Application Server, and the IBM Db2 or Oracle database.



Monitoring the IBM Java application stack

To monitor the IBM Java application stack, install the agents that are listed for each component in the order given.

Optionally, if you also want to monitor the system, install OS agents on all components.

For the web server, complete the following steps:

1. Install the HTTP Server agent.

   Fast path: This installation also installs IBM HTTP Server Response Time module and automatically configures JavaScript injection.
2. Configure the HTTP Server agent installation..
3. Install the Response Time Monitoring agent.

For the application server, install the WebSphere Applications agent.

For the database, install the Oracle Database agent or Db2 agent, depending on your database.

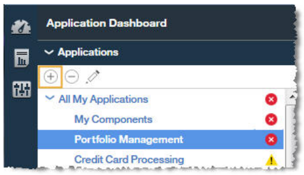
Adding web applications to the Application Performance Dashboard

Add the web applications that you want to monitor to the Application Performance Dashboard.

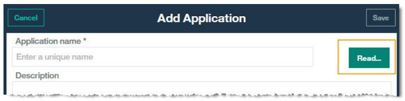
Procedure

To add web applications, complete the following steps:

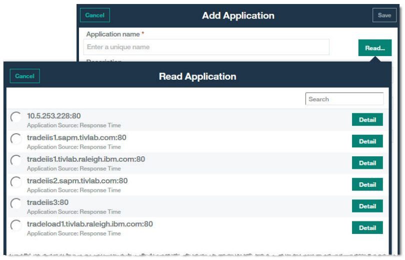
1. In the Application Performance Dashboard, click **Add Application**.

2. Click **Read** to open a list of discovered applications.

3. Select the web application that you want to monitor.

   
   Cancel
   Add Application
   Save
   Application name *
   Enter a unique name
   Read...
   Cancel
   Read Application
   Search
   10.5.253.228:80
   Application Source: Response Time
   Detail
   tradeiis1.sapm.tivlab.com:80
   Application Source: Response Time
   Detail
   tradeiis1.tivlab.raleigh.ibm.com:80
   Application Source: Response Time
   Detail
   tradeiis2.sapm.tivlab.com:80
   Application Source: Response Time
   Detail
   tradeiis3:80
   Application Source: Response Time
   Detail
   tradeload1.tivlab.raleigh.ibm.com:80
   Application Source: Response Time
   Detail

4. Click **Save**.

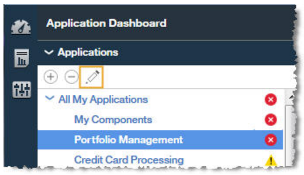
Associating the IBM Java application stack with the web application

Edit the web application to associate the WebSphere Application Server and database components that you want to monitor with it.

Procedure

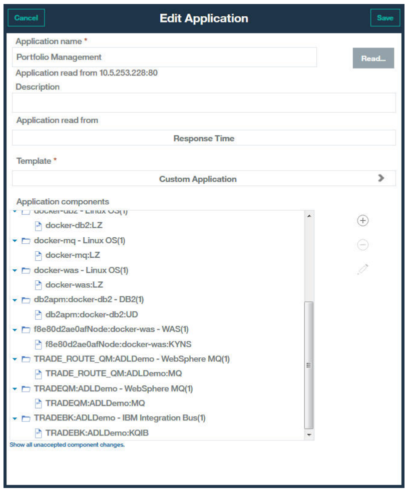
To display the components in the Java application stack, complete the following steps in the Application Performance Dashboard:

1. Select the web server and click **Edit Application**.



2. In the **Edit Application** window, click **Add components** ＋.
3. In the **Select Component** window, select WebSphere Application Server.
4. In the **Component Editor**, select the required component instances and click **Add**.

   Any detected WebSphere Application Server instances are automatically added to this list.
5. Click **Back** and repeat steps “3” on page 93 - “4” on page 93 for your database. Continue adding WebSphere Application Server and database instances until the Java application stack is complete.



Application name *
Portfolio Management
Application read from 10.5.253.228:80
Description

Application read from
Response Time

Template *
Custom Application

Application components
- docker-db2 - Linux OS(1)
  - docker-db2:LZ
- docker-mq - Linux OS(1)
  - docker-mq:LZ
- docker-was - Linux OS(1)
  - docker-was:LZ
- db2apm:docker-db2 - DB2(1)
  - db2apm:docker-db2:UD
- f8e80d2ae0afNode:docker-was - WAS(1)
  - f8e80d2ae0afNode:docker-was:KYNS
- TRADE_ROUTE_QM:ADLDemo - WebSphere MQ(1)
  - TRADE_ROUTE_QM:ADLDemo:MQ
- TRADEQM:ADLDemo - WebSphere MQ(1)
  - TRADEQM:ADLDemo:MQ
- TRADEBK:ADLDemo - IBM Integration Bus(1)
  - TRADEBK:ADLDemo:KQIB

Show all unaccepted component changes.

6. Click **Close**, then **Save** to return to the Application Performance Dashboard.

Results

Tip: If the Aggregate Transaction Topology does not initially show the topology that you expect, wait for it to refresh and check again in a few minutes. If the topology is still not what you expect, your application might not be communicating with the expected components. Check your environment.

Viewing results of IBM Java application stack monitoring

You can view the results of IBM Java application stack monitoring in the topologies.

About this task

In the topologies, you will see transaction monitoring information from the browser to the database, including resource monitoring from individual components. The following nodes are displayed in the Aggregate Transaction Topology and Transaction Instance Topology:

- Browser, displayed only when JavaScript Injection is enabled

- HTTP server
- WebSphere Application Server
- Database

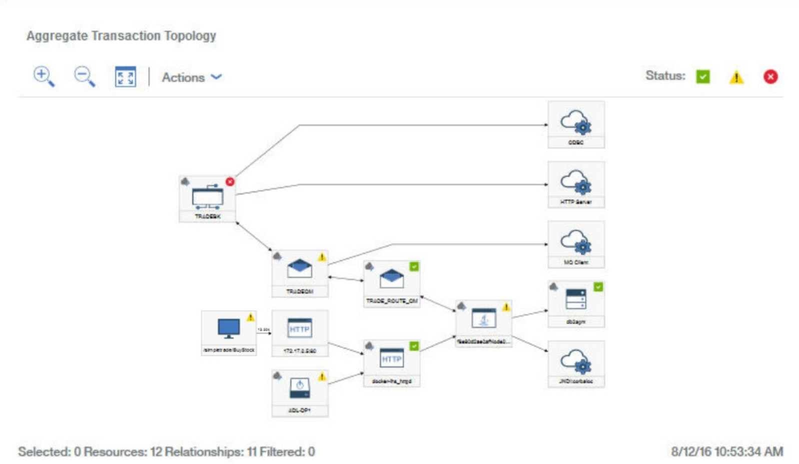
Procedure

You can link from nodes in the topology to more details about that node:

1. Hover your mouse over a node to display a window with additional information.
2. To drill down to a more detailed dashboard for the node, right-click the node and select the link.

Aggregate Transaction Topology

The Aggregate Transaction Topology is displayed in the Application summary dashboard.

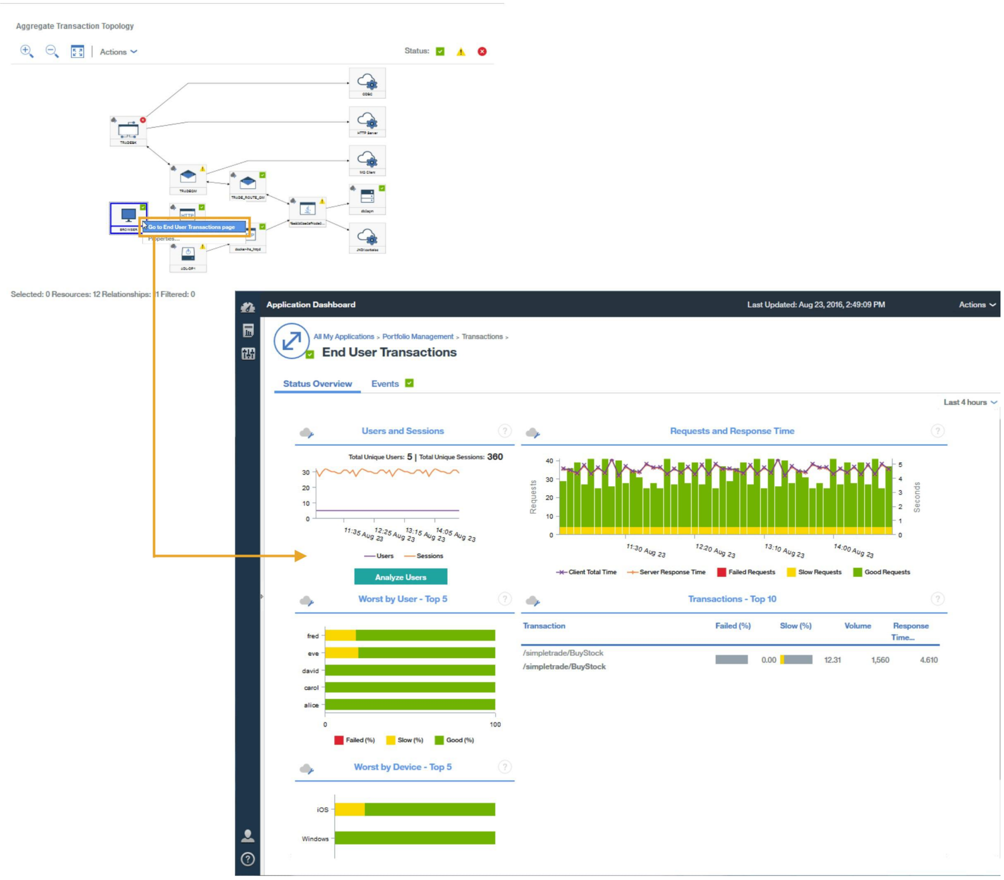


Selected: 0 Resources: 12 Relationships: 11 Filtered: 0

8/12/16 10:53:34 AM

Aggregate Transaction topologies display the following information:

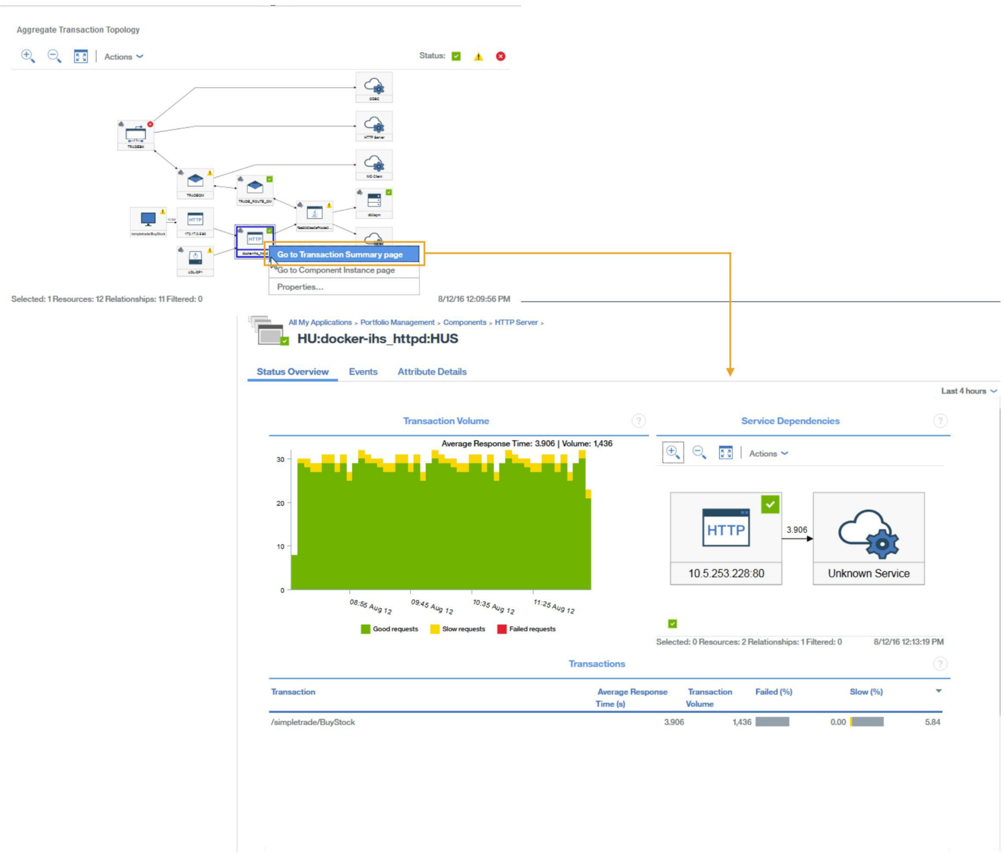
- Node for browser-based clients, drill down to the end-user experience



Transaction	Failed (%)	Slow (%)	Volume	Response Time...
/simpletrade/BuyStock /simpletrade/BuyStock	0.00	12.31	1,560	4.610

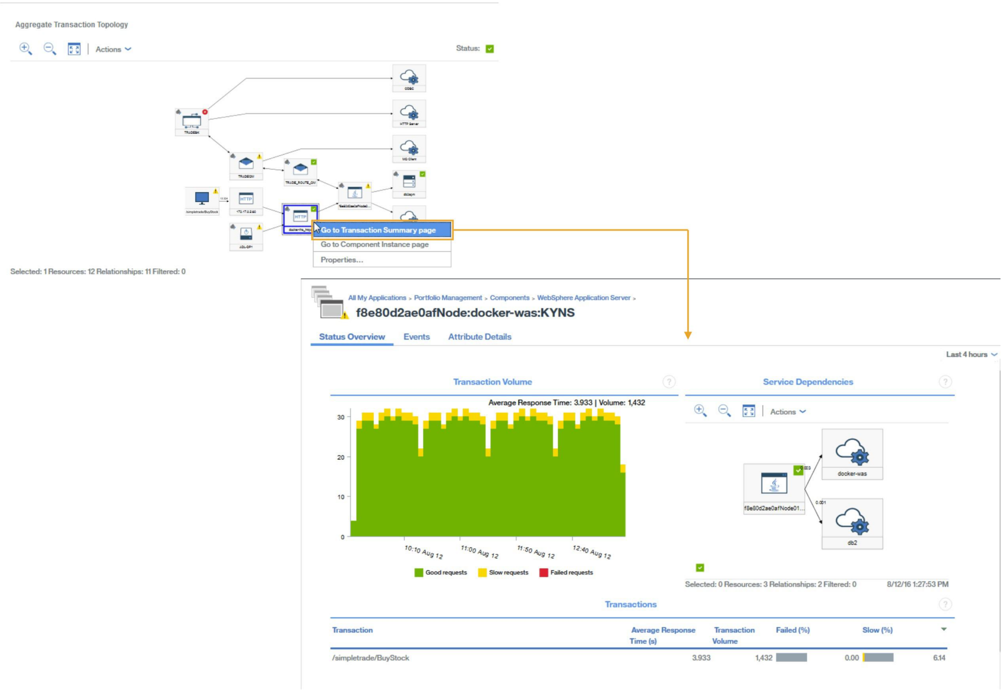
Remember: This node is displayed only when automatic JavaScript injection is measuring data from the browser.

- Nodes for HTTP-based applications, drill down to the web server resource page or a transaction summary page



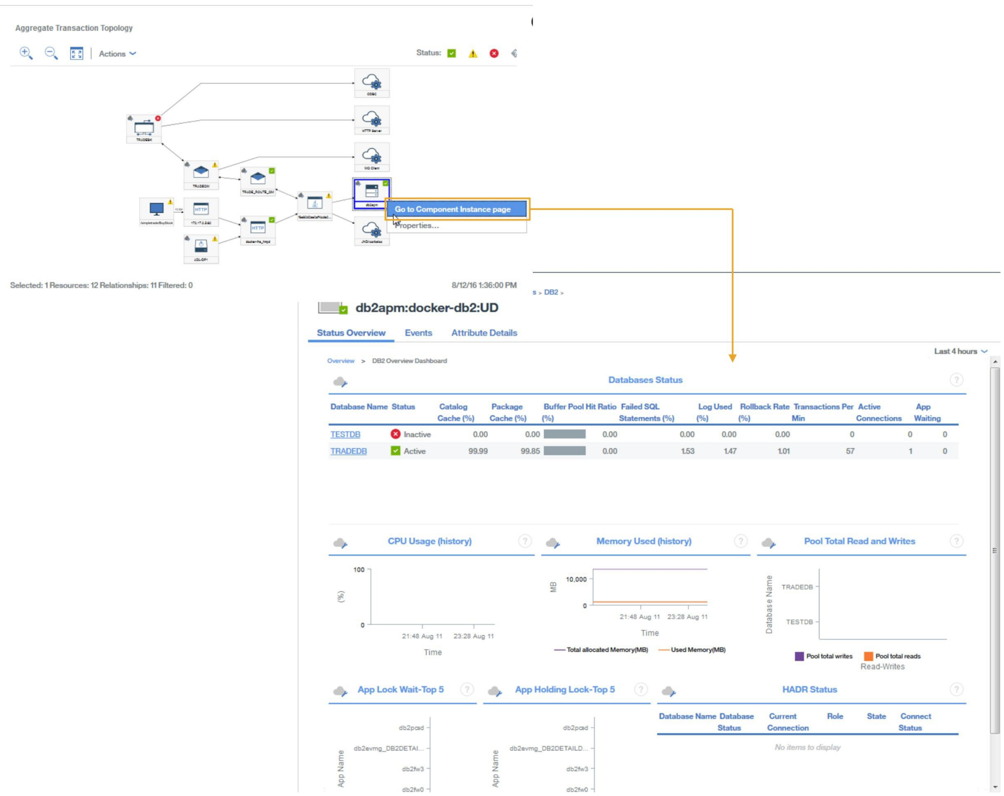
Transaction	Average Response Time (s)	Transaction Volume	Failed (%)	Slow (%)
/simpletrade/BuyStock	3.906	1,436	0.00	5.84

- Nodes for WebSphere Application Server based applications, drill down to an application resource page, or a transaction summary page


Aggregate Transaction Topology
Actions
Status:
Go to Transaction Summary page
Go to Component Instance page
Properties...
Selected: 1 Resources: 12 Relationships: 11 Filtered: 0
All My Applications > Portfolio Management > Components > WebSphere Application Server >
f8e80d2ae0afNode:docker-was:KYNS
Status Overview
Events
Attribute Details
Last 4 hours
Transaction Volume
Average Response Time: 3.933 | Volume: 1,432
30
20
10
0
10:10 Aug 12
11:00 Aug 12
11:50 Aug 12
12:40 Aug 12
Good requests
Slow requests
Failed requests
Service Dependencies
Actions
docker-was
f8e80d2ae0afNode01...
db2
Selected: 0 Resources: 3 Relationships: 2 Filtered: 0
8/12/16 1:27:53 PM
Transactions

Transaction	Average Response Time (s)	Transaction Volume	Failed (%)	Slow (%)
/simpletrade/BuyStock	3.933	1,432	0.00	6.14

- Nodes for specific database servers, drill down to a database resource page if available



Database Name	Status	Catalog Cache (%)	Package Cache (%)	Buffer Pool Hit Ratio (%)	Failed SQL Statements (%)		Log Used (%)	Rollback Rate (%)	Transactions Per Min	Active Connections	App Waiting
TESTDB	✗ Inactive	0.00	0.00	▬	0.00	0.00	0.00	0.00	0	0	0
TRADEDB	✓ Active	99.99	99.85	▬	0.00	1.53	1.47	1.01	57	1	0

Database Name	Database Status	Current Connection	Role	State	Connect Status

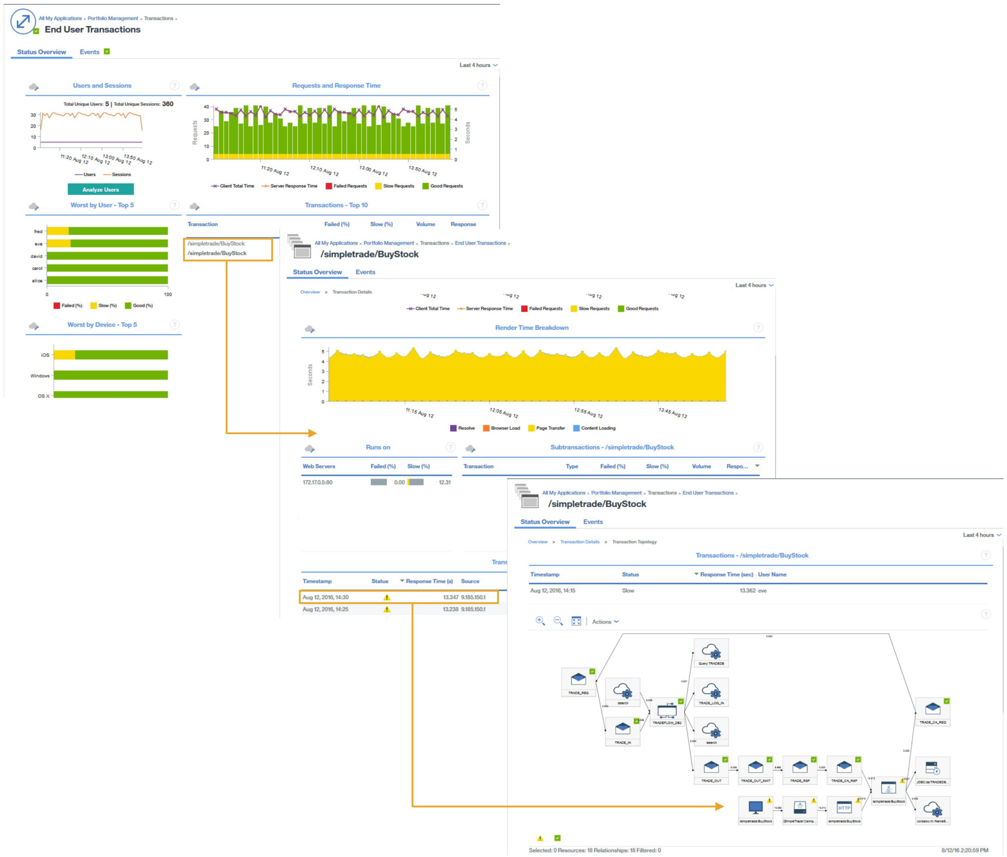
No items to display

Transaction Instance Topology

Transaction instance topologies are displayed for real end-user transactions.

Drill down from the **End User Transactions** summary through the following widgets:

1. Select a transaction in the Transactions - Top 10 table
2. Select an instance in the Transactions Instances table



Transaction instance topologies for the Java application stack display the following nodes. Click the node to display information about the node.

- Node for browser-based clients

  Remember: This node is displayed only when automatic JavaScript injection is measuring data from the browser.
- HTTP nodes, including response times from the browser
- DataPower nodes, if instrumented
- WebSphere Application Server nodes, from which you can drill down to an application resource page
- Database server nodes, from which you can drill down to a database resource status, and SQL statement diagnostic information for JDBC requests

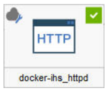
  Tip: When the topology indicates that most of the response time is spent in the database, SQL statement information is opened directly when you click **Diagnose**.

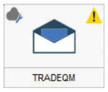
Also displayed are Gantt charts, which summarize instance timings.

Diagnosing problems in your environment

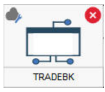
If transaction instances for one of the components in your environment are slow or failing, the affected component is assigned an appropriate status.

A node might have one of the following statuses:

- 

  Good, the node has a tick surrounded by a green square in the upper right corner
- 

  Warning, the node has an exclamation point surrounded by a yellow triangle in the upper right corner

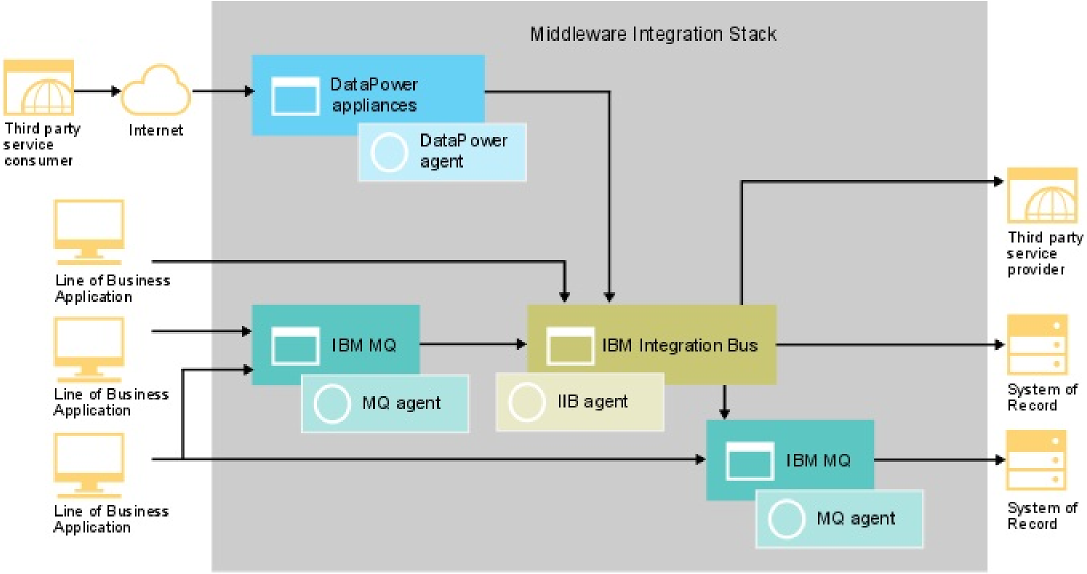
- 

Critical, the node has a red background and a cross that is encircled with red in the upper right corner

To identify the cause of the problems for these components with a warning or critical status, right-click the node and drill down to see more information about what might be causing the failures.

Scenario: Monitoring the IBM integration stack

You can monitor the IBM integration stack to see transaction tracking information for the middleware products and the services they expose and troubleshoot if any problems arise. The IBM integration stack includes IBM MQ, IBM Integration Bus, and DataPower appliance.


Middleware Integration Stack
Third party service consumer
Internet
DataPower appliances
DataPower agent
Line of Business Application
Line of Business Application
Line of Business Application
IBM MQ
MQ agent
IBM Integration Bus
IIB agent
IBM MQ
MQ agent
Third party service provider
System of Record
System of Record

Monitoring the IBM integration stack

To monitor the IBM integration stack, install the agents that are listed for each component in the order given.

Optionally, if you also want to monitor a system, install OS agents on that system.

For IBM MQ, complete the following steps:

1. Install the Monitoring Agent for WebSphere MQ.
2. Configure the WebSphere MQ agent to connect to the queue manager.
3. Enable **MQ Application Activity Trace** in the queue manager.

For IBM Integration Bus, complete the following steps:

1. Install the Monitoring Agent for IBM Integration Bus.
2. Enable IBM Integration Bus for transaction tracking.
3. Configure transaction tracking for the required IBM Integration Bus agent instances.

For DataPower appliance, complete the following steps:

1. Install the Monitoring Agent for DataPower.
2. Configure the DataPower agent to connect to the DataPower appliance.
3. Ensure that transaction tracking is enabled for the required DataPower agent instances.
4. Set up the DataPower appliance.

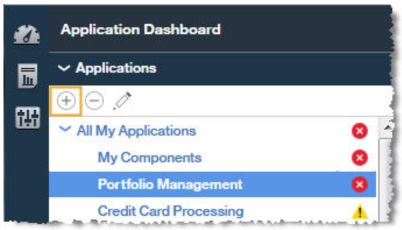
Adding middleware applications to the Application Performance Dashboard

Create an IBM integration stack application and add the IBM MQ, IBM Integration Bus, and DataPower appliance instances that you want to monitor to it.

Procedure

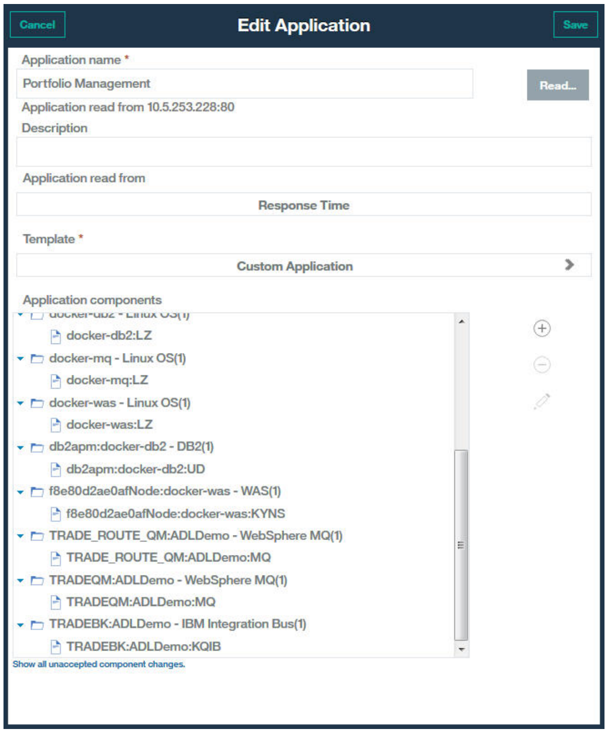
To display the components in the IBM integration stack, complete the following steps in the Application Performance Dashboard:

1. In the Application Performance Dashboard, click **Add Application**.

2. In the **Edit Application** window, add an application name and click **Add components** +.
3. In the **Select Component** window, select **IBM Integration Bus**.
4. In the **Component Editor**, select the required component instances and click **Add**.

   Any detected IBM Integration Bus instances are automatically added to this list.
5. Click **Back** and repeat steps 3 - 4 for **WebSphere MQ** and **DataPower Appliance**. Continue adding IBM Integration Bus, IBM MQ, and DataPower appliance instances until the IBM integration stack is complete.

   Application name *

   Portfolio Management

   Application read from 10.5.253.228:80

   Description

   Application read from

   Response Time

   Template *

   Custom Application

   Application components

   docker-db2 - Linux OS(1)
   docker-db2:LZ
   docker-mq - Linux OS(1)
   docker-mq:LZ
   docker-was - Linux OS(1)
   docker-was:LZ
   db2apm:docker-db2 - DB2(1)
   db2apm:docker-db2:UD
   f8e80d2ae0afNode:docker-was - WAS(1)
   f8e80d2ae0afNode:docker-was:KYNS
   TRADE_ROUTE_QM:ADLDemo - WebSphere MQ(1)
   TRADE_ROUTE_QM:ADLDemo:MQ
   TRADEQM:ADLDemo - WebSphere MQ(1)
   TRADEQM:ADLDemo:MQ
   TRADEBK:ADLDemo - IBM Integration Bus(1)
   TRADEBK:ADLDemo:KQIB

   Show all unaccepted component changes.

6. Click **Close**, then **Save** to return to the Application Performance Dashboard.

Results

Tip: If the Aggregate Transaction Topology does not initially show the topology that you expect, wait for it to refresh and check again in a few minutes. If the topology is still not what you expect, there might be a problem with your application not communicating with the expected components. Check your environment.

Viewing results of IBM integration stack monitoring

You can view the results of IBM integration stack monitoring in the topologies and middleware pages. You can also view events that are generated when a transaction violates a defined threshold.

About this task

In the topologies, you can see interactions between middleware components. The following middleware nodes are displayed in the Aggregate Transaction Topology and Transaction Instance Topology:

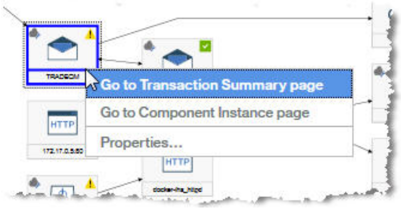
- IBM Integration Bus
- IBM MQ
- DataPower appliance

Hover your mouse over a node to display a Properties window that shows you information to explain why a node has a particular status. The status is determined by situations; the situations with a bad status are displayed.

Procedure

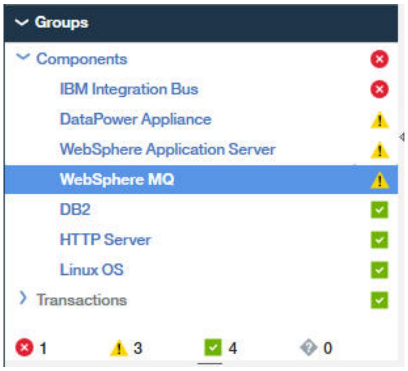
You can link from nodes in the topology to more details about that node:

1. Right-click a node.



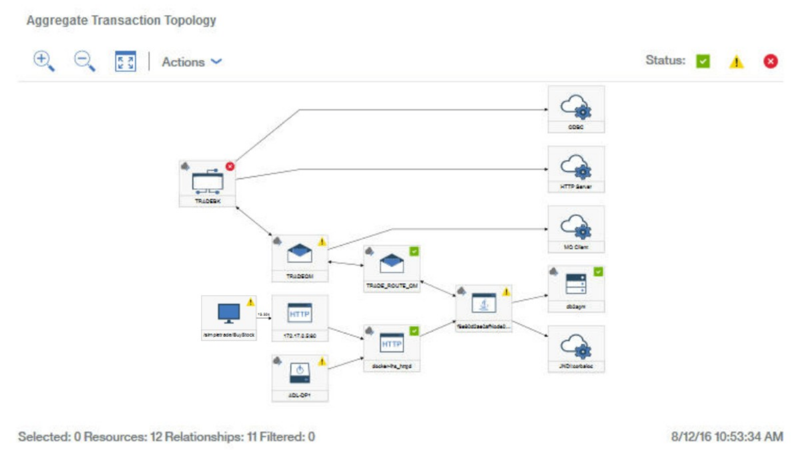
2. Select **Go to Component page** to display information about the component.
3. Select **Go to Transaction Summary page** to display information about the middleware transactions.

   Tip: Select **Groups** > **Components** > ***middleware component*** in the navigator and select a request period in the volume widget to go to the same dashboard.



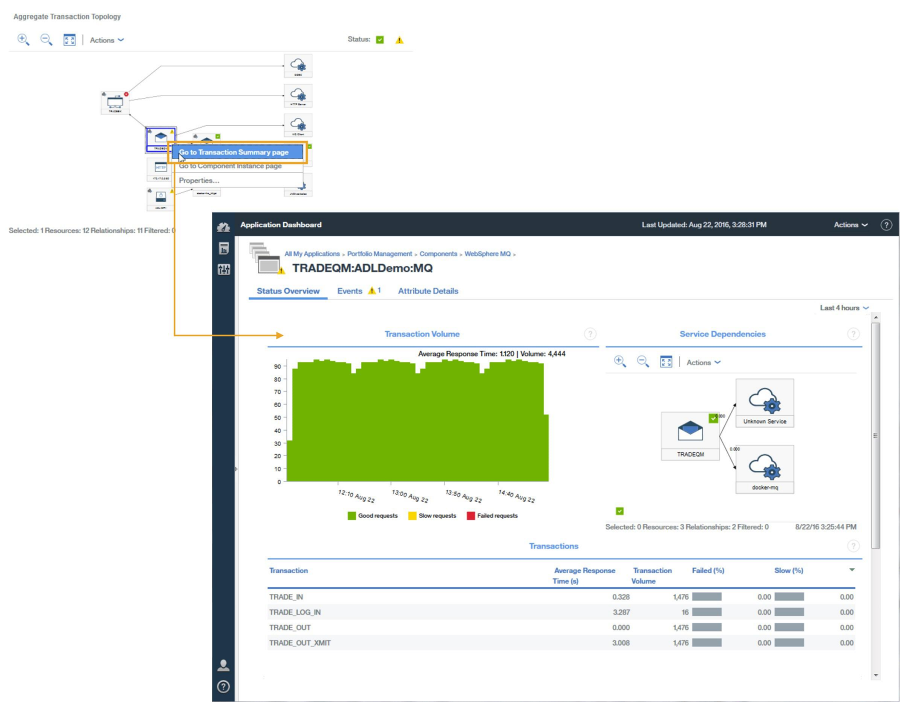
Aggregate Transaction Topology

The Aggregate Transaction Topology is displayed in the Application summary dashboard.



Aggregate Transaction topologies can display IBM MQ, IBM Integration Bus, and DataPower appliance nodes. Drill down from these nodes to more information about the middleware integration stack.

To drill down, right-click a middleware node in the Aggregate Transaction Topology and select **Go to Transaction Summary page**. Alternatively, select **Groups** > **Components** > ***middleware component*** in the navigator and select a request period in the volume widget to access the same information.

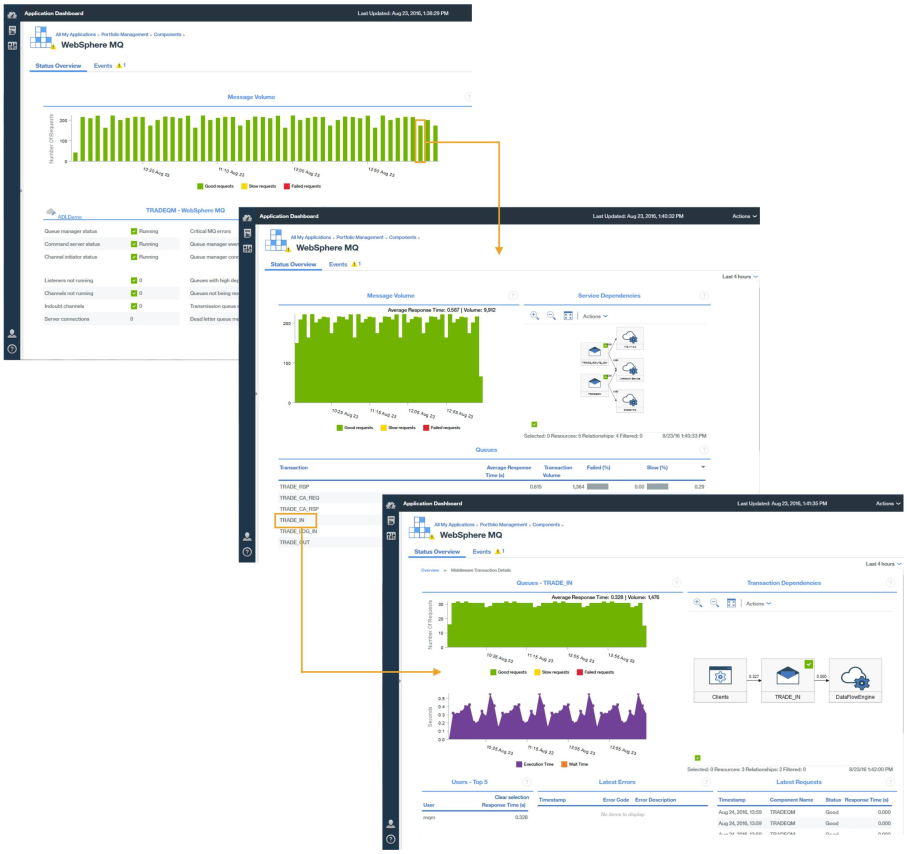

Aggregate Transaction Topology
Actions
Status:
Go to Transaction Summary page
Go to Component instance page
Properties...
Selected: 1 Resources: 12 Relationships: 11 Filtered: 0
Application Dashboard
Last Updated: Aug 22, 2016, 3:28:31 PM
Actions
All My Applications > Portfolio Management > Components > WebSphere MQ >
TRADEQM:ADLDemo:MQ
Status Overview
Events 1
Attribute Details
Last 4 hours
Transaction Volume
Average Response Time: 1.120 | Volume: 4,444
90
80
70
60
50
40
30
20
10
0
12:10 Aug 22
13:00 Aug 22
13:50 Aug 22
14:40 Aug 22
Good requests
Slow requests
Failed requests
Service Dependencies
Actions
Unknown Service
TRADEQM
docker-mq
Selected: 0 Resources: 3 Relationships: 2 Filtered: 0
8/22/16 3:25:44 PM
Transactions

Transaction	Average Response Time (s)	Transaction Volume	Failed (%)	Slow (%)
TRADE_IN	0.328	1,476	0.00	0.00
TRADE_LOG_IN	3.287	16	0.00	0.00
TRADE_OUT	0.000	1,476	0.00	0.00
TRADE_OUT_XMIT	3.008	1,476	0.00	0.00

Middleware transaction details

From the middleware Transactions Summary page, you can drill down to middleware transaction details.

To drill down to middleware transaction details for the component, complete the following steps:



1. In the middleware Transactions Summary page for the component, select a monitoring interval in the **Message or Volume** chart.
2. On the **Middleware Transaction Summary**, in the **Queues**, **Brokers**, or **Appliances** widget, select a queue, broker, or appliance.

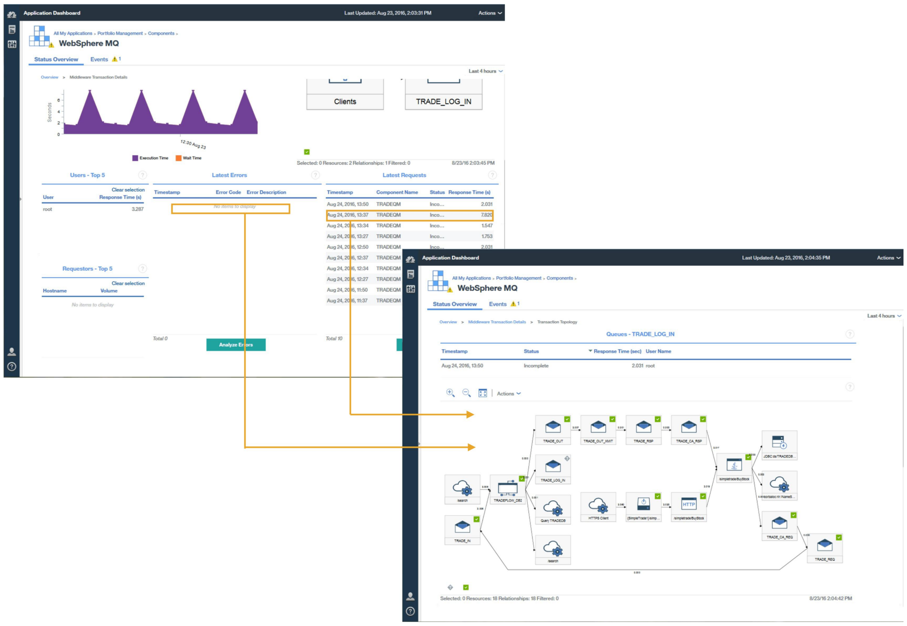
Analyzing errors and instances

From the middleware Transactions Details page, you can drill down further to information that helps you to analyze errors and instances and access the Transaction Instance topology.

To drill down to errors and instances for middleware components and then to the Transaction Instance Topology, on the **Transaction Details** page complete one of the following steps:

- Click **Analyze Errors** to display the **Error Analysis** page, then select an error.
- Click **Analyze Requests** to display the **Instance Analysis** page, then select an instance.

Alternatively, in the **Transaction Details** page, select an error or an instance to go directly to the Transaction Instance Topology.



Application Dashboard

Last Updated: Aug 23, 2016, 2:03:31 PM

Actions

All My Applications > Portfolio Management > Components >

WebSphere MQ

Status Overview | Events ⚠ 1

Last 4 hours

Overview > Middleware Transaction Details

Selected: 0 Resources: 2 Relationships: 1 Filtered: 0 8/23/16 2:03:45 PM

Users - Top 5

Clear selection

User	Response Time (s)
root	3.287

Requestors - Top 5

Clear selection

Hostname	Volume

No items to display

Latest Errors

Timestamp	Error Code	Error Description

No items to display

Total 0

Analyze Errors

Latest Requests

Timestamp	Component Name	Status	Response Time (s)
Aug 24, 2016, 13:50	TRADEQM	Inco...	2.031
Aug 24, 2016, 13:37	TRADEQM	Inco...	7.820
Aug 24, 2016, 13:34	TRADEQM	Inco...	1.547
Aug 24, 2016, 13:27	TRADEQM	Inco...	1.753
Aug 24, 2016, 12:50	TRADEQM	Inco...	2.031
Aug 24, 2016, 12:37	TRADEQM		
Aug 24, 2016, 12:34	TRADEQM		
Aug 24, 2016, 12:27	TRADEQM		
Aug 24, 2016, 11:50	TRADEQM		
Aug 24, 2016, 11:37	TRADEQM		

Total 10

Application Dashboard

Last Updated: Aug 23, 2016, 2:04:35 PM

Actions

All My Applications > Portfolio Management > Components >

WebSphere MQ

Status Overview | Events ⚠ 1

Last 4 hours

Overview > Middleware Transaction Details > Transaction Topology

Queues - TRADE_LOG_IN

Timestamp	Status	Response Time (sec)	User Name
Aug 24, 2016, 13:50	Incomplete	2.031	root

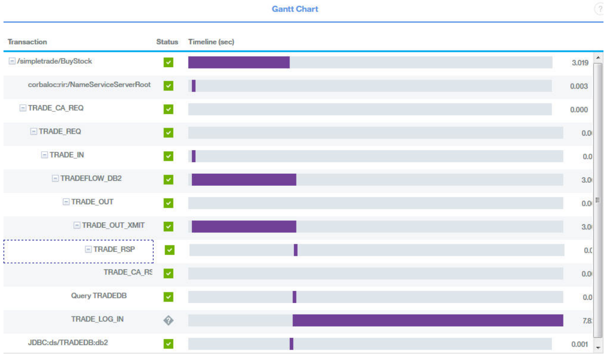
Selected: 0 Resources: 18 Relationships: 18 Filtered: 0 8/23/16 2:04:42 PM

Transaction instance topologies display the following middleware nodes:

- Message queue managers
- IBM Integration Bus brokers
- DataPower Appliances

Select a node to display its properties that explain why a node has a particular status.

A Transaction Gantt Chart is displayed for the selected queue or broker. The Gantt chart helps you isolate the most significant contributors to the overall response time of the transaction.

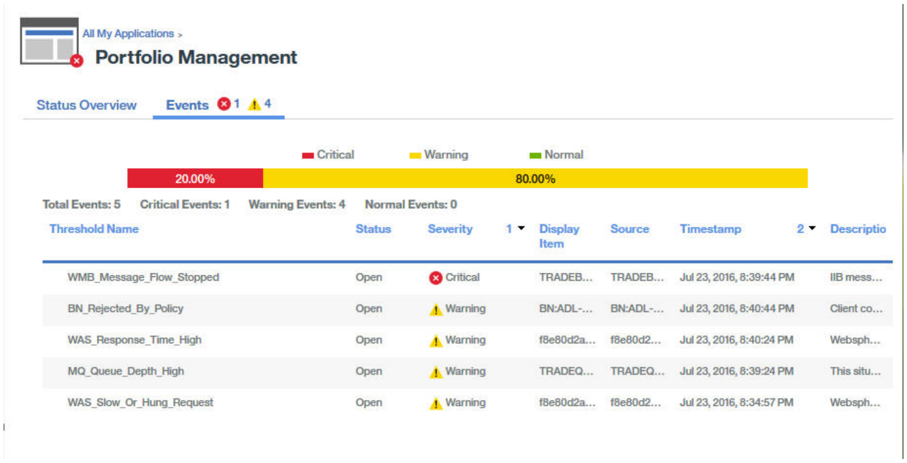


Gantt Chart

Transaction	Status	Timeline (sec)
/simpletrade/BuyStock	✓	3.019
corbaloc:rir:/NameServiceServerRoot	✓	0.003
TRADE_CA_REQ	✓	0.000
TRADE_REQ	✓	0.0
TRADE_IN	✓	0.0
TRADEFLOW_DB2	✓	3.0
TRADE_OUT	✓	0.0
TRADE_OUT_XMIT	✓	3.0
TRADE_RSP	✓	0.0
TRADE_CA_RS	✓	0.0
Query TRADEDB	✓	0.0
TRADE_LOG_IN	◆	7.8
JDBC:ds/TRADEDB:db2	✓	0.001

Events

Events are generated for the IBM integration stack by the default Transaction Tracking thresholds in addition to the other agents.



All My Applications >

Portfolio Management

Status Overview | Events ✖ 1 ⚠ 4

🟥 Critical 🟨 Warning 🟩 Normal

20.00% | 80.00%

Total Events: 5 Critical Events: 1 Warning Events: 4 Normal Events: 0

Threshold Name	Status	Severity 1 ▾	Display Item	Source	Timestamp 2 ▾	Descriptio
WMB_Message_Flow_Stopped	Open	✖ Critical	TRADEB...	TRADEB...	Jul 23, 2016, 8:39:44 PM	IIB mess...
BN_Rejected_By_Policy	Open	⚠ Warning	BN:ADL-...	BN:ADL-...	Jul 23, 2016, 8:40:44 PM	Client co...
WAS_Response_Time_High	Open	⚠ Warning	f8e80d2a...	f8e80d2...	Jul 23, 2016, 8:40:24 PM	Websph...
MQ_Queue_Depth_High	Open	⚠ Warning	TRADEQ...	TRADEQ...	Jul 23, 2016, 8:39:24 PM	This situ...
WAS_Slow_Or_Hung_Request	Open	⚠ Warning	f8e80d2a...	f8e80d2...	Jul 23, 2016, 8:34:57 PM	Websph...

For more information about Transaction Tracking default events, see "Event thresholds for Transaction Monitoring " on page 993.

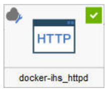
You can add thresholds to create more events, for example for transaction rates that are slow or fall below a certain threshold.

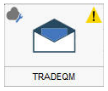
For more information about adding events, see "Creating thresholds to generate events for transaction monitoring" on page 996.

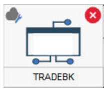
Diagnosing problems in your environment

If transaction instances for one of the components in your environment are slow or failing, the affected component is assigned an appropriate status.

A node might have one of the following statuses:

- 

Good, the node has a tick surrounded by a green square in the upper right corner
- 

Warning, the node has an exclamation point surrounded by a yellow triangle in the upper right corner
- 

Critical, the node has a red background and a cross that is encircled with red in the upper right corner

To identify the cause of the problems for these components with a warning or critical status, right-click the node and drill down to see more information about what might be causing the failures.

Downloading your agents and data collectors

You can access your Cloud APM subscription from the IBM Marketplace website. Sign in to your account and download the installation archive files. The installation archive files include installation and configuration files for the agents and data collectors.

You can learn more about downloading agents and data collectors by completing the steps in the following tutorials:

- "Tutorial: Downloading and installing an agent" on page 106
- "Tutorial: Downloading and configuring a data collector" on page 110

You can sign up for an active trial or purchased subscription for one of the IBM Cloud Application Performance Management offerings from IBM Marketplace.

IBM Marketplace

Sign up for a trial on IBM Marketplace > Cloud APM > Free Trial. Purchase a subscription on IBM Marketplace > Cloud APM > Purchase . Sign in to the Products and services page to download your agents and data collectors.

The **Products and services** page is available to active subscribers. If you have any issues, go to Marketplace support.

Test connectivity

For information about checking connectivity to the Cloud APM server, which is used to download packages, see Network connectivity.

Tutorial: Downloading and installing an agent

Use this tutorial to gain hands-on experience with downloading and installing a Cloud APM Windows OS agent from IBM Marketplace. You can then start the Cloud APM console and check the health of your monitored resource by viewing key performance indicators (KPIs) in the dashboards.

About this task

This tutorial involves downloading the Windows installation package from the **Products and services** page on IBM Marketplace, extracting the installation files, and installing the Windows OS agent. You return to **Products and services** to launch the Cloud APM console and open the Application Performance Dashboard to check the health of your Windows system.

Procedure

1. If you are not signed in to IBM Marketplace, sign in with your IBMid and password and go to **Products and services**.

   The **Products and services** page is available to active subscribers. If you have any issues, go to the Cloud Application Performance Management Forum or to Marketplace support.
2. Download the Windows installation archive file:
   a) In the Cloud APM subscription box, click **Manage** > **Downloads**.

   b) Select the Windows operating system.

      Select the IBM Cloud Application Performance Management Agents 64-bit package. If you are using Windows 32-bit version, select the 32-bit agent package.

   c) Click **Download** and save the agent installation archive file to your system.
      Example:

      ```
      C:\Users\MY_ADMIN\Downloads\IAPM_Agent_Install.zip
      ```
3. On your local Windows system, navigate to the directory where you saved the downloaded archive file and extract it.
   For example, in Windows Explorer, open the **Downloads** directory, right-click `IAPM_Agent_Install.zip`, and select **Extract All**.
4. Open a command prompt as an administrator:
   a) From the Windows **Start** menu, type `command` in the search box.

   b) Right-click **Command Prompt** from the list that displays and select **Run as administrator**.
5. Change to the directory where you extracted the installation files.
   Example:

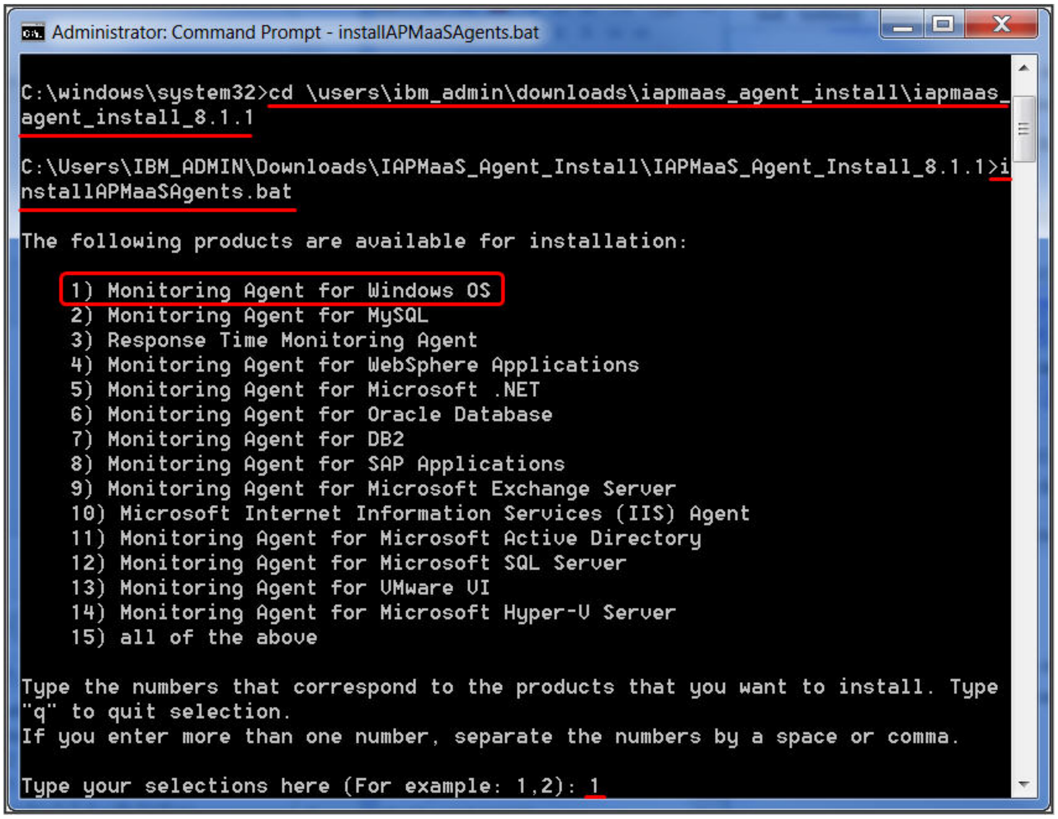
   ```
   cd C:\Users\MY_ADMIN\Downloads\IAPM_Agent_Install\IAPM_Agent_Install_8.1.4
   ```

6. Run the installation script to install the Windows OS agent:
   a) Enter the following command:

      ```
      installAPMAgents.bat
      ```

   b) From the list of available agents, enter the number that corresponds to the Windows OS agent.
   c) Answer the prompts to confirm that you want to install the Windows OS agent and to accept the license agreement.

      A prerequisite scan of your environment begins and takes a few moments to complete. For any missing requirements, you receive a message that directs you to a log file with the reason for the failure. A prerequisite such as insufficient disk space stops the installation. You must address the failure and start the installation script again. If you have any issues, go to the Cloud Application Performance Management Forum or send an email to `info@ibmserviceengage.com`.



```
C:\windows\system32>cd \users\ibm_admin\downloads\iapmaas_agent_install\iapmaas_
agent_install_8.1.1

C:\Users\IBM_ADMIN\Downloads\IAPMaaS_Agent_Install\IAPMaaS_Agent_Install_8.1.1>i
nstallAPMaaSAgents.bat

The following products are available for installation:

    1) Monitoring Agent for Windows OS
    2) Monitoring Agent for MySQL
    3) Response Time Monitoring Agent
    4) Monitoring Agent for WebSphere Applications
    5) Monitoring Agent for Microsoft .NET
    6) Monitoring Agent for Oracle Database
    7) Monitoring Agent for DB2
    8) Monitoring Agent for SAP Applications
    9) Monitoring Agent for Microsoft Exchange Server
    10) Microsoft Internet Information Services (IIS) Agent
    11) Monitoring Agent for Microsoft Active Directory
    12) Monitoring Agent for Microsoft SQL Server
    13) Monitoring Agent for VMware VI
    14) Monitoring Agent for Microsoft Hyper-V Server
    15) all of the above

Type the numbers that correspond to the products that you want to install. Type
"q" to quit selection.
If you enter more than one number, separate the numbers by a space or comma.

Type your selections here (For example: 1,2): 1
```

   To view the Windows operating system requirement report, see System requirements (APM Developer Center).

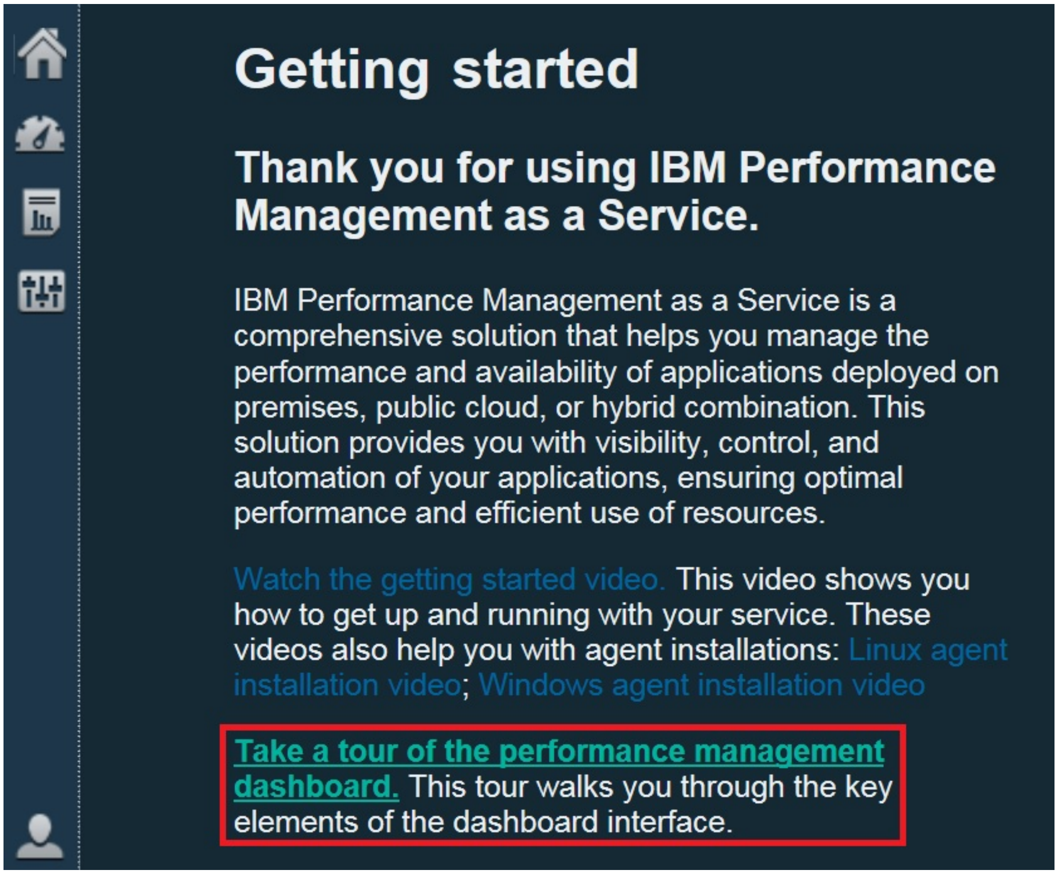
   After successful installation, the Windows OS agent is started automatically and you can start the Cloud APM console to begin monitoring your Windows system.

   Note: If your environment includes a firewall that does not allow transparent outbound HTTPS connections to an external host, you must set up a forward proxy for communications between the agent and Cloud APM server. By setting up a forwarding proxy, you can forward all traffic to a specific point on the network and then allow only a single connection through the firewall. For more information, see "Configuring agents to communicate through a forward proxy" on page 159.

7. Return to **Products and services** on IBM Marketplace and click **Launch** from the Cloud APM subscription box.

   The Cloud APM console opens to the **Getting Started** page where you can learn about the features, watch videos for different user scenarios, and open the associated console pages.

8. On the **Getting Started** page, click "Take a tour of the performance management dashboard" for a quick tour of the navigation elements.



Getting started

Thank you for using IBM Performance Management as a Service.

IBM Performance Management as a Service is a comprehensive solution that helps you manage the performance and availability of applications deployed on premises, public cloud, or hybrid combination. This solution provides you with visibility, control, and automation of your applications, ensuring optimal performance and efficient use of resources.

Watch the getting started video. This video shows you how to get up and running with your service. These videos also help you with agent installations: Linux agent installation video; Windows agent installation video

Take a tour of the performance management dashboard. This tour walks you through the key elements of the dashboard interface.

9. Open the Windows OS summary dashboard:

   a) From the navigation bar, click **Performance** > **Application Performance Dashboard**.

   - The **All My Applications** dashboard is displayed with a summary status box for each defined application in your environment. Initially, only the **My Components** predefined application is displayed.
   - If you see an **Add Application** window instead of **My Components**, create an application to see your monitored resource:

     1) Enter a name for the application, such as "My Apps".

     2) Click +.

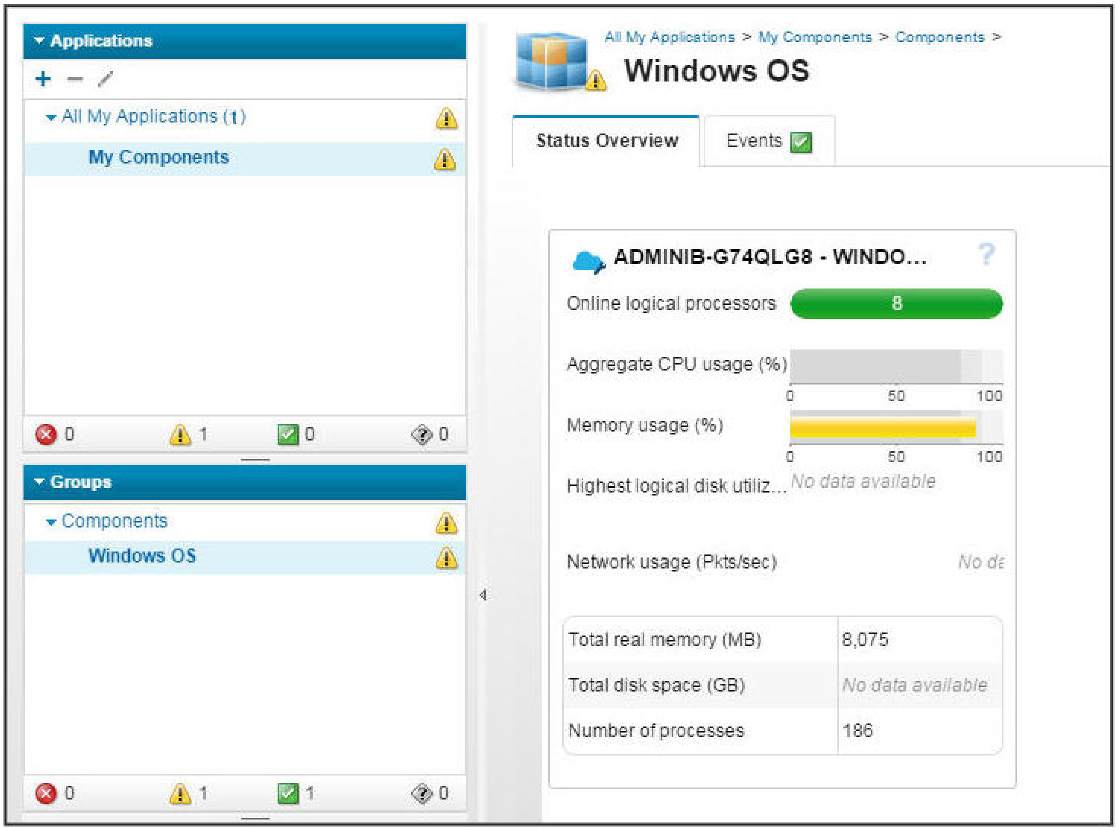
     3) Scroll to the end of the **Select Component** list and click **Windows OS**.

     4) In the Component Editor, click `Primary`:*`Host_Name`*:NT, click **Add**, and click **Back** to add your agent to the application.

     5) Click **Save** to close the window and view a summary status box for your new application in the dashboard.

   b) In the summary box, click **Components**.

   - The summary dashboard for your Windows OS managed system is displayed. From here, you can click anywhere in the status summary group widget to drill down to detailed dashboards with KPIs reported from your Windows OS agent.
   - It can take a few minutes for a newly started agent to communicate with the monitoring infrastructure and send KPIs to the console.



Total real memory (MB)	8,075
Total disk space (GB)	No data available
Number of processes	186

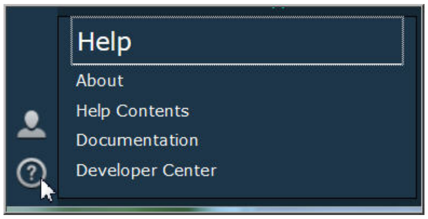
If the agent is not communicating with the Cloud APM server or is not started, the summary dashboard shows no KPIs and the status shows as ◇ unknown. You can use the **os-agent** command to check the status and start the agent if necessary. Open a command prompt as an administrator and enter the **os-agent status** command from the C:\\IBM\APM\bin folder. If the agent is not started, enter the **os-agent start** command.

Results

You installed a Cloud APM agent and observed monitoring data that is sent to the Application Performance Dashboard.

What to do next

- Explore the console: While you are using the Cloud APM console, explore the features. You can learn about the current dashboard by clicking ? in the window banner. You can open the help system or the Cloud APM topic collection on IBM Knowledge Center from the ? **Help** menu in the navigation bar.



- Install other agents: You have all the installation files that you need to install other types of Windows agents for monitoring your environment. You can also install agents on other systems in your environment. If you have AIX or Linux systems, download the associated installation archive file. Some agent types have prerequisites to be completed before you install them, and most agent types require some configuration after you install them. For more information, see “Preinstallation on AIX systems” on page 123, “Preinstallation on Linux systems” on page 129, “Preinstallation on Windows systems” on page 138, and Chapter 7, “Configuring your environment,” on page 159.

Tutorial: Downloading and configuring a data collector

Use this tutorial to gain hands-on experience with downloading and configuring a Cloud APM Bluemix Ruby data collector from IBM Marketplace. You can then start the Cloud APM console and check the health of your monitored resource by viewing key performance indicators (KPIs) in the dashboards.

About this task

This tutorial involves downloading the data collector package for Bluemix applications from the **Products and services** page on IBM Marketplace, extracting the configuration files, and configuring the Bluemix Ruby data collector on a Linux system. You return to **Products and services** to launch the Cloud APM console and open the Application Performance Dashboard to check the health of your Bluemix Ruby application.

Procedure

1. If you are not signed in to IBM Marketplace, sign in with your IBMid and password and go to **Products and services**.

   The **Products and services** page is available to active subscribers. If you have any issues, go to the Cloud Application Performance Management Forum or to Marketplace support.
2. Download the Data Collectors for Bluemix Applications package `IBM_Bluemix_Data_Collectors_Install.tgz`.
3. On your local system, navigate to the directory where you saved the downloaded archive file and extract it by running the following command:

   ```
   tar -zxvf IBM_Bluemix_Data_Collectors_Install.tgz
   ```

   You get four compressed files, each representing a data collector for a type of Bluemix application. The data collector package for Bluemix Ruby applications is `ruby_datacollector.tgz`.
4. Extract the files in `ruby_datacollector.tgz` by running the following command, for example:

   ```
   tar -zxvf ruby_datacollector.tgz
   ```

   You get an `ibm_ruby_dc` folder.
5. Copy the entire `etc` folder in `ibm_ruby_dc` to the root folder of your Ruby application by running the following command, for example:

   ```
   cp –r directory to the etc folder  home directory of your Ruby application
   ```

   If you extract the data collector to the `/opt/ibm/ccm/ibm_ruby_dc/etc` directory and the home directory of your Ruby application is `/root/ruby_app/`, the command is as follows:

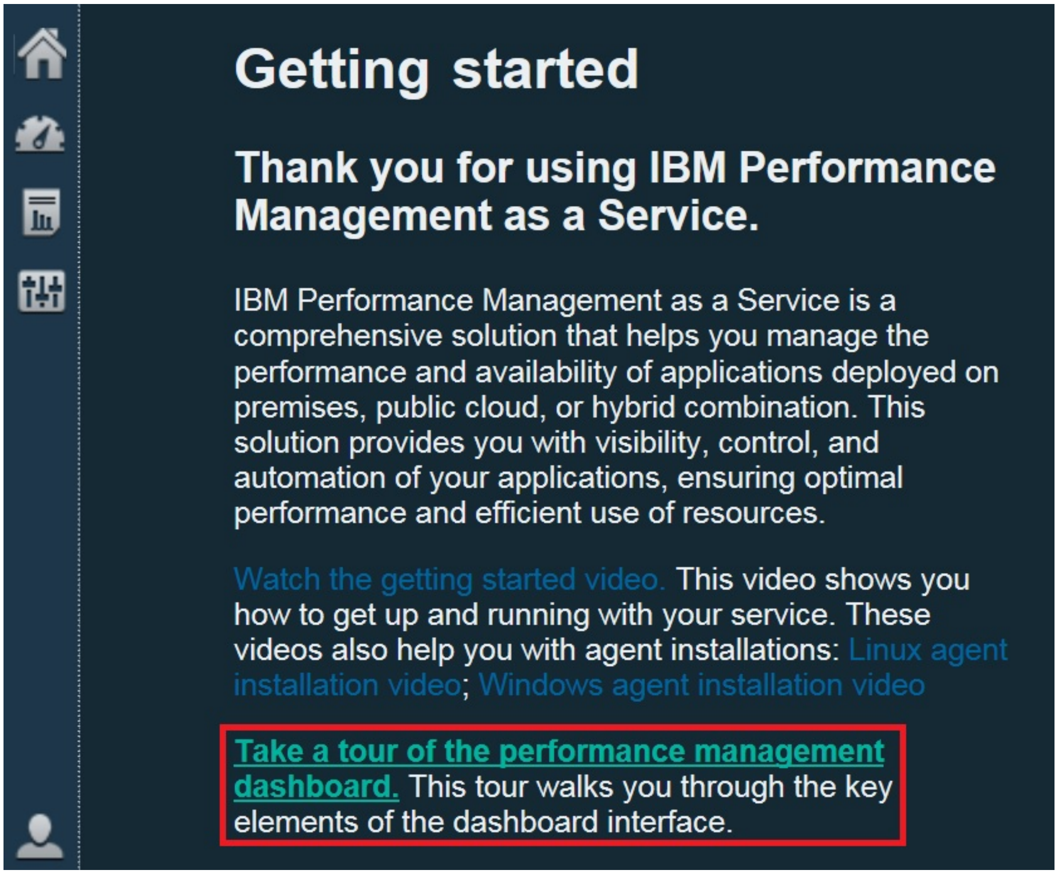
   ```
   cp –r /opt/ibm/ccm/ibm_ruby_dc/etc /root/ruby_app/
   ```
6. Add the following section to the `Gemfile` in the home folder of your Bluemix Ruby application:

   ```
   gem 'logger',  '>= 1.2.8'
   source 'https://managemserver.ng.bluemix.net' do
     gem 'ibm_resource_monitor'
     gem 'stacktracer'
   end
   ```
7. Run the `bundle lock` command to regenerate the `Gemfile.lock` file.
8. From the home directory of your Ruby application, run the following command:

   ```
   cf push
   ```
9. Return to **Products and services** on IBM Marketplace and click **Launch** from the Cloud APM subscription box.

   The Cloud APM console opens to the **Getting Started** page where you can learn about the features, watch videos for different user scenarios, and open the associated console pages.

10. On the **Getting Started** page, click "Take a tour of the performance management dashboard" for a quick tour of the navigation elements.



Getting started

Thank you for using IBM Performance Management as a Service.

IBM Performance Management as a Service is a comprehensive solution that helps you manage the performance and availability of applications deployed on premises, public cloud, or hybrid combination. This solution provides you with visibility, control, and automation of your applications, ensuring optimal performance and efficient use of resources.

Watch the getting started video. This video shows you how to get up and running with your service. These videos also help you with agent installations: Linux agent installation video; Windows agent installation video

Take a tour of the performance management dashboard. This tour walks you through the key elements of the dashboard interface.

11. Open the Bluemix Ruby applications summary dashboard:

    a) From the navigation bar, click **Performance** > **Application Performance Dashboard**.

    - The **All My Applications** dashboard is displayed with a summary status box for each defined application in your environment. Initially, only the **My Components** predefined application is displayed.
    - If you see an **Add Application** window instead of **My Components**, create an application to see your monitored resource:

        1) Enter a name for the application, such as "My Apps".

        2) Click +.

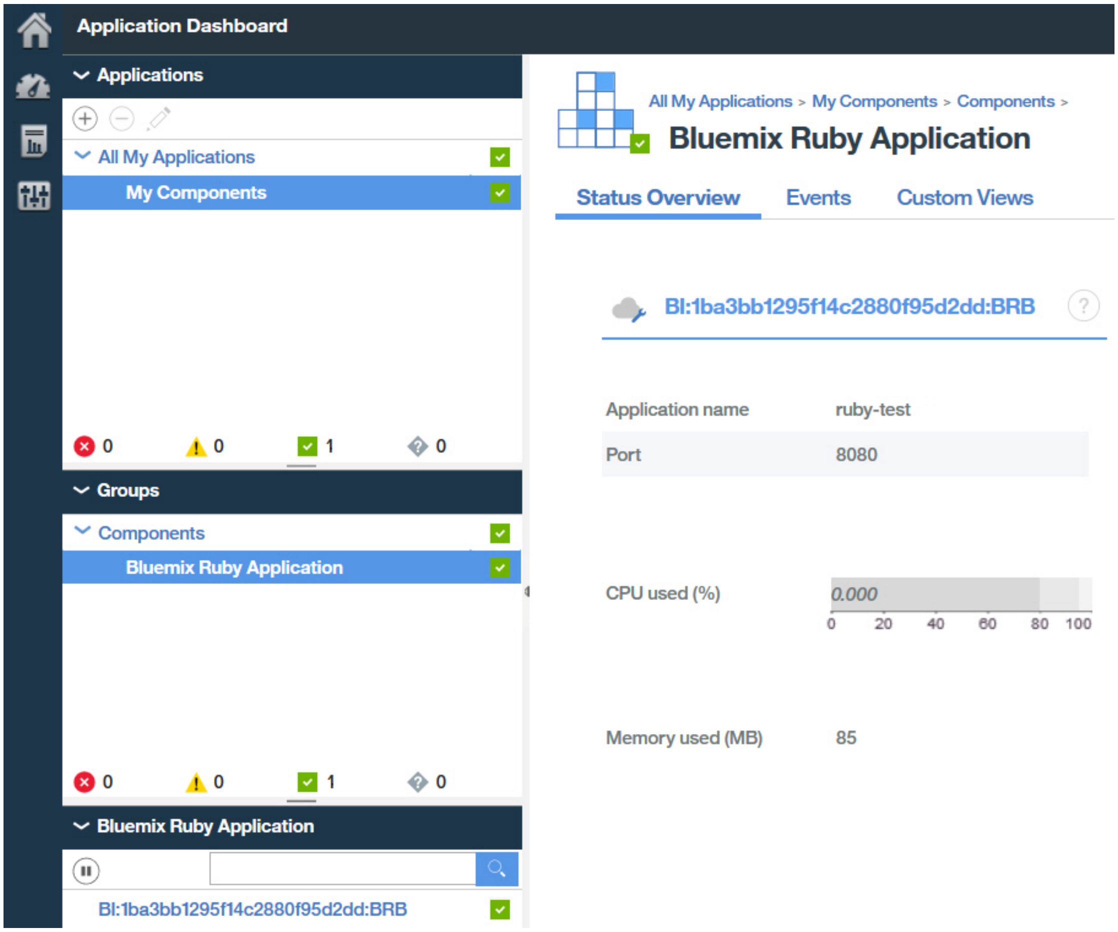
        3) Click **Bluemix Ruby Application**.

        4) In the Component Editor, select an instance, click **Add**, and click **Back** to add your data collector to the application.

        5) Click **Save** to close the window and view a summary status box for your new application in the dashboard.

    b) In the summary box, click **Components**.

    - The summary dashboard for your Bluemix Ruby application is displayed. From here, you can click anywhere in the status summary group widget to drill down to detailed dashboards with KPIs reported from your Bluemix Ruby data collector.
    - It can take a few minutes for a newly started data collector to communicate with the monitoring infrastructure and send KPIs to the console.



Application name ruby-test

Port 8080

CPU used (%) 0.000

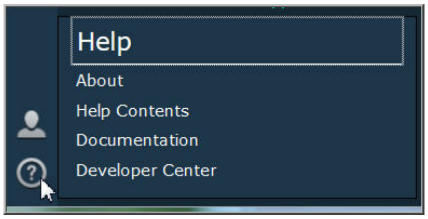
Memory used (MB) 85

Results

You installed a Cloud APM data collector and observed monitoring data that is sent to the Application Performance Dashboard.

What to do next

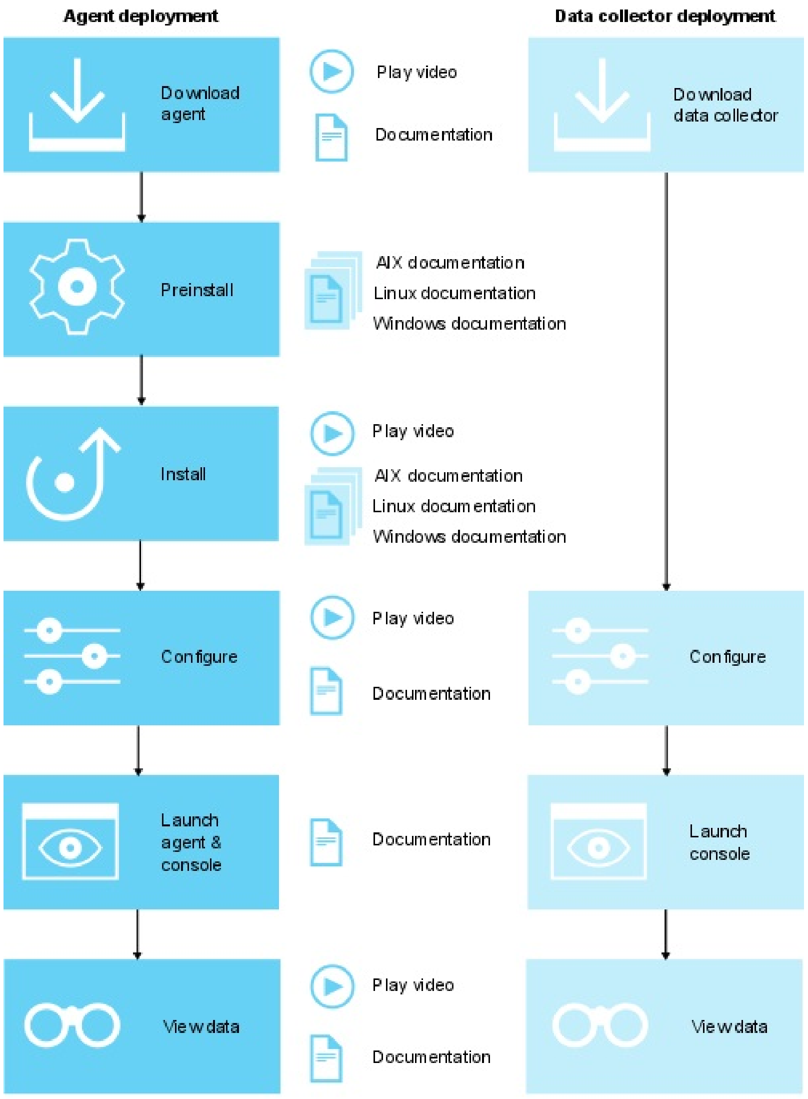
- Explore the console: While you are using the Cloud APM console, explore the features. You can learn about the current dashboard by clicking (?) in the window banner. You can open the help system or the Cloud APM topic collection on IBM Knowledge Center from the (?) **Help** menu in the navigation bar.



- Install other data collectors: You have all the installation files that you need to install other types of data collectors for monitoring your environment. You can also install data collectors on other systems in your environment. For more information, see Chapter 7, “Configuring your environment,” on page 159.

Chapter 5. Agent and data collector deployment

Agents vary in the tasks that are required between installation and viewing the data that they collect. Some tasks are automatic and other tasks are manual. After you download your data collectors, you must manually configure each data collector.



1. Play download video
2. Download documentation
3. AIX preinstallation documentation
4. Linux preinstallation documentation
5. Windows preinstallation documentation
6. Play installation video
7. AIX installation documentation
8. Linux installation documentation
9. Windows installation documentation

10. Play configuration videos
11. Configuration documentation
12. Launch documentation
13. Play view data video
14. Features documentation

1. Play download video
2. Download documentation
3. AIX preinstallation documentation
4. Linux preinstallation documentation
5. Windows preinstallation documentation
6. Play installation video
7. AIX installation documentation
8. Linux installation documentation
9. Windows installation documentation
10. Play configuration videos
11. Configuration documentation
12. Launch documentation
13. Play view data video
14. Features documentation

After installation, some agents are configured and started automatically

For any agent that is started, the agent is configured with the default settings. To determine which agents are configured and started manually, see Table 7 on page 114.

After installation, some agents require manual configuration but start automatically

For information about how to configure your agents, see Chapter 7, "Configuring your environment," on page 159 To determine which agents are configured manually and started automatically, see Table 7 on page 114.

After installation, some agents must be configured and started manually

For any agent that is not started automatically, you must configure the agent before it can be started. To determine which agents are configured and started manually, see Table 7 on page 114.

Multiple instance agents require creating a first instance and starting manually

You must create the first instance and start the agent manually. A multiple instance agent means that a single installation of the agent instantiates a unique monitoring instance for each unique application instance. These instances are visualized from the Cloud APM console as a result. To determine which agents are multiple instance agents, see Table 7 on page 114.

OS agents and log file monitoring

The Linux OS agent, UNIX OS agent, and Windows OS agent are configured and started automatically. However, you can configure log file monitoring for the OS agents, so that you can monitor application log files. For more information, see "Configuring OS agent log file monitoring" on page 626.

Agent and data collector configuration, startup, and instance characteristics

Table 7. Postinstallation checklist

Agent or data collector	Configured and started automatically	Configured manually and started automatically	Configured and started manually	Multiple instance (started manually)
Amazon EC2 agent	—	—	—	✓

Table 7. Postinstallation checklist (continued)

Agent or data collector	Configured and started automatically	Configured manually and started automatically	Configured and started manually	Multiple instance (started manually)
Amazon ELB agent	—	—	—	✓
Azure Compute agent	—	—	—	✓
Cassandra agent	—	—	✓	✓
Cisco UCS agent	—	—	✓	✓
Citrix VDI agent	—	—	—	✓
DataPower agent	—	—	—	✓
DataStage agent	—	—	✓	✓
Db2 agent	—	—	—	✓
Hadoop agent	—	—	✓	—
HMC Base agent	—	—	—	✓
HTTP Server agent	✓ You must review the configuration file that the agent creates for the HTTP Server. Then, you must add the data collector configuration manually to the server configuration file.	—	—	—
IBM Cloud agent	—	—	—	✓
IBM Integration Bus agent	—	—	—	✓
J2SE data collector	—	✓	—	—
JBoss agent	—	—	—	✓
Liberty data collector	—	✓	—	—
Linux KVM agent	—	—	✓	✓
Linux OS agent	✓	—	—	—

Table 7. Postinstallation checklist (continued)

Agent or data collector	Configured and started automatically	Configured manually and started automatically	Configured and started manually	Multiple instance (started manually)
Microsoft Active Directory agent	–	✓ This agent is started automatically. However, you must configure the agent to view data for some attributes.	–	–
Microsoft Cluster Server agent	–	–	✓	–
Microsoft Exchange Server agent	–	✓ This agent is started automatically. However, you must configure the agent to view data for all attributes.	–	–
Microsoft Hyper-V Server agent	✓	–	–	–
Microsoft IIS agent	✓	–	–	–
Skype for Business Server agent (formerly known as Microsoft Lync Server agent)	✓	✓ This agent is started automatically. However, you must configure the agent to view data for some attributes.	–	–
Microsoft Office 365 agent	–	–	✓	–
Microsoft .NET agent	–	✓ The data collector must be configured before data is reported.	–	–
Microsoft SharePoint Server agent	✓	–	–	–

Table 7. Postinstallation checklist (continued)

Agent or data collector	Configured and started automatically	Configured manually and started automatically	Configured and started manually	Multiple instance (started manually)
Microsoft SQL Server agent	—	—	—	✓ Each agent instance must be configured and started manually.
MQ Appliance agent	—	—	—	✓
MongoDB agent	—	—	—	✓
MySQL agent	—	—	—	✓
NetApp Storage agent	—	—	✓	✓
Node.js agent	—	✓ The agent must be configured before data is reported. You must add a monitoring plug-in to your Node.js application.	—	—
Node.js data collector	—	✓	—	—
OpenStack agent	—	—	✓	✓
Oracle Database agent	—	—	—	✓
PHP agent	—	—	—	✓
PostgreSQL agent	—	—	—	✓
Python data collector	—	✓	—	—
RabbitMQ agent	—	—	✓	✓
Response Time Monitoring Agent	✓	—	—	—

Table 7. Postinstallation checklist (continued)

Agent or data collector	Configured and started automatically	Configured manually and started automatically	Configured and started manually	Multiple instance (started manually)
Ruby agent	—	—	—	✓ For deep-dive diagnostics, the agent must be configured before data is reported. To enable the diagnostics dashboards, you must install and configure the diagnostics data collector.
Ruby data collector	—	✓	—	—
SAP agent	—	—	✓	✓
SAP HANA Database agent	—	—	✓	✓
SAP NetWeaver Java Stack agent	—	—	✓	✓
Siebel agent	—	—	—	✓
Sterling Connect Direct agent	—	—	—	✓
Sterling File Gateway agent	—	—	✓	✓
Sybase agent	—	—	✓	✓
Synthetic Playback agent	✓ The agent is configured and started automatically for public, external-facing applications, but transactions must be created in the Synthetic Script Manager before data is reported.	✓ The agent is started automatically but the agent must be configured for private, internal-facing applications. Transactions must be created in the Synthetic Script Manager before data is reported.	—	✓
Tomcat agent	—	—	—	✓

Table 7. Postinstallation checklist (continued)

Agent or data collector	Configured and started automatically	Configured manually and started automatically	Configured and started manually	Multiple instance (started manually)
UNIX OS agent	✓	—	—	—
VMware VI agent	—	—	✓	✓
WebLogic agent	—	—	—	✓
WebSphere Applications agent	—	✓ The agent is started automatically but the data collector must be configured before data is reported.	—	—
WebSphere Infrastructure Manager agent	—	—	—	✓
WebSphere MQ agent	—	—	—	✓
Windows OS agent	✓	—	—	—

Chapter 6. Installing your agents

The IBM Cloud Application Performance Management infrastructure is installed and managed by IBM. To monitor your applications, you select and install the monitoring agents for the applications you want to monitor. You can install the agents on Linux, AIX, or Windows operating systems. The stand-alone data collectors do not require installation.

If you choose stand-alone data collectors to monitor your applications, you can skip the installation procedure. Continue to Chapter 7, "Configuring your environment," on page 159 for instructions about how to deploy data collectors for monitoring your applications.

Remote monitoring

Some agents can be installed remotely from the resource that they are monitoring. The following agents support remote monitoring:

- Monitoring Agent for Amazon EC2
- Monitoring Agent for AWS Elastic Load Balancer
- Monitoring Agent for Azure Compute
- Monitoring Agent for Cassandra
- Monitoring Agent for Cisco UCS
- Monitoring Agent for Citrix Virtual Desktop Infrastructure
- Monitoring Agent for DataPower - This agent can be installed only on a remote machine.
- Monitoring Agent for Db2
- Monitoring Agent for Hadoop
- Monitoring Agent for HMC Base
- Monitoring Agent for IBM Cloud
- Monitoring Agent for InfoSphere DataStage
- Monitoring Agent for JBoss - If you want to use this agent for resource monitoring, install it remotely or locally. If you want to use the agent for transaction tracking and deep dive diagnostics, install it locally.
- Monitoring Agent for Linux KVM
- Monitoring Agent for MariaDB
- Monitoring Agent for Microsoft Cluster Server
- Monitoring Agent for Microsoft Exchange Server
- Monitoring Agent for Microsoft Office 365
- Monitoring Agent for Microsoft SharePoint Server
- Monitoring Agent for MongoDB
- Monitoring Agent for MySQL
- Monitoring Agent for NetApp Storage
- Monitoring Agent for OpenStack
- Monitoring Agent for Oracle Database
- Monitoring Agent for PostgreSQL
- Monitoring Agent for RabbitMQ
- Monitoring Agent for SAP Applications
- Monitoring Agent for SAP HANA Database

- Monitoring Agent for SAP NetWeaver Java Stack - If you want to use this agent for resource monitoring, install it remotely or locally. If you want to use the agent for transaction tracking and deep dive diagnostics, install it locally.
- Monitoring Agent for Sterling Connect Direct
- Monitoring Agent for Sterling File Gateway
- Monitoring Agent for VMware VI
- Monitoring Agent for WebLogic - If you want to use this agent for resource monitoring, install it remotely or locally. If you want to use the agent for transaction tracking and deep dive diagnostics, install it locally.

Installing agents on UNIX systems

Install monitoring agents on your AIX or Solaris systems for the resources that you want to manage.

Agent list that you can install on AIX

- Monitoring Agent for DataPower
- Monitoring Agent for Cassandra
- Monitoring Agent for Db2
- Monitoring Agent for Hadoop
- Monitoring Agent for HMC Base
- Monitoring Agent for HTTP Server
- Monitoring Agent for IBM Integration Bus
- Monitoring Agent for MQ Appliance
- Monitoring Agent for Oracle Database
- Monitoring Agent for SAP Applications
- Monitoring Agent for SAP HANA Database
- Monitoring Agent for SAP NetWeaver Java Stack
- Monitoring Agent for Siebel
- Monitoring Agent for Sybase Server
- Monitoring Agent for UNIX OS
- Monitoring Agent for WebLogic
- Monitoring Agent for WebSphere Applications
- Monitoring Agent for WebSphere MQ
- Response Time Monitoring Agent

Agent list that you can install on Solaris Sparc

- Monitoring Agent for Db2
- Monitoring Agent for HTTP Server
- Monitoring Agent for JBoss
- Monitoring Agent for MySQL
- Monitoring Agent for Oracle Database
- Monitoring Agent for SAP Applications
- Monitoring Agent for Sybase Server
- Monitoring Agent for UNIX OS
- Monitoring Agent for WebSphere Applications
- Monitoring Agent for WebLogic

Agent list that you can install on Solaris X86

- Monitoring Agent for Db2
- Monitoring Agent for IBM Integration Bus
- Monitoring Agent for SAP Applications
- Monitoring Agent for Sybase Server
- Monitoring Agent for UNIX OS
- Monitoring Agent for WebSphere Applications
- Monitoring Agent for WebSphere MQ

Preinstallation on AIX systems

You must complete the required preinstallation tasks before you install agents on AIX systems. Some preinstallation tasks are agent-specific and other tasks apply to multiple agents.

Note: These requirements are in addition to the requirements identified in the Software Product Compatibility Reports for your agent: For the current version requirements and dependencies for your agent, see the IBM Cloud Application Performance Management Software Product Compatibility Report for agents on AIX.

All agents

The following preinstallation tasks are applicable to all agents:

Test connectivity
Before you install agents, ensure that your system can communicate with the Cloud APM server. For information about checking connectivity to the Cloud APM server, see Network connectivity.

Non-root user installation
You must have read, write, and execute permissions for the installation directory. Otherwise, the installation is canceled. For more information about non-root user installation, see "Installing agents as a non-root user" on page 142.

70-character limitation for installation path
The installation directory and the path to it must be no more than 70 characters.

100-character limitation for `.tar` file names
The default **`tar`** command on AIX systems cannot handle file names that are longer than 100 characters. To avoid installation issues, complete the following steps:

1. Download and install the GNU version of the **`tar`** command from the AIX Toolbox for Linux Applications website.
2. Make the GNU version your default **`tar`** command. Complete one of the following steps:
   - Add `/opt/freeware/bin` to the beginning of the current *PATH* environment variable. For example:

     ```
     export PATH=/opt/freeware/bin:$PATH
     ```

     where /opt/freeware/bin is the directory of GUN bin.
   - Replace `/bin/tar` with a symbolic link to `/opt/freeware/bin/tar` as below:

     ```
     ln -s /opt/freeware/bin/tar /bin/tar
     ```

Alternatively, upgrade to the latest version of AIX to receive the code fix for handling file names longer than 100 characters. For details, see the TAR command Technote for AIX V6.1 or the TAR command Technote for AIX V7.1.

Setting the *CANDLEHOME* environment variable
If you used the ITM Agent Converter to install and configure an agent on the same managed system before, the *CANDLEHOME* environment variable changed to that directory where you installed the agent with the Agent Converter. Before you install and configure a native Cloud APM agent, you must

set the *CANDLEHOME* environment variable to a different directory, otherwise, the native Cloud APM agent cannot start.

Specific agents

The following preinstallation tasks are applicable to the specified agents:

DataPower agent

Before the agent is installed, the prerequisite checker checks that *ulimit* is set to **`unlimited`** on AIX. You must run the **`ulimit -d unlimited`** command to ensure that the *max data segment size* system environment variable is set to **`unlimited`**. This agent cannot be installed on the same machine as the DataPower appliance that you want to monitor.

HMC Base agent

If you plan to install the agent as a root user, you must ensure that system TL07 is installed. If you plan to install the agent as a non-root user, you must ensure that system TL08 is installed for AIX version 6 only.

HTTP Server agent

Install and run this agent as a root user. Use the same user ID to install and run the agent. If you install and run the agent as a non-root user, the non-root user must have the same user ID as the user who started the IBM HTTP Server. Otherwise, the agent has problems with discovering the IBM HTTP Server.

The installation fails on AIX because on the AIX system the default **`.tar`** command truncated a long path. For more information, see the "100-character limitation for `.tar` file names" section in this topic.

AIX only: Install the lynx utility or the curl application.

Oracle Database agent

On Red Hat Enterprise Linux version 5 and version 6 and SUSE Linux Enterprise Server version 11 and version 12 x64, if the Oracle Database agent monitors the Oracle database remotely, you must install the Oracle instant clients first. Install the Oracle instant clients from Oracle Technology Network - Instant Client Downloads.

The instant client v10.x, v11.x, and v12.x are supported by the Oracle Database agent.

Response Time Monitoring Agent

Before you install the Response Time Monitoring agent, review the installation planning section here: “Planning the installation ” on page 679.

SAP HANA Database agent

1. Install SAP HANA database client HDBSQL version 1.00.102.06 or later on the AIX system.
2. Run the following command to add the path of the installation directory to the **`LIBPATH`** environment variable:

   ```
   export LIBPATH=$LIBPATH:install_directory_path
   ```

   Example: `export LIBPATH=$LIBPATH:/usr/sap/hdbclient`, where `/usr/sap/hdbclient` indicates the installation path of the SAP HANA database client.

Important:

If the installation path of the SAP HANA database client is not added to the **`LIBPATH`** environment variable, the prerequisite scanner returns the `FAIL` result.

The environment variable that you added by using the export command persists only for a particular session of the terminal. Therefore, ensure that you run the agent installation script from the same terminal that was used for adding the environment variable.

WebSphere Applications agent

Before the agent is installed, the prerequisite checker checks that *ulimit* is set to **`524000`** on the AIX system. You must run the **`ulimit -d 524000`** command to ensure that the *max data segment size* system environment variable is set to **`524000`**.

Preinstallation on Solaris systems

You must complete the required preinstallation tasks before you install agents on Solaris systems. Some preinstallation tasks are agent-specific and other tasks apply to multiple agents.

Note: These requirements are in addition to the requirements identified in the Software Product Compatibility Reports for your agent: For the current version requirements and dependencies for your agent, see the IBM Cloud Application Performance Management Software Product Compatibility Report for agents on Solaris.

All agents

The following preinstallation tasks are applicable to all agents:

Test connectivity
Before you install agents, ensure that your system can communicate with the Cloud APM server. For information about checking connectivity to the Cloud APM server, see Network connectivity.

Non-root user installation
You must have read, write, and execute permissions for the installation directory. Otherwise, the installation is canceled. For more information about non-root user installation, see "Installing agents as a non-root user" on page 142.

70-character limitation for installation path
The installation directory and the path to it must be no more than 70 characters.

100-character limitation for `.tar` file names
The default **`tar`** command on Solaris systems cannot handle file names that are longer than 100 characters. To avoid @LongLink error issues, complete the following steps:

1. Download and install the GNU version of the **`tar`** command from the http://www.gnu.org website.
2. Make the GNU version your default **`tar`** command. Complete one of the following steps:
   - In the *PATH* environment variable, put the following variable first:

     ```
     export PATH=/opt/freeware/bin:$PATH
     ```

   - Replace `/bin/tar` with symbolic link to `/opt/freeware/bin/tar`

Setting the *CANDLEHOME* environment variable
If you used the ITM Agent Converter to install and configure an agent on the same managed system before, the *CANDLEHOME* environment variable changed to that directory where you installed the agent with the Agent Converter. Before you install and configure a native Cloud APM agent, you must set the *CANDLEHOME* environment variable to a different directory, otherwise, the native Cloud APM agent cannot start.

Specific agents

The following preinstallation tasks are applicable to the specified agents:

HTTP Server agent
Install and run this agent as a root user. Use the same user ID to install and run the agent. If you install and run the agent as a non-root user, the non-root user must have the same user ID as the user who started the IBM HTTP Server. Otherwise, the agent has problems with discovering the IBM HTTP Server.

Installing agents

You can install any combination of monitoring agents on a managed system. For example, if you install the Ruby agent to monitor Ruby On Rails applications, you might want to also install the Response Time Monitoring Agent, the Linux OS agent, or both agents. With the Response Time Monitoring agent, you can gather more response time information for your Ruby applications. With the Linux OS agent, you can monitor other aspects of the system, such as the overall CPU, memory, and disk.

The offering determines which monitoring agents are available for installation. For a list of the agents included in each offering, see "Capabilities" on page 56.

For a list of the agents that run on AIX and Solaris systems, see "Installing agents on UNIX systems" on page 122.

Before you begin

Download the agents. See "Downloading your agents and data collectors" on page 105.

Review the information in "System requirements" on page 85 to make sure that you have met the requirements for the agents you plan to install.

Review the agent preinstallation tasks before you install the agents.

- For AIX systems, see "Preinstallation on AIX systems" on page 123.
- For Solaris systems, see "Preinstallation on Solaris systems" on page 125.

Important: Java Runtime is installed only when the agent requires it and is not always available. Also, ksh is no longer required for agent installation, and SELinux in enforcing mode is supported.

About this task

You can install monitoring agents as a root user or non-root user. If you do not have root privileges and you want to install a monitoring agent, you can install the agent as a non-root user, see "Installing agents as a non-root user" on page 142. Also, you can install the agent as a non-root user if you are a host administrator and you do not want to run the monitoring agent as a root user. Installation flow is the same as for a root user.

Agent coexistence is supported. You can install IBM Cloud Application Performance Management agents on the same computer where IBM Tivoli Monitoring agents are installed. However, both agent types cannot be installed in the same directory. For more information about agent coexistence see "Cloud APM agent and Tivoli Monitoring agent coexistence" on page 930.

Procedure

1. Open a terminal shell session on the AIX system or Solaris system.
2. On your system, navigate to the directory where you downloaded the .tar file.

   The agents must be installed on the system where the application that you want to monitor is installed. If needed, transfer the installation archive file to the system to be monitored. The archive file contains the agents and installation script.

   Remember: Make sure that the directory does not contain an older version of the archive file.
3. Extract the installation files by using the following command:

   ```
   tar -xf ./installation files
   ```

   where `installation files` is the installation file name for your offering.

   The installation script is extracted to a directory named for the archive file and version. For example: *offering*`_Agent_Install_8.1.4.0`. Agent binary and configuration-related files are extracted into subdirectories within that directory.
4. Optional: This step is required ONLY for Solaris 10. You must create a soft link to ksh before you run the installation script on Solaris 10.

   a) Backup the `/bin/sh` command:

   ```
   mv /bin/sh /bin/sh.bkup_origin
   ```

   b) Create a soft link to ksh command:

   ```
   ln -s /bin/ksh /bin/sh
   ```

   c) Confirm that the result points to ksh:

   ```
   ls -l /bin/sh
   ```

5. Run the installation script from the directory that is named for the archive file and version:

   ```
   ./installAPMAgents.sh
   ```

   To install the agents in silent mode, see “Installing agents silently” on page 145.
6. Specify whether to install individual agents, a combination of the agents, or all of the agents.
7. Depending on whether you are installing or upgrading the agents, take one of the following steps:

   - If you are installing the agents, specify a different agent installation home directory or use the applicable default directory:

     – /opt/ibm/apm/agent
   - If you are upgrading the agents, after you are prompted for the agent installation home directory, enter the installation directory of the previous version of the agents.
8. When you are asked if you accept the license agreement, enter 1 to accept the agreement and continue, or enter 2 to decline.

   After you enter 1 (accept), a prerequisite scan of your environment starts and takes a few moments to complete. If any requirements are missing, a message directs you to a log file with the reason for the failure. A prerequisite, such as a missing library or insufficient disk space, stops the installation. You must address the failure, and start the installation script again.
9. If you installed the agents by using a non-root user ID, you must update the system startup scripts (see “Installing agents as a non-root user” on page 142).
10. After installation is complete and the command line is available, you can repeat the steps in this procedure to install more monitoring agents on the managed system.

What to do next

Configure the agent as required. If your monitoring agent requires configuration as described in Chapter 5, “Agent and data collector deployment,” on page 113 or if you want to review the default settings, see Chapter 7, “Configuring your environment,” on page 159.

- If you are using a forward proxy because your firewall does not allow transparent outbound HTTPS connections to external hosts, you must edit the agent environment configuration file. For instructions, see “Configuring agents to communicate through a forward proxy” on page 159.
- If you upgraded an agent from a previous version, identify any reconfiguration or migration tasks that you must complete before logging in to the Cloud APM console. For information about those tasks, see “Upgrading your agents” on page 1111. After an upgrade, you must restart any agent that is not both automatically configured and started by the installer.

To start an agent, run the following command:

```
./name-agent.sh start
```

For information about the monitoring agent commands, including the *name* to use, see “Using agent commands” on page 177. For information about which agents are started automatically and manually, see Chapter 5, “Agent and data collector deployment,” on page 113.

After an upgrade, you must restart any agent that is not both automatically configured and started by the installer.

After you configure and start the agent, view the data that the agent is collecting.

- If you are not logged in, follow the instructions in “Starting the Cloud APM console” on page 957.
- If you want to view managed systems from your IBM Tivoli Monitoring domain in the Application Performance Dashboard, complete the tasks that are described in “Integrating with IBM Tivoli Monitoring V6.3 ” on page 929.

- Restart the apmui service on the Cloud APM server so that agent online help updates are displayed in the Cloud APM console. The apmui service is restarted by using the `apm restart apmui` command.

Installing agents on Linux systems

Install monitoring agents on your Linux systems for the resources that you want to manage.

- Monitoring Agent for Amazon EC2
- Monitoring Agent for AWS Elastic Load Balancer
- Monitoring Agent for Azure Compute
- Monitoring Agent for Cassandra
- Monitoring Agent for Cisco UCS
- Monitoring Agent for Citrix Virtual Desktop Infrastructure
- Monitoring Agent for DataPower
- Monitoring Agent for Db2
- Monitoring Agent for Hadoop
- Monitoring Agent for HTTP Server
- Monitoring Agent for IBM Cloud
- Monitoring Agent for IBM Integration Bus
- Monitoring Agent for Internet Services
- Monitoring Agent for MQ Appliance
- Monitoring Agent for InfoSphere DataStage
- Monitoring Agent for JBoss
- Monitoring Agent for Linux OS
- Monitoring Agent for Linux KVM
- Monitoring Agent for MariaDB
- Monitoring Agent for Microsoft SQL Server
- Monitoring Agent for MongoDB
- Monitoring Agent for MySQL
- Monitoring Agent for NetApp Storage
- Monitoring Agent for Node.js
- Monitoring Agent for OpenStack
- Monitoring Agent for Oracle Database
- Monitoring Agent for PHP
- Monitoring Agent for PostgreSQL
- Monitoring Agent for RabbitMQ
- Monitoring Agent for Ruby
- Monitoring Agent for SAP Applications
- Monitoring Agent for SAP HANA Database
- Monitoring Agent for SAP NetWeaver Java Stack
- Monitoring Agent for Siebel
- Monitoring Agent for Sterling Connect Direct
- Monitoring Agent for Sterling File Gateway
- Monitoring Agent for Sybase Server
- Monitoring Agent for Synthetic Playback

- Monitoring Agent for Tomcat
- Monitoring Agent for VMware VI
- Monitoring Agent for WebLogic
- Monitoring Agent for WebSphere Applications
- Monitoring Agent for WebSphere Infrastructure Manager
- Monitoring Agent for WebSphere MQ
- Response Time Monitoring Agent

The following agents are supported on Linux on Power Little Endian (pLinux LE) systems:

- Monitoring Agent for Db2
- Monitoring Agent for IBM Integration Bus
- Monitoring Agent for Linux OS
- Monitoring Agent for Tomcat - Support available for resource monitoring.
- Monitoring Agent for WebSphere Applications
- Monitoring Agent for WebSphere MQ

The following agents are supported on Linux for System z systems:

- Monitoring Agent for Db2
- Monitoring Agent for HTTP Server - Transaction tracking is not supported.
- Monitoring Agent for IBM Integration Bus
- Monitoring Agent for Linux OS
- Response Time Monitoring Agent
- Monitoring Agent for Tomcat
- Monitoring Agent for WebSphere Applications
- Monitoring Agent for WebSphere MQ

The following agent is supported on Linux for System x systems:

- Monitoring Agent for HTTP Server - Transaction tracking is not supported.

Preinstallation on Linux systems

You must complete the required preinstallation tasks before you install agents on Linux systems. Some preinstallation tasks are agent-specific and other tasks apply to multiple agents.

Note: These requirements are in addition to the requirements identified in the Software Product Compatibility Reports for your agent: For the current version requirements and dependencies for your agent, see the IBM Cloud Application Performance Management Software Product Compatibility Report for agents on Linux.

All agents

The following preinstallation tasks are applicable to all agents:

Test connectivity
Before you install agents, ensure that your system can communicate with the Cloud APM server. For information about checking connectivity to the Cloud APM server, see Network connectivity.

Non-root user installation
You must have read, write, and execute permissions for the installation directory. Otherwise, the installation is canceled. For more information about non-root user installation, see “Installing agents as a non-root user” on page 142.

70-character limitation for installation path
The installation directory and the path to it must be no more than 70 characters.

Setting the *CANDLEHOME* environment variable

If you used the ITM Agent Converter to install and configure an agent on the same managed system before, the *CANDLEHOME* environment variable changed to that directory where you installed the agent with the Agent Converter. Before you install and configure a native Cloud APM agent, you must set the *CANDLEHOME* environment variable to a different directory, otherwise, the native Cloud APM agent cannot start.

Ensure that the binary for bc is available on your system

The binary for bc is required to run `prereq_checker` when installing agents. But bc is missing on some Linux platforms, for example, SUSE Linux Enterprise Server 15 and CentOS 7.6. You must install bc and set it in the PATH environment variable. You can run `# which bc` to check. If it is available, you can see the following message:

```
# which bc
/usr/bin/bc
```

If bc is not found, you can see the following message:

```
# which bc
which: no bc in (/sbin:/usr/sbin:/usr/local/sbin:/root/bin:/usr/local/bin:/usr/bin:/bin)
```

Specific operating systems

Red Hat Enterprise Linux (RHEL) 8

The libnsl.so.1 package is needed on RHEL 8

By default, `libnsl.so.1` is not installed in Red Hat Enterprise Linux release 8.0. Without this package, no agent can be installed successfully. Have your administrator set up a yum repository for you, and then run this command:

```
yum install libnsl
```

After successful installation, you can see `/usr/lib64/libnsl.so.1`.

Note: The `libnsl.so.1` package is required only for agents. You do not need to do this step for data collectors.

Bypassing the prerequisite scanner for some agents

Before the prerequisite scanner is updated to be compatible with the latest requirements, for some agents, you can bypass the prerequisite scanner. For suitable scenarios and instructions, see "Bypassing the prerequisite scanner" on page 146.

Note: You do not need to do this step for data collectors.

SUSE Linux Enterprise 15

The binary for bc is needed on SUSE Linux Enterprise 15

The binary for bc is required to run `prereq_checker` when installing agents. By default, bc is not installed on SUSE Linux Enterprise 15. You must install bc and set it in the PATH environment variable. You can run `# which bc` to check. If it is available, you can see the following message:

```
# which bc
/usr/bin/bc
```

If bc is not found, you can see the following message:

```
# which bc
which: no bc in (/sbin:/usr/sbin:/usr/local/sbin:/root/bin:/usr/local/bin:/usr/bin:/bin)
```

CentOS 7.6

The binary for bc is needed on CentOS 7.6

The binary for bc is required to run `prereq_checker` when installing agents. By default, `bc` is not installed on CentOS 7.6. You must install bc and set it in the PATH environment variable. You can run `# which bc` to check. If it is available, you can see the following message:

```
# which bc
/usr/bin/bc
```

If bc is not found, you can see the following message:

```
# which bc
which: no bc in (/sbin:/usr/sbin:/usr/local/sbin:/root/bin:/usr/local/bin:/usr/bin:/bin)
```

CentOS 8.1

The libnsl.so.1 package is needed on CentOS 8.1

By default, `libnsl.so.1` is not installed in CentOS 8.1. Without this package, no agent can be installed successfully. Have your administrator set up a yum repository for you, and then run this command:

```
yum install libnsl
```

After successful installation, you can see `/lib64/libnsl.so.1`.

Note: The `libnsl.so.1` package is required only for agents. You do not need to do this step for data collectors.

Specific agents

The following preinstallation tasks are applicable to the specified agents:

DataPower agent

You must run the **`ulimit -d unlimited`** command to ensure that the *max data segment size* system environment variable is set to **`unlimited`**. This agent cannot be installed on the same machine as the DataPower appliance that you want to monitor.

DataStage agent

1. Enable parameters in the `DSODBConfig.cfg` file. Complete the following steps:

   a. Open the `DSODBConfig.cfg` file at the following location in an editor:

      `infosphere_information_server_install_dir/Server/DSODB`

   b. Uncomment the following parameters by removing # symbol:

      `MonitorLinks`=1
      `JobRunUsage`=1
      `ResourceMonitor`=1
      `DSODBON`=1

   c. Edit values of these parameters equal to 1.

2. Copy the JDBC driver of the database that is used for metadata repository configuration on the agent computer.

   a. Type 4 JDBC 4, or later. Example: `db2jcc4.jar`

   b. Type 4 JDBC driver for Oracle. Example: `ojdbc6.jar`

   c. JDBC driver for MS SQL:

      - `Sqljdbc41.jar` requires a JRE of 7 and supports the JDBC 4.1 API.
      - `Sqljdbc42.jar` requires a JRE of 8 and supports the JDBC 4.2 API.

HTTP Server agent

If you install this agent as a root user, you must use the same user ID to run and configure the agent.

If you install and run the agent as a non-root user, the non-root user must have the same user ID as the user who started the IBM HTTP Server. Otherwise, the agent has problems with discovering the IBM HTTP Server. You can use the same user ID to run and configure the agent.

Linux KVM agent

The Monitoring Agent for Linux KVM is a multi-instance and multi-connection agent and supports connections to the Enterprise Linux based KVM hypervisor and Red Hat Enterprise Virtualization Manager (RHEV-M) environments. You can create multiple instances of this agent to monitor multiple hypervisors in an RHEV-M or KVM hypervisor environment. You can monitor virtualized workloads and analyze the resource capacity across different virtual machines. To connect the agent to a virtual machine in the KVM hypervisor environment, you must install the prerequisites: `libvirt*.rpm` and Korn Shell Interpreter (pdksh). The agent collects metrics by connecting remotely to a libvirt hypervisor that manages the virtual machines.

Microsoft SQL Server agent

To monitor a Microsoft SQL environment, the Microsoft SQL Server and Microsoft SQL ODBC driver must be installed before you install the Monitoring Agent for Microsoft SQL Server. For example, to install the ODBC driver on Red Hat Enterprise Linux, use the following command:

```
sudo yum install unixODBC
sudo yum install msodbcsql17
```

To complete the execution of prerequisite checker, the agent needs to be configured on the Cloud Application Performance Management Version 8.1.4.0 Server Interim Fix 15 (8.1.4.0-IBM-APM-SERVER-IF0015. tar) or later.

MongoDB agent

You must install and configure the MongoDB agent on the system where the MongoDB database server is installed.

MySQL agent

To monitor a MySQL environment, the MySQL server and MySQL JDBC driver must be installed before you install the Monitoring Agent for MySQL. For example, to install the JDBC driver on Red Hat Enterprise Linux, use the following command:

```
yum install mysql-connector-java
```

After you start the agent installation and during the prerequisite check for the MySQL package name, you might get a warning if a provider other than Red Hat is used, such as Oracle. If the MySQL Server and JDBC driver are available, the warning does not cause the installation to fail, and you can disregard the message. Sample output:

```
Scenario: Prerequisite Scan
KSE - Monitoring Agent for MySQL Prerequisite Checker [version 01000005]:
Property                      Result Found       Expected
========                      ====== =====       ========
os.package.mysql-server         WARN Unavailable mysql-server-5.1.12+
os.package.mysql-connector-java WARN Unavailable mysql-connector-java-5.
1.12+
```

Node.js agent

The version of Node.js that you use to run your monitored application must be the same as the default installed version.

Currently Node.js v5 is not supported.

OpenStack agent

Before you can use the OpenStack agent, you must have the following software on the server where you install the agent:

- Python 2.6.0 or later, or Python 2.7.0 or later
- Latest OpenStack clients:

  - OpenStack
  - Keystone
  - Neutron
  - Swift

  To install the OpenStack command-line clients, see Install the OpenStack command-line clients.

- Paramiko library for remote access in Python.

  Note: If you want to install the OpenStack agent on a clean RedHat Linux server, before you install the Paramiko library, run the following command to install the required software:

  ```
  wget https://ftp.dlitz.net/pub/dlitz/crypto/pycrypto/pycrypto-2.6.1.tar.gz
  yum install gcc/openssl-devel/libffi-devel
  ```

- KornShell

Oracle Database agent

On Red Hat Enterprise Linux version 5 and version 6 and SUSE Linux Enterprise Server version 11 and version 12 x64, if the Oracle Database agent monitors the Oracle database remotely, you must install the Oracle instant clients first. Install the Oracle instant clients from Oracle Technology Network - Instant Client Downloads.

The instant client v10.x, v11.x, and v12.x are supported by the Oracle Database agent.

PHP agent

If the PHP application is deployed by using the root user, you must use the root user to install, configure, start, or stop the agent. If the PHP application is deployed by using a non-root user, you can use root user or the same non-root user to install, configure, start, or stop the agent.

You must have an existing WordPress application installed. The PHP agent monitors WordPress V3.7.1 or later.

The agent evaluates only the performance of PHP requests in WordPress applications. CSS and JS loading are not evaluated.

The agent does not use URL arguments to identify URLs.

Python data collector

The Python data collector monitors Django applications.

Response Time Monitoring Agent

Before you install the Response Time Monitoring agent, review the installation planning section here: “Planning the installation ” on page 679.

SAP HANA Database agent

1. Install SAP HANA database client HDBSQL version 1.00.102.06 or later on the Linux system.

   Important: For the RHEL 5.x 64-bit operating system, install the Linux SUSE 9 on x86_64 64bit SAP HANA database client instead of Linux on x86_64 64bit. For the RHEL 6.x, or later 64-bit operating systems, install the Linux on x86_64 64bit SAP HANA database client.

2. Run the following command to add the path of the installation directory to the **`LD_LIBRARY_PATH`** environment variable:

   `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:`*`install_directory_path`*

   Example: `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/sap/hdbclient`, where `/usr/sap/hdbclient` indicates the installation path of the SAP HANA database client.

Important:

If the installation path of the SAP HANA database client is not added to the **`LD_LIBRARY_PATH`** environment variable, the prerequisite scanner returns the FAIL result.

The environment variable that you added by using the export command persists only for a particular session of the terminal. Therefore, ensure that you run the agent installation script from the same terminal that was used for adding the environment variable.

Synthetic Playback agent

To install the Synthetic Playback agent, the operating system user requires the following permissions:

- Enable read and execution permission for the installation image
- Enable write permission for the agent home

To run the Synthetic Playback agent, the operating system user requires the following permissions:

- Enable read, write, and execution permission for the agent installation location and its subdirectories and files.
- Enable permission to run Mozilla Firefox.
- Ensure that the Mozilla Firefox execution binary is in the PATH environment variable of the user's profile.

Before you install the Synthetic Playback agent, you must complete the following steps:

1. Synchronize agent installation locations with the Cloud APM console.
2. Install Mozilla Firefox and the Xvfb display server.
3. Verify that the Xvfb display server is working. Run the command:

   ```
   # Xvfb -ac
   ```

   There should be no error output.
4. Check that the Xvfb process is running. Run the following command:

   ```
   # ps -ef|grep Xvfb
   ```

   Sample output:

   ```
   root 7192 1 0 Jan14 ? 00:00:14 Xvfb -ac
   root 20393 17900 0 02:05 pts/0 00:00:00 grep -i xvfb
   ```
5. Stop the Xvfb process. Run the following command:

   ```
   # kill -9 7192
   ```
6. Navigate to *install_dir*/etc/hosts and edit the beginning of the hosts file to include the following parameters:

   ```
   127.0.0.1 localhost
   ```

   Then, save and close the hosts file.

WebSphere Applications agent

- Before the agent is installed, the prerequisite checker checks that *ulimit* is set to **524000** on the Linux system. You must run the **ulimit -d 524000** command to ensure that the *max data segment size* system environment variable is set to **524000**.
- libstdc++.so.5 is required when installing the WebSphere Applications agent on Linux for IBM Z. On SUSE Linux Enterprise Server 12 and 15 for IBM Z, libstdc++.so.5 is not installed by default. Ask your system admin to install before deploying the WebSphere Applications agent. Otherwise, you will meet the following error:

  ```
  KYN - WAS Monitoring Agent [version 07301410]:

  Property              Result       Found            Expected
  ========              ======       =====            ========
  os.lib.libstdc++_64   FAIL         Unavailable      regex{libstdc++.so.5}
  ```

Installing agents

You can install any combination of monitoring agents on a managed system. For example, if you install the Ruby agent to monitor Ruby On Rails applications, you might want to also install the Response Time Monitoring Agent, the Linux OS agent, or both agents. With the Response Time Monitoring agent, you can gather more response time information for your Ruby applications. With the Linux OS agent, you can monitor other aspects of the system, such as the overall CPU, memory, and disk.

The offering determines which monitoring agents are available for installation. For a list of the agents included in each offering, see "Capabilities" on page 56.

For a list of the agents that run on Linux systems, see "Installing agents on Linux systems" on page 128.

Before you begin

Download the agents. See "Downloading your agents and data collectors" on page 105.

Review the information in "System requirements" on page 85 to make sure that you have met the requirements for the agents you plan to install.

Review the agent preinstallation tasks before you install the agents. For details, see "Preinstallation on Linux systems" on page 129.

Note: Java Runtime is installed only when the agent requires it and is not always available. Also, ksh is no longer required for agent installation, except for installation of the Summarization and Pruning agent, which is installed during Cloud APM server installation. SELinux in enforcing mode is supported.

About this task

You can install monitoring agents as a root user or non-root user. If you do not have root privileges and you want to install a monitoring agent, you can install the agent as a non-root user, see "Installing agents as a non-root user" on page 142. Also, you can install the agent as a non-root user if you are a host administrator and you do not want to run the monitoring agent as a root user. Installation flow is the same as for a root user.

Agent coexistence is supported. You can install IBM Cloud Application Performance Management agents on the same computer where IBM Tivoli Monitoring agents are installed. However, both agent types cannot be installed in the same directory. For more information about agent coexistence see "Cloud APM agent and Tivoli Monitoring agent coexistence" on page 930.

Procedure

1. Open a terminal shell session on the Red Hat Enterprise Linux system.
2. On your system, navigate to the directory where you downloaded the .tar file

   The agents must be installed on the system where the application that you want to monitor is installed. If needed, transfer the installation archive file to the system to be monitored. The archive file contains the agents and installation script.

   Remember: Make sure that the directory does not contain an older version of the archive file.
3. Extract the installation files by using the following commands, which depend on your offering:

   ```
   tar -xf ./installation files.tar
   ```

   where *installation files* is the installation file name for your offering.

   The installation script is extracted to a directory named for the archive file and version. For example: *offering*_Agent_Install_8.1.4.0. Agent binary and configuration-related files are extracted into subdirectories within that directory.
4. Run the installation script from the directory that is named for the archive file and version:

   ```
   ./installAPMAgents.sh
   ```

   To install the agents in silent mode, see "Installing agents silently" on page 145.

5. Specify whether to install individual agents, a combination of the agents, or all of the agents.
6. Depending on whether you are installing or upgrading the agents, take one of the following steps:
   - If you are installing the agents, specify a different agent installation home directory or use the applicable default directory:
     - `/opt/ibm/apm/agent`
   - If you are upgrading the agents, after you are prompted for the agent installation home directory, enter the installation directory of the previous version of the agents.
     a. If an older version of the agents exists in the `/opt/ibm/apm/agent` directory, you must specify a new installation directory. In the next step, you are asked if you want to migrate agent configuration from the `/opt/ibm/apm/agent` directory.
     b. If you confirm that you want to migrate the agent configuration from the old installation directory (`/opt/ibm/ccm/agent`) to the new installation directory, for example, `/opt/ibm/apm/agent`, you must start the agent in the new installation location.

        Restriction: The older version of the agent is stopped automatically in the old installation location, but it is not started automatically in the new installation location.
     c. After installation is complete and you verify that the agent works in the new installation directory, you must uninstall the older version of the agent from the `/opt/ibm/ccm/agent` directory. If you want to remove all agents, run the **`/opt/ibm/ccm/agent/bin/smai-agent.sh uninstall_all`** command.
7. When you are asked if you accept the license agreement, enter 1 to accept the agreement and continue, or enter 2 to decline.

   After you enter 1 (accept), a prerequisite scan of your environment starts and takes a few moments to complete. If any requirements are missing, a message directs you to a log file with the reason for the failure. A missing prerequisite, such as a missing library or insufficient disk space, stops the installation. You must address the failure, and start the installation script again.

   Note: If the installation exits with the following message, check whether the Server service is started (Start -> Administrative Tools -> Services). If not, start the Server service and run `installAPMAgents.bat` again.

   ```
   This script [installAPMAgents.bat] must be run as Administrator.
   ```
8. If you installed the agents by using a non-root user ID, you must update the system startup scripts (see “Installing agents as a non-root user” on page 142).
9. After installation is complete and the command line is available, you can repeat the steps in this procedure to install more monitoring agents on the managed system.

What to do next

Configure the agent as required. If your monitoring agent requires configuration as described in Chapter 5, “Agent and data collector deployment,” on page 113 or if you want to review the default settings, see Chapter 7, “Configuring your environment,” on page 159.

- If you are using a forward proxy because your firewall does not allow transparent outbound HTTPS connections to external hosts, you must edit the agent environment configuration file. For instructions, see “Configuring agents to communicate through a forward proxy” on page 159.
- If you upgraded an agent from a previous version, identify any reconfiguration or migration tasks that you must complete before logging in to the Cloud APM console. For information about those tasks, see “Upgrading your agents” on page 1111. After an upgrade, you must restart any agent that is not both automatically configured and started by the installer.

To start an agent, run the following command:

```
./name-agent.sh start
```

For information about the monitoring agent commands, including the name to use, see "Using agent commands" on page 177. For information about which agents are started automatically and manually, see Chapter 5, "Agent and data collector deployment," on page 113.

After an upgrade, you must restart any agent that is not both automatically configured and started by the installer.

After you configure and start the agent, view the data that the agent is collecting.

- If you are not logged in, follow the instructions in "Starting the Cloud APM console" on page 957.
- If you want to view managed systems from your IBM Tivoli Monitoring domain in the Application Performance Dashboard, complete the tasks that are described in "Integrating with IBM Tivoli Monitoring V6.3 " on page 929.
- Restart the apmui service on the Cloud APM server so that agent online help updates are displayed in the Cloud APM console. The apmui service is restarted by using the `apm restart apmui` command.

Installing agents on Windows systems

You can install some of the Cloud APM monitoring agents on Windows systems.

The following monitoring agents are supported on Windows 64-bit systems. Where indicated, agents are also supported on Windows 32-bit systems.

- Monitoring Agent for Amazon EC2
- Monitoring Agent for AWS Elastic Load Balancer
- Monitoring Agent for Azure Compute
- Monitoring Agent for Cassandra
- Monitoring Agent for Cisco UCS
- Monitoring Agent for Citrix Virtual Desktop Infrastructure
- Monitoring Agent for Db2
- Monitoring Agent for Hadoop
- Monitoring Agent for HTTP Server*
- Monitoring Agent for IBM Cloud
- Monitoring Agent for IBM Integration Bus
- Monitoring Agent for Internet Services*
- Monitoring Agent for MQ Appliance
- Monitoring Agent for JBoss
- Monitoring Agent for MariaDB
- Monitoring Agent for Microsoft Active Directory*
- Monitoring Agent for Microsoft Cluster Server*
- Monitoring Agent for Microsoft Exchange Server
- Monitoring Agent for Microsoft Hyper-V Server
- Monitoring Agent for Microsoft Internet Information Services
- Monitoring Agent for Skype for Business Server (formerly known as Microsoft Lync Server)*
- Monitoring Agent for Microsoft .NET
- Monitoring Agent for Microsoft Office 365
- Monitoring Agent for Microsoft SharePoint Server
- Monitoring Agent for Microsoft SQL Server*
- Monitoring Agent for MySQL
- Monitoring Agent for NetApp Storage

- Monitoring Agent for Oracle Database
- Monitoring Agent for PostgreSQL
- Monitoring Agent for RabbitMQ
- Monitoring Agent for SAP Applications
- Monitoring Agent for SAP HANA Database
- Monitoring Agent for SAP NetWeaver Java Stack
- Monitoring Agent for Siebel
- Monitoring Agent for Sterling Connect Direct
- Monitoring Agent for Sterling File Gateway
- Monitoring Agent for Sybase Server
- Monitoring Agent for Tomcat
- Monitoring Agent for VMware VI
- Monitoring Agent for WebLogic
- Monitoring Agent for WebSphere Applications
- Monitoring Agent for WebSphere MQ
- Monitoring Agent for Windows OS*
- Response Time Monitoring Agent*

* Supported on both 64-bit and 32-bit Windows systems.

Preinstallation on Windows systems

You must complete the required preinstallation tasks before you install agents on Windows systems. Some preinstallation tasks are agent-specific and other tasks apply to multiple agents.

Note: These requirements are in addition to the requirements identified in the Software Product Compatibility Reports for your agent: For the current version requirements and dependencies for your agent, see the IBM Cloud Application Performance Management Software Product Compatibility Report for agents on Windows.

All agents

The following preinstallation tasks are applicable to all agents:

Test connectivity
Before you install agents, ensure that your system can communicate with the Cloud APM server. For information about checking connectivity to the Cloud APM server, see Network connectivity.

Installing from the command prompt on a local drive
Use the Windows command prompt to start the installation script. Do not use Windows PowerShell to start the installation script.

Copy the installation files to a local disk or a mapped network drive, and then start the installation script. Do not start the installation script from a network location.

Start the installation script from a new command prompt. Do not start the installation script from an existing command prompt because the command prompt might have outdated environment variables.

Setting the *CANDLEHOME* environment variable
If you used the ITM Agent Converter to install and configure an agent on the same managed system before, the *CANDLEHOME* environment variable changed to that directory where you installed the agent with the Agent Converter. Before you install and configure a native Cloud APM agent, you must set the *CANDLEHOME* environment variable to a different directory, otherwise, the native Cloud APM agent cannot start.

Specific agents

The following preinstallation tasks are applicable to the specified agents:

DataStage agent

1. Enable parameters in the `DSODBConfig.cfg` file. Complete the following steps:

   a. Open the `DSODBConfig.cfg` file at the following location in an editor:

   ```
   infosphere_information_server_install_dir\Server\DSODB
   ```

   b. Uncomment the following parameters by removing # symbol:

   ```
   MonitorLinks=1
   JobRunUsage=1
   ResourceMonitor=1
   DSODBON=1
   ```

   c. Edit values of these parameters equal to 1.

2. Copy the JDBC driver of the database that is used for metadata repository configuration on the agent computer.

   a. Type 4 JDBC 4, or later. Example: `db2jcc4.jar`

   b. Type 4 JDBC driver for Oracle. Example: `ojdbc6.jar`

   c. JDBC driver for MS SQL:

   - `Sqljdbc41.jar` requires a JRE of 7 and supports the JDBC 4.1 API.
   - `Sqljdbc42.jar` requires a JRE of 8 and supports the JDBC 4.2 API.

IBM Integration Bus agent

Make sure the user id to install the IBM Integration Bus agent is in the `mqbrkrs` user group.

Internet Service Monitoring

For the Internet Service Monitoring, you must apply IBM Cloud Application Performance Management 8.1.4.0 core framework Interim Fix 3 on the APM Server from here and then preconfigure the agent. The Agent and the bridge module uses ports 9510 and 9520. In case these ports are already in use the installation will not progress.

Note:

- For existing users it is recommended to install Internet Service Monitoring agent on 64-bit platforms either Windows or Linux rather than upgrading the agent on Windows 32-bit platform to a newer version.
- Internet Service Monitoring Agent do not support Windows 2008 R2 on Windows 64-bit platform.

MySQL agent

For the Monitoring Agent for MySQL, you must install the MySQL server and MySQL JDBC driver before you install the MySQL agent on that system. To install the JDBC driver, see MySQL Connector/J JDBC driver.

Oracle Database agent

If the Oracle Database agent monitors the Oracle database remotely, you must install the Oracle instant clients first from Oracle Technology Network - Instant Client Downloads on the following systems:

- Windows Server 2012 64-bits
- Windows Server 2012 R2 64-bits
- Windows Server 2008 R2 Datacenter 64-bits
- Windows Server 2008 R2 Enterprise 64-bits
- Windows Server 2008 R2 Standard 64-bits

The instant clients v10.x, v11.x, and v12.x are supported by the Oracle Database agent.

Response Time Monitoring Agent

Before you install the Response Time Monitoring agent, review the installation planning section here: "Planning the installation " on page 679.

SAP HANA Database agent

1. Install SAP HANA database client HDBSQL version 1.00.102.06 or later on the Windows system.
2. Add the installation path of the SAP HANA client to the **PATH** environment variable.

   Example: Add `C:\Program Files\sap\hdbclient` to the **PATH** environment variable, where `C:\Program Files\sap\hdbclient` indicates the installation path of the SAP HANA database client.

Tomcat agent

1. Java SDK is installed on the Tomcat server where the agent is installed.
2. The SDK path is added to the *PATH* variable directly or by using the **`set path`** command before installing the agent.
3. The **JAR** command is working.

Installing agents

You can install any combination of monitoring agents on a managed system. For example, if you install the Monitoring Agent for MySQL for monitoring MySQL servers, you might want to also install the Response Time Monitoring Agent to gather more response time information for your Ruby applications. You might also want to install the Monitoring Agent for Windows OS to monitor other aspects of your system, such as the overall CPU, memory, and disk.

Your offering determines which monitoring agents are available for installation. For a list of the agents included in each offering, see "Capabilities" on page 56.

For a list of the agents that run on a Windows system, see "Preinstallation on Windows systems" on page 138.

Before you begin

Download the agents. See "Downloading your agents and data collectors" on page 105.

Review the information in "System requirements" on page 85 to make sure that you have met the requirements for the agents you plan to install.

Review the agent prerequisite tasks before you install the agents. For details, see "Preinstallation on Windows systems" on page 138.

About this task

Ensure that you have adequate permission to run the agent installation script and agent commands. You must be logged in using one of the following user account types:

- default Windows administrator user account
- administrator user account
- user account, which is a member of the administrators group
- user account, which is registered as an administrator in Active Directory services

Agent coexistence is supported. You can install IBM Cloud Application Performance Management agents on the same computer where IBM Tivoli Monitoring agents are installed. However, both agent types cannot be installed in the same directory. For more information about agent coexistence, see "Cloud APM agent and Tivoli Monitoring agent coexistence" on page 930.

Procedure

Complete these steps to install monitoring agents on VMs and systems where the Windows operating system is installed:

1. On your system, navigate to the directory where you downloaded the compressed file.
2. Extract the agent installation files for your offering (or offerings) to the location where you want to install the monitoring agent software.
   The .bat installation script is extracted to a directory named for the archive file and version. For example: `offering_Agent_Install_8.1.4.0`. Agent binary and configuration-related files are extracted into subdirectories within that directory.
3. Open a command prompt as administrator.
   a) From the **Start** menu, type `command` in the search box.
   b) Right-click **Command Prompt** from the list that displays and select **Run as administrator**.
4. From the command prompt, run the installation script with Administrator privileges from the directory that is named for the archive file and version:

   ```
   cd offering_Agent_Install_version
   installAPMAgents.bat
   ```

   Restriction: For the WebSphere Applications agent, the Administrator privileges must be the same privileges that were used to install the WebSphere Application Server.

   To install the agents in silent mode, see "Installing agents silently" on page 145.
5. If you are installing the agents, provide the name of the installation directory.

   The default installation path is `C:\IBM\APM`. The name of the installation directory cannot exceed 80 characters or contain non-ASCII, special, or double-byte characters. Directory names in the path can contain only the following characters:
   `abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ _\:0123456789()~-./`.

   Note: When short file name creation (*8dot3Name*) is disabled, if directory names in the path contain spaces, installation is not supported.

   If you are upgrading the agent, this step is skipped, and the agent installs into the previous installation directory.
6. When you are asked if you accept the license agreement, enter 1 to accept the agreement and continue, or enter 2 to decline.

   After you enter 1 (accept), a prerequisite scan of your environment starts and takes a few moments to complete. If any requirements are missing, a message directs you to a log file with the reason for the failure. A missing prerequisite, such as a missing library or insufficient disk space, stops the installation. You must address the failure, and start the installation script again.

   Note: If the installation exits with the following message, check whether the Server service is started (Start -> Administrative Tools -> Services). If not, start the Server service and run `installAPMAgents.bat` again.

   ```
   This script [installAPMAgents.bat] must be run as Administrator.
   ```
7. After installation is complete and the command prompt is available, repeat these steps to install more monitoring agents.

What to do next

Configure your agents as required. To check if your monitoring agent requires manual configuration, see Chapter 5, "Agent and data collector deployment," on page 113. For configuration instructions, or, if you want to review default configuration settings, see Chapter 7, "Configuring your environment," on page 159.

Before installing new agents, Windows Installer temporarily stops all agents currently running in the installed product location. After installation completes, the installer restarts any stopped agents. You must manually restart any monitoring agent that is not automatically started by the installer.

- If you are using a forward proxy because your firewall does not allow transparent outbound HTTPS connections to external hosts, you must edit the agent environment configuration file. For instructions, see "Configuring agents to communicate through a forward proxy" on page 159.
- If you upgraded an agent from a previous version, identify any reconfiguration or migration tasks that you must complete before you log in to the Cloud APM console. For information about those tasks, see "Upgrading your agents" on page 1111.

Use one of the following methods to start the agent:

- Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Cloud Application Performance Management**. Right-click on an agent and click **Start**.
- Run the following command

```
name-agent.bat start
```

For information about the monitoring agent commands, including the name to use, see "Using agent commands" on page 177. For information about which agents are started automatically and manually, see Chapter 5, "Agent and data collector deployment," on page 113

After an upgrade, you must restart any agent that is not both automatically configured and started by the installer.

After you configure and start the agent, view the data that the agent is collecting.

- If you are not logged in, follow the instructions in "Starting the Cloud APM console" on page 957.
- If you want to view managed systems from your IBM Tivoli Monitoring domain in the Application Performance Dashboard, complete the tasks that are described in "Integrating with IBM Tivoli Monitoring V6.3 " on page 929.
- Restart the apmui service on the Cloud APM server so that agent online help updates are displayed in the Cloud APM console. The apmui service is restarted by using the `apm restart apmui` command.

Installing agents as a non-root user

If you do not have root privileges and you want to install a monitoring agent, you can install the agent as a non-root user. Also, you can install the agent as a non-root user if you are a host administrator and you do not want to run the monitoring agent as a root user. Installation flow is the same as for a root user. After a non-root installation, run the **`UpdateAutoRun.sh`** script with root user or sudo user access.

Before you begin

To uniquely identify the computer system, the Linux OS agent must identify the computer system board Universal Unique Identifier (UUID), manufacturer, model, and serial number. This information is required before the agent is added to an application in the Cloud APM console.

Obtain the computer system information by verifying the following entities exist on the computer system:

1. Check whether the **`/usr/bin/hal-get-property`** command is installed on the computer system and the hald process (HAL daemon) is running. If the command is not installed, continue to step 2. If the command is installed, skip step 2 and step 3. Note: If the OS version is Red Hat 7, the hald process is not available.
2. If the **`/usr/bin/hal-get-property`** command is not installed on the computer system, then confirm that the `/sys/class/dmi/id/product_uuid` file exists and contains the computer system UUID, and the user who installs the Linux OS agent has read access to this file. If this file does not exist, continue to step 3. If the file exists, skip step 3.
3. If the **`/usr/bin/hal-get-property`** command is not installed and the `/sys/class/dmi/id/product_uuid` file does not exist, you must ensure that the **`hostname`** or **`hostnamectl`** commands

return the fully qualified hostname. If these commands return the short hostname without the domain, you must set the fully qualified hostname by entering the**"hostname <fqhn>"** or **"hostnamectl set-hostname <fqhn>"** commands where *<fqhn>* must be replaced with the fully qualified hostname.

Note: The Linux OS agent retrieves this information periodically so the commands or files in the previous steps must remain in place even after installation.

Note: The Linux OS Agent does not support monitoring of Docker when running as non-root.

Procedure

1. Install your monitoring agents on Linux or UNIX systems, as described in "Installing agents on Linux systems" on page 128 and "Installing agents on UNIX systems" on page 122.
2. Optional: If you installed your agent as a selected user and want to configure the agent as a different user, run the `./secure.sh` script.

   For more information about the **`./secure.sh`** script, see "Configuring agents as a non-root user" on page 183 and Securing the agent installation files.

   For example: **`./secure.sh -g db2iadm1`**
3. Optional: Configure your monitoring agents on Linux or UNIX as necessary, see Chapter 7, "Configuring your environment," on page 159.
4. To update the system startup scripts, run the following script with root user or sudo user access: *`install_dir`*`/bin/UpdateAutoRun.sh`

What to do next

If you installed your agent as a non-root user and you want to configure the agent as the same user, no special action is required. If you installed your agent as a selected user and want to configure the agent as a different user, see "Configuring agents as a non-root user" on page 183.

If you installed and configured your agent as a non-root user and you want to start the agent as the same user, no special action is required. If you installed and configured your agent as a selected user and want to start the agent as a different user, see "Starting agents as a non-root user" on page 992.

Use the same user ID for agent installation and upgrades.

If you run the **`UpdateAutoRun.sh`** script as root user, the agent is configured to automatically start after the operating system restart. If you do not want this agent behavior, you can disable the automatic agent start. For more information, see "Disabling automatic agent start on UNIX and Linux systems" on page 184.

Securing the agent installation files

After you install monitoring agents as a non-root user on Linux or UNIX systems, you can run the `secure.sh` script to secure the agent installation by removing world write permissions and setting correct file ownership.

Before you begin

- You must have read, write, and execute permissions for the installation directory.
- Installation of the monitoring agents and any agent configuration must be completed on the system and the agents must be successfully started.
- If you are running agents as different user accounts, they must be members of the same group. (See option `-g`.)

About this task

Complete this step to lock down the file permissions in your installation. Options are available to require no root password, to specify a group name, and to view help for the command.

Procedure

- Run the following command from the *install_dir*/bin directory. Usage:

  ```
  secure.sh [-g common_group] [-n] [-h]
  ```

  - In the simplest mode, run the **./secure.sh** script, which removes world write permissions, and sets the current user and user's group as the file owners. If the script is run by a non-root user, the user is prompted for the root password.
  - If a non-root user runs the **./secure.sh** script with the -n option, this user is not prompted for a root password. In this case, changing file permissions and changing ownership are done by using this user's privileges. If the installation directory contains files that are owned by different users and the current user has no privileges to modify permissions and ownership of other user's files, this mode can fail.
  - If you want to set a certain group as the group owner, the owner must provide the -g option with a valid group name as an argument to that option. (See Example.)
    Run secure.sh -g *common_group*.
    The command changes ownership of the files and directories recursively.

    If the *common_group* group is not the user's primary group, you can set the *common_group* group to be the group owner of new files created in a directory, by running the following command:

    ```
    chmod g+s install_dir/bin/sub_dir
    ```

    where, *sub_dir* is any sub-directory, for example, /opt/ibm/ccm/agent.
  - Run the **./secure.sh** script with the –h option to get help information for the script.

Results

The installation directory allows access to only the user who ran the script or to only the users in the specified group.

Example

If user Alice is a member of the system group that is named "apmgroup", she can use the group to set file group ownership with the following command:

```
./secure.sh -g apmgroup
```

After the script is run, the group is set as "apmgroup" for all files in *install_dir* for the group.

What to do next

Running the **./secure.sh** script should result in the following permissions being set for the agents.

```
rwx rwx ---
```

After you run the script, check the permissions for the agent files. For example, for Response Time Monitoring, check the files in *install_dir*/*arch*/hu/lib/mod_wrt.so. If the permissions for these files are not set correctly, update the permissions manually. For example, for the Response Time Monitoring agent:

1. Set the permissions, run:

   ```
   chmod g+rx $AGENT_HOME/bin/rt-agent.sh
   ```

2. Set the user and group, run:

   ```
   chown newuser:newgroup $AGENT_HOME/bin/rt-agent.sh
   ```

Installing agents silently

Installing agents silently reduces installation time. To install a monitoring agent in silent mode, you must download an agent installation image archive file from the IBM Marketplace website, extract the agent installation files, prepare a silent response file, and run the installation script in silent mode.

Before you begin

1. Review the prerequisite tasks for installing the monitoring agents, and download and extract the agent installation files. For details, see Installing agents on UNIX systems, Installing agents on Linux systems, or Installing agents on Windows systems.
2. Complete the following steps to prepare a silent response file for installing agents:
   a. Locate the silent installation file for your offering (or offerings) `offering_silent_install.txt`, make a copy of this file, and open it in a text editor.
   b. Uncomment the license agreement.
   c. Complete one of the following steps to specify the agents that you want to install:
      - Uncomment the individual agents to be installed. For example:

        ```
        INSTALL_AGENT=os
        ```

        ```
        INSTALL_AGENT=ruby
        ```

      - Uncomment `INSTALL_AGENT=all` to install all agents.

   d. Uncomment `AGENT_HOME` and specify the directory where you want to install the agents.

      Remember: Linux If you are upgrading agents on a Linux system, you must not specify the `/opt/ibm/apm/agent` directory.

   e. Linux If you are upgrading the agents on a Linux system, uncomment `MIGRATE_CONF=yes`.
   f. Save the file.

Procedure

1. On the command line, change to the directory where you extracted the installation script and run the following command:

   ```
   cd offering_Agent_Install_version
   ```

2. Optional: This step is required ONLY for Solaris 10. You must create a soft link to ksh before you run the installation script on Solaris 10.

   a) Backup the `/bin/sh` command:

   ```
   mv /bin/sh /bin/sh.bkup_origin
   ```

   b) Create a soft link to ksh command:

   ```
   ln -s /bin/ksh /bin/sh
   ```

   c) Confirm that the result points to ksh:

   ```
   ls -l /bin/sh
   ```

3. Run the installation command:
   - Linux AIX

     ```
     ./installAPMAgents.sh -p path_to_silent_response_file
     ```

   - Windows

```
installAPMAgents.bat -p path_to_silent_response_file
```

The agents installation will fail on Windows if the prerequisite scanner can't obtain the type of disk where the agent will be installed to. If this occurs, you will see a fail result for the validDestLocation propery in the installation log file. In this case, you can skip the prerequisite scanning by adding SKIP_PRECHECK=1 to the installation command. For example:

```
installAPMAgents.bat -p path_to_silent_response_file SKIP_PRECHECK=1
```

Note: Windows When short file name creation (*8dot3Name*) is disabled, if directory names in the path contain spaces, installation is not supported.

Results
The agents are installed.

What to do next

Configure the agents. See the procedure and table of commands for Linux and UNIX systems and for Windows systems.

Bypassing the prerequisite scanner

When you install monitoring agents, a prerequisite scan of your environment starts and takes a few moments to complete. If any requirements are missing, a message directs you to a log file with the reason for the failure. In some installation scenarios, you might want to either ignore warning messages or completely bypass the prerequisite check.

About this task

There are two levels of failure messages, WARN and FAIL, and there are two levels of bypassing:

- Setting the **IGNORE_PRECHECK_WARNING** variable causes the installer to ignore the warning (WARN) messages.
- Setting the **SKIP_PRECHECK** variable causes the installer to ignore all failure messages.

If your agent installation failed and you received a warning (WARN) from the prerequisite checker, review the warning. If you want to continue with the installation, set **IGNORE_PRECHECK_WARNING** and install again.

In an environment where you have virtual machine images that serve as templates, the prerequisite scan that is undertaken before installation begins can be done on only the first template image. If a VM image passes the scan, the other VMs created from that image will also pass. You can save time by bypassing the prerequisite check for other VMs that were created from the same image. Set **SKIP_PRECHECK** variable and install again.

The **SKIP_PRECHECK** setting is also appropriate for the scenario where you have a new operating system that IBM Support or the Software Product Compatibility Reports indicate that it is supported but the prerequisite checker has not yet been updated. Be sure to first try to install the agent, check the log, and make sure that this new OS is the only item failing – and the only item that you are bypassing – because **SKIP_PRECHECK** causes the installer to bypass every item in the prerequisite checklist.

After downloading and extracting the installation files, complete this procedure to ignore the warning messages or to bypass the prerequisite scan.

Procedure

On the system where you plan to install monitoring agents, enter one of the following commands:

- Ignore the warning (WARN) messages during the prerequisite check:
  - Linux AIX `export IGNORE_PRECHECK_WARNING=1`

  - Windows `set IGNORE_PRECHECK_WARNING=1`
- Bypass the prerequisite scan:
  - Linux AIX `export SKIP_PRECHECK=1`
  - Windows `set SKIP_PRECHECK=1`

What to do next

To restore the default setting the next time you want to install the agent with the prerequisite scanner, turn off the **`IGNORE_PRECHECK_WARNING`** or **`SKIP_PRECHECK`** variable:

- Linux AIX `unset IGNORE_PRECHECK_WARNING`
- Windows `set IGNORE_PRECHECK_WARNING=`

or

- Linux AIX `unset SKIP_PRECHECK`
- Windows `set SKIP_PRECHECK=`

Uninstalling your agents

Uninstall a single agent or all the agents from a managed system.

Before you begin

For multi-instance agents, you must remove all agent instances before you uninstall the agent. Otherwise, agent entries are not cleared from the registry. To remove instances, run the following command:

- Windows *`name`*`-agent.bat remove `*`instance_name`*
- Linux AIX `./`*`name`*`-agent.sh remove `*`instance_name`*

where, *name* is the name of the agent and *instance_name* is the instance name. For more information, see "Using agent commands" on page 177. For a list of multiple-instance agents, see Table 7 on page 114.

For the following agents, an agent-specific task must be completed before you complete the uninstallation procedure:

- For the Monitoring Agent for HTTP Server, you must delete the `Include` statement in the `http.conf` file, for example, `"Include "/opt/ibm/apm/agent/tmp/khu/kvm65s2_8044.conf"`, before you restart the IBM HTTP server.
- For the Monitoring Agent for Python, run *`install_dir`*`/lx8266/pg/bin/uninstall.sh` to remove injection codes before you uninstall the agent.
- For the Monitoring Agent for PHP, run *`install_dir`*`/bin/lx8266/pj/lib/uninstall.`*`instance_name`*`.sh` to move injection codes before you uninstall the agent.
- For the Monitoring Agent for WebSphere Applications, you must unconfigure the data collector for all monitored server instances before you uninstall the agent. Follow the instructions in "WebSphere Applications agent: Unconfiguring the data collector" on page 149.

  For the WebSphere Applications agent, make sure that the user ID, which is used to uninstall the agent, has full read and write permissions to the `logs` and `runtime` directories and all their contained subdirectories and files within the data collector home directory. The data collector home directory is as follows:

  – Windows *`install_dir`*`\dchome\7.3.0.14.08`

  – Linux AIX *`install_dir`*`/yndchome/7.3.0.14.08`
- For the Node.js agent, you must remove the monitoring plug-in from your Node.js application before you uninstall the agent. Follow the instructions in "Node.js agent: Removing the monitoring plug-in" on page 157.

- For the Microsoft .NET agent, you must remove the data collector from your .NET applications before you uninstall the agent. Follow the instructions in "Microsoft .NET agent: Removing the .NET data collector" on page 158.
- For the IBM Integration Bus agent, if you configured transaction tracking for brokers with the agent provided user exit, you must remove the user exit before you uninstall the agent. Follow the instructions in "Removing the KQIUserExit user exit" on page 287.
- For the Internet Service Monitoring, go to `<candle_home>\BIN` and run the ism-agent.bat file with `uninstall` as an argument. In case you want to uninstall all monitoring agents on the server using smai-agent.bat, first run the ism-agent.bat with `uninstall` as an argument and then run the smai-agent.bat
- For the Monitoring Agent for SAP NetWeaver Java Stack, before you uninstall the agent, stop all SAP NetWeaver Java Stack agent instances by using the following command:
  - Windows `sap_netweaver_java_stack-agent.bat stop instance_name`

About this task

The Oracle agent on Windows systems can be uninstalled only by using the command prompt.

Procedure

1. On the VM or system where the monitoring agent (or agents) is installed, start a command line and change to the binary directory:
   - Linux AIX `*install_dir*/bin`
   - Windows `*install_dir*\BIN`

   where *install_dir* the installation directory of the monitoring agent or agents.
2. To uninstall a specific monitoring agent, enter the agent script name and the uninstall option where *name* is the agent script name:
   - Linux AIX `./*name*-agent.sh uninstall`
   - Windows `*name*-agent.bat uninstall`

   For a list of the agent script names, see "Using agent commands" on page 177.

   Remember: For the Monitoring Agent for Microsoft .NET, you must run the command with Administrator privileges.

   The monitoring agent is uninstalled from the managed system.

   If you have uninstalled all of your monitoring agents individually, continue to remove the framework files. See What to do next.
3. To uninstall all the monitoring agents from the managed system with a confirmation prompt, enter the script name and uninstall all option:
   - Linux AIX `./smai-agent.sh uninstall_all`
   - Windows `smai-agent.bat uninstall_all`

   A confirmation prompt is displayed. Type 1 to continue, or type 2 to cancel.
   All monitoring agents are uninstalled from the system or VM.
4. Linux AIX

   On Linux, and UNIX, to force the uninstall of all the monitoring agents without a prompt for confirmation, enter the script name and the force uninstall all option:

   ```
   ./smai-agent.sh uninstall_all force
   ```

What to do next

For the Monitoring Agent for HTTP Server, after you uninstall the agent, you must remove the following files manually:

- `/tmp/khu_cps.properties`
- `/tmp/httpserver-disc.error`

For the Monitoring Agent for Python:

1. Delete the Django `pyc` configuration file to ensure the restored Django `pyc` file generates its binary.
2. Restart the Apache server to remove the loaded middleware in the Apache processes.

For the Monitoring Agent for Ruby, to uninstall the diagnostics data collector:

1. Navigate to the home directory of your application, open its `Gemfile`, and remove the following line from the file: `gem 'stacktracer'`
2. Restart your Ruby on Rails application.
3. Uninstall the diagnostics data collector. Enter: `gem uninstall` *Gemfile*
4. Remove the runtime directory of the data collector. The default location of this directory is *install_dir*`/install-images/kkm/dchome`

For the Monitoring Agent for Microsoft .NET, complete these steps:

1. Remove the data collector `dll` files using one of the following options:
   - Reboot your operating system.
   - Try to delete the file *install_dir*`\qe\bin64\CorProfLog.dll`.

     A `File in Use` dialog is displayed. It identifies the .NET processes that are currently running.
   - Restart each of the .NET processes.
2. Restart your .NET applications.

WebSphere Applications agent: Unconfiguring the data collector

If you uninstall the WebSphere Applications agent before you unconfigure the data collector, the agent uninstallation fails. You can remove the data collector from an application server instance manually or by using the interactive utility or the silent unconfiguration process.

For instances monitored with PMI resource monitoring, unconfiguration is not available. Monitoring of PMI data continues while the server is available.

Unconfiguring the data collector interactively

If you no longer want the data collector to monitor one or more application server instances, you can unconfigure the data collector for them.

Before you begin

Use the user ID for configuring the data collector to unconfigure the data collector, which is also the user ID for installing the application server. Verify that this user ID has read and write permissions to the data collector home directory and all its sub-directories. The data collector home directory is as follows, where *install_dir* is the WebSphere Applications agent installation directory.

- Windows *install_dir*`\dchome\7.3.0.14.08`
- Linux AIX *install_dir*`/yndchome/7.3.0.14.08`

About this task

The unconfiguration utility (`unconfig.sh` or `unconfig.bat`) is a menu driven command-line utility for unconfiguring the data collector.

Procedure

1. Log in to the system as the user ID that is used to configure the data collector.
2. Navigate to the following `bin` directory:
   - Windows `install_dir\dchome\7.3.0.14.08\bin`
   - Linux AIX `install_dir/yndchome/7.3.0.14.08/bin`
3. Optional: Set the location of the Java home directory before you start the utility.

   For example:

   Linux AIX `export JAVA_HOME=/opt/IBM/AppServer80/java`

   Windows `set JAVA_HOME=C:\Progra~1\IBM\WebSphere\AppServer80\java`
4. Start the unconfiguration utility by issue the following command:

   Linux AIX `./unconfig.sh`

   Windows `unconfig.bat`
5. The utility searches for all server instances that are monitored by the data collector. Enter the number that corresponds to the application server instance to unconfigure for data collection or enter an asterisk (*) to unconfigure data collection for all application server instances. To specify a subset of servers, enter the numbers, separated by commas, that represent the servers. For example: `1,2,3`.

   Remember:

   - For a stand-alone environment, application server instances must be running during the configuration. (A WebSphere Application Server Liberty instance does not need to be running).
   - For a Network Deployment environment, the Node Agent and Deployment Manager must be running.
6. The utility prompts you to specify whether you want to create a backup of your current WebSphere Application Server configuration. Enter 1 to create a backup of the current configuration. Otherwise, enter 2 and skip to step 8.
7. The utility prompts you to specify the directory in which to store the backup of the configuration. Specify a directory in which to store the backup of the configuration or accept the default directory.

   The utility displays the name of the WebSphere home directory and the WebSphere profile for which a backup is created.
8. The utility indicates whether WebSphere Global Security is enabled for the WebSphere Application profile that you specified. If global security is not enabled, skip to step 10.
9. The utility prompts you to specify whether to retrieve security settings from a client properties file. Enter 1 to allow the utility to retrieve the user name and password from the appropriate client properties file and skip to step “10” on page 150. Otherwise, enter 2 to enter the user name and password.

   The data collector communicates with the WebSphere Administrative Services using the RMI or the SOAP protocol. If global security is enabled for a profile, you must specify the user ID and password of a user who is authorized to log in to the IBM WebSphere Application Server administrative console for the profile. Alternatively, you can encrypt the user name and password and store them in client properties files before configuring the data collector. You must use the `sas.client.props` file for an RMI connection, or the `soap.client.props` file for an SOAP connection.

   If you selected the option to back up the current WebSphere configuration, the utility starts backing up the configuration.
10. The utility unconfigures the data collector for the specified application server instances. A status message is displayed to indicate that the data collector was successfully unconfigured.
11. After the data collector unconfiguration completes, restart the application server instances.

The data collector configuration takes effect when the application server instances are restarted. PMI resource monitoring for the server instance is still available.

12. Optional: If you want to use resource monitoring for a server instance after unconfiguring the data collector, restart the monitoring agent by running the following commands:

    - Windows

      ```
      cd install_dir\bin
      was-agent.bat stop
      was-agent.bat start
      ```

    - Linux AIX

      ```
      cd install_dir/bin
      ./was-agent.sh stop
      ./was-agent.sh start
      ```

Results

The data collector is unconfigured for the specified application server instances.

Unconfiguring the data collector in silent mode

You can unconfigure the data collector using the unconfiguration utility in silent mode.

Before you begin

Use the user ID for configuring the data collector to unconfigure the data collector, which is also the user ID for installing the application server. Verify that this user ID has read and write permissions to the data collector home directory and all its sub-directories. The data collector home directory is as follows, where *install_dir* is the WebSphere Applications agent installation directory.

- Windows `install_dir\dchome\7.3.0.14.08`
- Linux AIX `install_dir/yndchome/7.3.0.14.08`

About this task

When you unconfigure the data collector in silent mode, you first specify configuration options in a properties file. A sample properties file, `sample_silent_unconfig.txt`, is packaged with the unconfiguration utility. The file is available in `bin` directory within data collector home directory.

Procedure

1. Log in to the system with the user ID that is used to configure the data collector.
2. Specify the configuration options in the `properties.txt` file.

   The following properties are available for unconfiguring the data collector in silent mode:

   WebSphere Application Server connecting settings

   was.wsadmin.connection.host
   Specifies the name of the host to which the wsadmin tool is connecting.

   WebSphere Application Server global security settings

   was.wsadmin.username
   Specifies the user ID of a user who is authorized to log on to the IBM WebSphere Application Server administrative console. This user must have the agent role on the application server.

   was.wsadmin.password
   Specifies the password that corresponds to the user specified in the `was.wsadmin.username` property.

WebSphere Application Server settings

was.appserver.profile.name
Specifies the name of the application server profile you want to unconfigure.

was.appserver.home
Specifies the WebSphere Application Server home directory.

was.appserver.cell.name
Specifies the WebSphere Application Server cell name.

was.appserver.node.name
Specifies the WebSphere Application Server node name.

Backup of the WebSphere Application Server configuration

was.backup.configuration
Specifies whether to back up the current configuration of the WebSphere Application Server data collector before unconfiguring the data collector. Valid values are `True` and `False`.

was.backup.configuration.dir
Specifies the location of the backup directory.

WebSphere Application Server runtime instance settings

was.appserver.server.name
Specifies an application server instance within the application server profile for which you want to unconfigure the data collector.

Tip: The silent response file can have multiple instances of this property.

3. Navigate to the following directory:
   - Windows *install_dir*\dchome\7.3.0.14.08\bin
   - Linux AIX *install_dir*/yndchome/7.3.0.14.08/bin
4. Run the following command:
   - Windows

   ```
   unconfig.bat -silent path_to_silent_file
   ```

   - Linux AIX

   ```
   unconfig.sh -silent path_to_silent_file
   ```

5. After the data collector unconfiguration completes, restart the application server instances.

   The data collector configuration takes effect when the application server instances are restarted. PMI resource monitoring for the server instance is still available.
6. Optional: If you want to use resource monitoring for a server instance after unconfiguring the data collector, restart the monitoring agent by running the following commands:
   - Windows

   ```
   cd install_dir\bin
   was-agent.bat stop
   was-agent.bat start
   ```

   - Linux AIX

   ```
   cd install_dir/bin
   ./was-agent.sh stop
   ./was-agent.sh start
   ```

Manually removing data collector configuration from an application server instance
To manually remove the data collector configuration from an application server instance, you must be able to connect to the application server by using the wsadmin tool. This is possible only if you are using

WebSphere Application Server Network Deployment and the Deployment Manager is running. If the WebSphere application server cannot start, you must restore the WebSphere application server from the backup taken when you run the configuration utility.

About this task

You can manually remove the data collector configuration from an application server instance, if any of the following conditions apply:

- In a non-Network Deployment environment, you manually added the data collector configuration to the application server instance and you want to unconfigure data collection. The application server instance must be running.
- In a Network Deployment environment, you manually added the data collector configuration to the application server instance and you want to unconfigure data collection. The Node Agent and Deployment Manager on the application server must be running.
- In a Network Deployment environment, you configured the application server instance for data collection manually and the application server fails to start. The Node Agent and Deployment Manager on the application server must be running.

If you configured a stand-alone application server instance for data collection either manually or with the configuration or migration utility and the application server fails to start, you must restore your WebSphere Application Server configuration with your backup configuration. For more information, see "Restoring the application server configuration from a backup" on page 872.

Remember:

- You must make manual changes to the WebSphere application server configuration for data collectors as the WebSphere administrative user.
- Making manual changes to the WebSphere application server for data collection must be performed by an experienced WebSphere administrator only. Any error in the manual configuration change can result in the application server not starting.
- If you manually configure the data collector to monitor application server instances, you cannot use the unconfiguration utility to unconfigure the data collector.

Procedure

To manually remove the data collector configuration, complete the following procedure:

1. Log in to the WebSphere Administration Server Console.
2. Click **Servers**.
3. Expand **Server Type** and select **WebSphere application servers**.
4. Click the name of the server.
5. In the Configuration tab, go to **Server Infrastructure** > **Java and Process Management** > **Process Definition** > **Java Virtual Machine** > **Additional Properties: Custom Properties**.
6. Remove any of the following JVM Custom Properties, if they are present:
   - `am.home`
   - `ITCAM.DC.ENABLED`
   - `TEMAGCCollector.gclog.path`
   - `com.ibm.tivoli.itcam.toolkit.ai.runtimebuilder.enable.rebuild`
   - `com.ibm.tivoli.jiti.injector.ProbeInjectorManagerChain. primaryInjectorFile`
7. Identify the JVM arguments that were added for the data collector.
   a) In the navigation pane, click **Environment** > **WebSphere Variables**.
   b) If you manually configured the application server for data collection, locate the JVM arguments you added manually.

If you configured the application server for data collection with the configuration utilities, compare the values of the **AM_OLD_ARGS** and **AM_CONFIG_JVM_ARGS** arguments to determine which arguments were added by the configuration utility.

8. Click **Server** > **Application Server** and select the appropriate server name.
9. In the Configuration tab, go to **Server Infrastructure** > **Java and Process Management** > **Process Definition** > **Java Virtual Machine**.
10. In **Generic JVM Arguments** field, remove the JVM arguments that you identified in Step 7 for the data collector.
11. Click **Apply** or **OK**.
12. In the **Messages** dialog box, click **Save**.
13. In the **Save to Master Configuration** dialog box, complete one of the following steps:
    - If you are under a Network Deployment environment, make sure that the **Synchronize changes with Nodes** check box is selected, and then click **Save**.
    - If you are not under a Network Deployment environment, click **Save**.
14. Remove environment entries that were added for the data collector.
    a) In the Configuration tab, go to **Server Infrastructure** > **Java and Process Management** > **Process Definition** > **Environment Entries**.
    b) Depending on the operating system, delete the following environment entry:
    - UNIX **LIBPATH**
    - Linux **LD_LIBRARY_PATH**
    - Windows **PATH**

    c) Remove the **NLSPATH** environment entry.
15. Click **Apply** or **OK**.
16. In the **Messages** dialog box, click **Save**.
17. In the **Save to Master Configuration** dialog box, complete one of the following steps:
    - If you are under a Network Deployment environment, make sure the check box **Synchronize changes with Nodes** is selected, and then click **Save**.
    - If you are not under a Network Deployment environment, click **Save**.
18. In the navigation pane, click **Environment** > **WebSphere Variables**.
19. Delete the following variables:
    - **AM_CONFIG_JVM_ARGS**
    - **AM_OLD_JVM_ARGS**
    - **ITCAMDCHOME**
    - **ITCAMDCVERSION**
20. In the **Messages** dialog box, click **Save**.
21. In the **Save to Master Configuration** dialog box, complete one of the following steps:
    - If you are under a Network Deployment environment, make sure the check box **Synchronize changes with Nodes** is selected, and then click **Save**.
    - If you are not under a Network Deployment environment, click **Save**.
22. If you configured the server instance for data collection with the data collector configuration tool, rather than manually, complete the following steps:
    a) Navigate to the `*dc_home*/runtime` directory.
    b) Rename the `$profile.$cell.$node.$server.input.properties` file to `$profile.$cell.$node.$server.input.properties.bak`.
23. If you are manually removing the data collector configuration from all application server instances in a profile, perform the following steps:

a) Navigate to the `$appserverhome/bin` directory.

b) Run the **`osgiCfgInit.sh/bat -all`** command on Windows systems or the **`osgiCfgInit.sh -all`** command on UNIX and Linux systems.

24. Restart the application server instance that was monitored by the data collector.

Manually unconfigure the data collector
After you manually configure the data collector for the WebSphere Applications agent, to remove data collection within the configured application server, you must manually unconfigure the data collector.

About this task

The following procedure applies only after you manually configure the data collector following the instructions in "Manually configure the data collector if the configuration utilities fail" on page 849. If you used the configuration utilities to configure the data collector, you must also use the unconfiguration utility to unconfigure the data collector. For instructions, see "Unconfiguring the data collector interactively" on page 149 or "Unconfiguring the data collector in silent mode" on page 151.

Procedure

- To manually unconfigure the data collector for the WebSphere application server, see "Manually unconfiguring the data collector for WebSphere Application Server traditional" on page 155.
- To manually unconfigure the data collector for the Liberty server, see "Manually unconfiguring the data collector for WebSphere Application Server Liberty" on page 156.

Manually unconfiguring the data collector for WebSphere Application Server traditional

Procedure

1. Log in to the WebSphere Administrative Console as the administrator.
2. In the navigation pane, click **Servers**, expand **Server Type** and select **WebSphere application servers**.
3. Click the name of the application server.
4. Under the **Server Infrastructure** section in the Configuration tab, expand **Java Virtual Machine** and click **Process Definition**.
5. Under the **Additional Properties** section, click **Java Virtual Machine**.
6. In the **Generic JVM arguments** field, remove the following entries from the content.

```
-agentlib:am_ibm_16=${WAS_SERVER_NAME} -Xbootclasspath/p:${ITCAMDCHOME}/
toolkit/lib/bcm-bootstrap.jar -Djava.security.policy=${ITCAMDCHOME}/itcamdc/
etc/datacollector.policy -verbosegc
```

7. Click **Apply** and click **Save**. In the Save to Master Configuration dialog box, complete the following steps:
   - If you are under a Network Deployment environment, ensure that **Synchronize changes with Nodes** is selected and then click **Save**.
   - If you are not under a Network Deployment environment, click **Save**.
8. In the navigation pane, click **Servers**, expand **Server Types**, click **WebSphere application servers** and then click the server name.
9. In the Configuration tab, go to **Server Infrastructure** > **Java and Process Management** > **Process Definition** > **Environment Entries**.
10. Depending on the operating system, the hardware platform, and the application server JVM, remove the following environment entry.
    - UNIX LIBPATH
    - Linux LD_LIBRARY_PATH
    - Windows PATH

11. In the navigation pane, click **Environment** > **WebSphere Variables**.
12. Remove the *ITCAMDCHOME* variable if it exists.
13. Click **Apply** and click **Save**. In the Save to Master Configuration dialog box, complete the following steps:
    - If you are under a Network Deployment environment, ensure that **Synchronize changes with Nodes** is selected and then click **Save**.
    - If you are not under a Network Deployment environment, click **Save**.
14. Restart the application server instance.
15. Go to the `runtime` directory in the agent installation directory and remove the *profile_name.cell_name.node_name.server_name*`.manual.input.properties` file.
    - Linux AIX *install_dir*`/yndchome/7.3.0.14.08/runtime/`*profile_name.cell_name.node_name.server_name*`.manual.input.properties`
    - Windows *install_dir*`\dchome\7.3.0.14.08\runtime\`*profile_name.cell_name.node_name.server_name*`.manual.input.properties`

Manually unconfiguring the data collector for WebSphere Application Server Liberty

Procedure

1. Navigate to the liberty server directory and open the `jvm.options` file in the *server_name* directory within the Liberty server installation directory. For example, `/opt/ibm/wlp/usr/servers/defaultServer`.
2. Remove the following parameters from the `jvm.options` file.

```
-agentlib:am_ibm_16=server_name
–Xbootclasspath/p:dc_home/toolkit/lib/bcm-bootstrap.jar
-Djava.security.policy=dc_home/itcamdc/etc/datacollector.policy
-verbosegc
```

   where, *server_name* is the name of the Liberty server; *dc_home* is the data collector home directory.
3. Open the `server.xml` file and remove the following lines:

```
<feature>webProfile-7.0</feature>
<feature>monitor-1.0</feature>
<feature>usr:itcam-730.140</feature>
```

4. Open the `server.env` file and remove the following entry value from the environment entry per the operating system:

Table 8. Environment entry

Platform	Environment entry name	Environment entry value
AIX R6.1 (64-bit JVM)	LIBPATH	`/lib:`*dc_home*`/toolkit/lib/aix536`
AIX R7.1 (64 bit JVM)	LIBPATH	`/lib:`*dc_home*`/toolkit/lib/aix536`
Solaris 10 (64-bit JVM)	LIBPATH	`/lib:`*dc_home*`/toolkit/lib/sol296`
Solaris 11 (64-bit JVM)	LIBPATH	`/lib:`*dc_home*`/toolkit/lib/sol296`
Linux x86_64 R2.6 (64-bit JVM)	LD_LIBRARY_PATH	`/lib:`*dc_home*`/toolkit/lib/lx8266`

Table 8. Environment entry (continued)

Platform	Environment entry name	Environment entry value
Linux Intel R2.6 (32-bit JVM)	LD_LIBRARY_PATH	`/lib:`*`dc_home`*`/toolkit/lib/li6263`
Windows (32-bit JVM)	PATH	`/lib;`*`dc_home`*`/toolkit/lib/win32`
Windows (64-bit JVM)	PATH	`/lib;`*`dc_home`*`/toolkit/lib/win64`

5. Restart the Liberty server.
6. Go to the `runtime` directory in the WebSphere Applications agent installation directory and remove the *`cell_name.node_name.server_name`*`.manual.input.properties` file.

   - Linux AIX *`install_dir`*`/yndchome/7.3.0.14.08/runtime/`*`cell_name.node_name.server_name`*`.manual.input.properties`
   - Windows *`install_dir`*`\dchome\7.3.0.14.08\runtime\`*`cell_name.node_name.server_name`*`.manual.input.properties`

Node.js agent: Removing the monitoring plug-in

Before you uninstall the Node.js agent, you must remove the monitoring plug-in from your Node.js application.

Procedure

1. Remove data collector plug-ins from the beginning of the Node.js application file.

   - If you upgrade the Node.js agent from V01.00.12.00 to V01.00.13.00, complete the following procedure:

     – If you enabled resource data collection, remove the following line from the beginning of the Node.js application file:

     ```
     require('KNJ_NPM_LIB_LOCATION/node_modules/ibm-apm/knj_index.js');
     ```

     where *KNJ_NPM_LIB_LOCATION* is the directory to the `lib` folder of your npm package global installation directory. The default directory is `/usr/local/lib`.

     – If you enabled resource data collection and deep-dive diagnostics data collection, remove the following line from the beginning of the Node.js application file:

     ```
     require('KNJ_NPM_LIB_LOCATION/node_modules/ibm-apm/knj_deepdive.js');
     ```

     – If you enabled resource data collection, deep-dive diagnostics data collection, and method traces collection, remove the following line from the beginning of the Node.js application file:

     ```
     require('KNJ_NPM_LIB_LOCATION/node_modules/ibm-apm/knj_methodtrace.js');
     ```

   - If you upgrade the Node.js agent from V01.00.10.00 to V01.00.13.00, complete the following procedure:

     – If you enabled resource data collection, remove the following line from the beginning of the Node.js application file.

     ```
     require('install_dir/lx8266/nj/bin/plugin/knj_index.js');
     ```

     , where *install_dir* is the installation directory of Node.js agent.

     – If you enabled resource data collection and deep-dive diagnostics data collection, remove the following line from the beginning of the Node.js application file.

```
require('install_dir/lx8266/nj/bin/plugin/knj_deepdive.js');
```

– If you enabled resource data collection, deep-dive diagnostics data collection, and method traces collection, remove the following line from the beginning of the Node.js application file.

```
require('install_dir/lx8266/nj/bin/plugin/knj_methodtrace.js');
```

2. Restart your Node.js application to disable the data collector plug-ins.
   - If the version of your current Node.js agent is V01.00.10.00, till now the data collector plug-ins are successfully removed.
   - If the version of your current Node.js agent is V01.00.12.00, continue to the following step.
3. Run the `./uninstall.sh` command from the *`install_dir`*`/lx8266/nj/bin` directory to remove your previous agent settings.

What to do next

For more information about uninstalling the Node.js agent, see "Uninstalling your agents" on page 147.

Microsoft .NET agent: Removing the .NET data collector

Before you uninstall the Microsoft .NET agent, you must remove the .NET data collector from your .NET applications.

Procedure

1. Unregister the data collector.

   As an administrator, enter:

   ```
   cd install_dir\qe\bin configdc unregisterdc
   ```

   Where *install_dir* is the installation directory of the Microsoft .NET agent.
2. Stop all of your .NET applications to disable the data collector.

   Enter `net stop was /y`
3. To ensure the complete clean-up of the .NET Data Collector after uninstallation, follow these steps:

   a) On the command prompt, go to the `<APM_HOME>\qe\bin` directory.

   b) Run the `ProcListCaller.bat` file.

   c) Verify the `CorProfAttach.Log` log file at the `<APM_HOME>\qe\logs` directory. The log file lists the processes to which .NET DC profiler component is attached.

   d) Before you uninstall the agent, stop the processes from the `CorProfAttach.Log` file.

   e) If no processes are listed, then continue with the agent uninstallation.

What to do next

Uninstall the Microsoft .NET agent. See "Uninstalling your agents" on page 147.

Chapter 7. Configuring your environment

If your monitoring agent requires configuration or you want to review the default settings for an agent, follow the steps provided for your agent.

Common topics

Some topics are common when you configure agents and data collectors.

Network connectivity

To ensure that agent server communications are established, test your system for connectivity to the Cloud APM server.

To validate communications, test your connectivity to the Cloud APM data center. To ensure that your firewall rules allow return traffic from three specific IP addresses and port 443, find the three data center IP addresses that you need to verify connection as described in Validating connectivity to the data center. Check that your agents can connect to these three IP addresses by using the **openssl** command. For more information about using the **openssl** command, see Configuring agents to communicate through a forward proxy. If your agents cannot connect, contact your local IT team. They can adjust your firewall rules, enable port 443, and enable TLS 1.2 traffic from your servers, or configure a proxy server to connect to the Cloud APM server.

If your firewall rules do not allow transparent outbound HTTPS connections to external hosts, you can configure your agents to send traffic to a forward proxy. For more information, see "Configuring agents to communicate through a forward proxy" on page 159.

Browser connectivity

To check browser connectivity to the Cloud APM console, locate the **Launch** URL, which was provided to you by IBM when your subscription was provisioned. You can also sign in to your account and start the console. Sign in to the **Products and Services** (http://ibm.biz/my-prodsvcs) page with your IBM Marketplace subscription details. Click **Launch** to start the console and view the URL, for example: `8b68ba1b9.agents.na.apm.ibmserviceengage.com`. Verify that you can use the URL to log into the console.

Secure communication

Secure communication between the agents and the Cloud APM server requires TLS 1.2.

Communication between the agents and the Cloud APM server in the IBM Cloud uses HTTPS with TLS 1.2 and the FIPS Suite-B cipher suites. The following ciphers are used:

```
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
```

Communications between the browser and the Cloud APM server also requires TLS 1.2. In some browsers, TLS 1.2 is not enabled by default and must be enabled manually.

Configuring agents to communicate through a forward proxy

If your firewall rules do not allow transparent outbound HTTPS connections to external hosts, you can configure IBM monitoring agents to send traffic to a forward proxy. Edit the `KDH_FORWARDPROXY` environment variable to configure agents to communicate through the forward proxy.

Before you begin

To determine the IP address of the Cloud APM data center that your data collectors connect to, see Validating connectivity to the data center. Then, adjust your firewall rules to allow requests to be sent to those IP addresses from your forward proxy.

You can use the **openssl** command to check whether the computer system where your agents are installed has connectivity to the Cloud APM data center servers. You can also use the **openssl** command to check whether your network supports the cipher suites that are used by Cloud APM. If the **openssl** command results indicate that the computer system cannot connect, you might need to set up a forward proxy. If the command results indicate that the Cloud APM server certificate could not be obtained, then work with your network team to determine why the required cipher suites are not supported. For the list of cipher suites that are used by Cloud APM, see "Secure communication" on page 159.

Run the **openssl** command, as shown in the following example:

```
echo quit | openssl s_client
-state -connect <domain-name>:443
-tls1_2 -cipher
ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384
```

where *domain-name* is the domain name for your Cloud APM subscription, such as `8b68ba1b9.agents.na.apm.ibmserviceengage.com`.

To determine the domain name for your subscription, complete the following steps:

1. Open the agent environment configuration file in a text editor:

   Linux AIX `/opt/ibm/apm/agent/config/global.environment`

   Windows `install_dir\TMAITM6_x64\K`*pc*`ENV` for 64-bit Windows systems and `install_dir\TMAITM6\K`*pc*`ENV` for 32-bit Windows systems, where *pc* is the agent product code.

   For a list of product codes, see "Using agent commands" on page 177.

2. Find the *IRA_ASF_SERVER_URL* variable. The value is in the form: `https://domain-name/ccm/asf/request`. Use the domain name portion of the value with the **openssl** command.

   If the connection is successful, messages similar to the following example are displayed:

```
CONNECTED(00000003)
SSL_connect:before/connect initialization
SSL_connect:SSLv3 write client hello A
SSL_connect:SSLv3 read server hello A
depth=2 C = US, O = IBM Service Engage,
CN = ca_ec_384.ibmserviceengage.com
verify error:num=19:self signed certificate in certificate chain
verify return:0
SSL_connect:SSLv3 read server certificate A
SSL_connect:SSLv3 read server key exchange A
SSL_connect:SSLv3 read server certificate request A
SSL_connect:SSLv3 read server done A
SSL_connect:SSLv3 write client certificate A
SSL_connect:SSLv3 write client key exchange A
SSL_connect:SSLv3 write change cipher spec A
SSL_connect:SSLv3 write finished A
SSL_connect:SSLv3 flush data
SSL_connect:SSLv3 read finished A
---
Certificate chain
0 s:/C=US/O=IBM Service Engage/OU=Application Performance
Management/CN=*.agents.na.apm.ibmserviceengage.com
i:/C=US/O=IBM Service Engage/OU=Application Performance
Management/CN =ca_ec_384.apm.ibmserviceengage.com
1 s:/C=US/O=IBM Service Engage/OU=Application Performance
Management/CN=ca_ec_384.apm.ibmserviceengage.com
i:/C=US/O=IBM Service Engage/CN=ca_ec_384.ibmserviceengage.com
2 s:/C=US/O=IBM Service Engage/CN=ca_ec_384.ibmserviceengage.com
i:/C=US/O=IBM Service Engage/CN=ca_ec_384.ibmserviceengage.com
---
```

```
Server certificate
-----BEGIN CERTIFICATE-----
MIICkjCCAhegAwIBAgIIXlr284nLPaMwDAYIKoZIzj0EAwMFADCBhDELMAkGA1UE
BgwCVVMxGzAZBgNVBAoMEklCTSBTZXJ2aWNlIEVuZ2FnZTErMCkGA1UECwwiQXBw
bGljYXRpb24gUGVyZm9ybWFuY2UgTWFuYWdlbWVudDErMCkGA1UEAwwiY2FfZWNf
Mzg0LmFwbS5pYm1zZXJ2aWNlZW5nYWdlLmNvbTAeFw0xMzEyMDIxNjM2MDlaFw0y
MzEyMDExNjM2MDlaMIGGMQswCQYDVQQGDAJVUzEbMBkGA1UECgwSSUJNIFNlcnZp
Y2UgRW5nYWdlMSswKQYDVQQLDCJBcHBsaWNhdGlvbiBQZXJmb3JtYW5jZSBNYW5h
Z2VtZW50MS0wKwYDVQQDDCQqLmFnZW50cy5uYS5hcG0uaWJtc2VydmljZWVuZ2Fn
ZS5jb20wdjAQBgcqhkjOPQIBBgUrgQQAIgNiAAQmrGoCkAMoNAC3F6MIo1zR8fcO
mczYXtUux2bhlOibn3jQdxamhDR91nr2RBerGjMIITKNXd2MaOr3b6m8euk1BAL3
KsbN9lqvw94kXg0BTO1IHAcdsZQB+AuEVVhmDVGjUDBOMAwGA1UdEwEB/wQCMAAw
HwYDVR0jBBgwFoAU/zpE5TOnQ8LSuvbSWRfpbiGea08wHQYDVR0OBBYEFHL0At4O
GUdcOHVGg4Tfo4hl7LLGMAwGCCqGSM49BAMDBQADZwAwZAIwDWPHo5I04ZFVrkfk
St6gwH2UNF37jBscRN1lOE4SIwezZAqVs42BNMkWRjJBgiHzAjBm4m3zOjsXzNL8
+u8ALjQQCpBDT6dUHujzY5CRxG0xEHi5IXsXf4QwbctnjjvTeYA=
-----END CERTIFICATE-----
subject=/C=US/O=IBM Service Engage/OU=Application Performance
Management/CN=*.agents.na.apm.ibmserviceengage.com
issuer=/C=US/O=IBM Service Engage/OU=Application Performance
Management/CN=ca_ec_384.apm.ibmserviceengage.com
---
Acceptable client certificate CA names
/C=US/O=IBM Service Engage/CN=ca_ec_384.ibmserviceengage.com
/C=US/O=IBM Service Engage/OU=Application Performance
Management/CN=ca_ec_384.apm.ibmserviceengage.com
/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert
Global Root CA/C=US/O=IBM Service Engage/OU=Application Performance
Management/CN=*.agents.na.apm.ibmserviceengage.com
Server Temp Key: ECDH, prime256v1, 256 bits
---
SSL handshake has read 2659 bytes and written 261 bytes
---
New, TLSv1/SSLv3, Cipher is ECDHE-ECDSA-AES128-GCM-SHA256
Server public key is 384 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
SSL-Session:
Protocol : TLSv1.2
Cipher : ECDHE-ECDSA-AES128-GCM-SHA256
Session-ID:
A18C31D0B45A1166357C917E1CFCD86A9FBEDB4A0EB768EF5390AC28C95CB7EF
Session-ID-ctx:
Master-Key:
252B8FE2731E51AC0B79A27C7BED33CA8B15AF4CFD015C98DBACA46EA01DC40B
9E6B56E62E0F332FF6B56266B5ADD7B0
Key-Arg : None
Krb5 Principal: None
PSK identity: None
PSK identity hint: None
Start Time: 1510772474
Timeout : 7200 (sec)
Verify return code: 19 (self signed certificate in certificate chain)
---
DONE
SSL3 alert write:warning:close notify
```

If the computer system does not have connectivity to the Cloud APM server, messages similar to the following example are displayed:

```
getaddrinfo: Name or service not known
connect:errno=2
```

If the computer system cannot obtain the server certificate, because the cipher suites are being blocked somewhere in the network, messages similar to the following example are displayed:

```
SSL_connect:failed
---
no peer certificate available
---
No client certificate CA names sent
```

About this task

When a forward proxy is used, the agent first opens a TCP connection with the proxy. The agent sends an HTTP CONNECT request and the target endpoint (Cloud APM server) URL to the forward proxy. Then, the forward proxy establishes a TCP connection with the target endpoint and sets up an HTTPS tunneling session between the agent and the Cloud APM server.

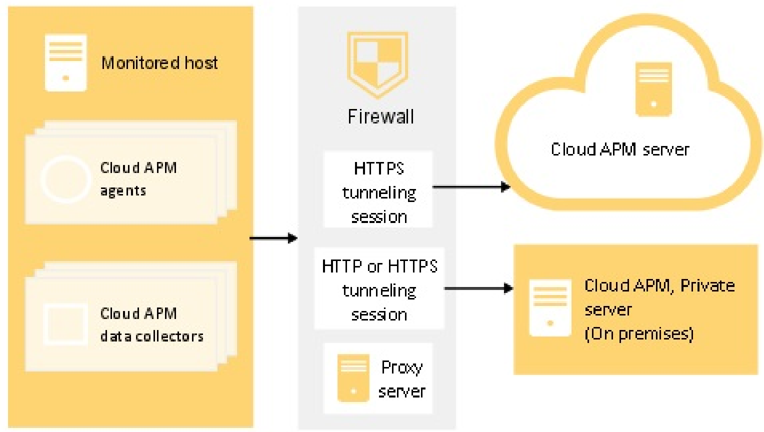

Monitored host
Cloud APM agents
Cloud APM data collectors
Firewall
HTTPS tunneling session
HTTP or HTTPS tunneling session
Proxy server
Cloud APM server
Cloud APM, Private server (On premises)

Figure 1. Connection diagram for using a forward proxy

The monitoring agent does not support authenticating proxies, which means the agent does not support logging on to a forward proxy by using a configured proxy user ID and password.

Procedure

Complete these steps to configure agents to communicate through a forward proxy.

1. Open the agent environment configuration file in a text editor:

   Linux AIX *install_dir*/config/global.environment file, where *install_dir* is the installation home directory of the agents. The global.environment file configures all agents in the installation directory.

   The customized settings in the .global.environment file are lost after agent upgrade. To preserve your settings, make customization changes in the global.environment files. The settings in this file are not overwritten by agent upgrade.

   Windows *install_dir*\TMAITM6_x64\K*pc*ENV file for 64-bit agents, and *install_dir*\TMAITM6\K*pc*ENV for 32-bit agents, where *pc* is the agent product code. Configure the K*pc*ENV file for each agent.

   For a list of product codes, see "Using agent commands" on page 177.
2. Edit the KDH_FORWARDPROXY environment variable to specify the proxy address and port:

```
KDH_FORWARDPROXY=http://proxy-address:proxy-port-number
```

For example:

```
KDH_FORWARDPROXY=http://HostA:8085
```

3. Restart the agent to implement your changes. See “Using agent commands” on page 177.

Configuring data collectors to communicate through a forward proxy

If your firewall rules do not allow transparent outbound HTTPS connections to external hosts, you can configure data collectors to send traffic to a forward proxy. Edit the APM_GW_PROXY_CONNECTION environment variable to configure data collectors to communicate through the forward proxy.

Before you begin

To determine the IP address of the Cloud APM data center that your data collectors connect to, see Validating connectivity to the data center. Then, adjust your firewall rules to allow requests to be sent to those IP addresses from your forward proxy.

You can use the **openssl** command to check whether the computer system where your data collectors are installed has connectivity to the Cloud APM data center servers. You can also check whether your network supports the cipher suites that are used by Cloud APM. If the **openssl** command results indicate that the computer system cannot connect, you might need to set up a forward proxy. If the command results indicate that the Cloud APM server certificate could not be obtained, then work with your network team to determine why the required cipher suites are not supported. For the list of cipher suites that are used by Cloud APM, see “Secure communication” on page 159.

Run the **openssl**, as shown in the following example:

```
echo quit | openssl s_client
-state -connect <domain-name>:443
-tls1_2 -cipher
ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384
```

where, *domain-name* is the domain name for your Cloud APM subscription.

To determine the domain name for your subscription, see “Configuring agents to communicate through a forward proxy” on page 159.

If the connection is successful, messages similar to the following example are displayed:

```
CONNECTED(00000003)
SSL_connect:before/connect initialization
SSL_connect:SSLv3 write client hello A
SSL_connect:SSLv3 read server hello A
depth=2 C = US, O = IBM Service Engage,
CN = ca_ec_384.ibmserviceengage.com
verify error:num=19:self signed certificate in certificate chain
verify return:0
SSL_connect:SSLv3 read server certificate A
SSL_connect:SSLv3 read server key exchange A
SSL_connect:SSLv3 read server certificate request A
SSL_connect:SSLv3 read server done A
SSL_connect:SSLv3 write client certificate A
SSL_connect:SSLv3 write client key exchange A
SSL_connect:SSLv3 write change cipher spec A
SSL_connect:SSLv3 write finished A
SSL_connect:SSLv3 flush data
SSL_connect:SSLv3 read finished A
---
Certificate chain
0 s:/C=US/O=IBM Service Engage/OU=Application Performance
Management/CN=*.agents.na.apm.ibmserviceengage.com
```

```
i:/C=US/O=IBM Service Engage/OU=Application Performance
Management/CN =ca_ec_384.apm.ibmserviceengage.com
1 s:/C=US/O=IBM Service Engage/OU=Application Performance
Management/CN=ca_ec_384.apm.ibmserviceengage.com
i:/C=US/O=IBM Service Engage/CN=ca_ec_384.ibmserviceengage.com
2 s:/C=US/O=IBM Service Engage/CN=ca_ec_384.ibmserviceengage.com
i:/C=US/O=IBM Service Engage/CN=ca_ec_384.ibmserviceengage.com
---
Server certificate
-----BEGIN CERTIFICATE-----
MIICkjCCAhegAwIBAgIIXlr284nLPaMwDAYIKoZIzj0EAwMFADCBhDELMAkGA1UE
BgwCVVMxGzAZBgNVBAoMEklCTSBTZXJ2aWNlIEVuZ2FnZTErMCkGA1UECwwiQXBw
bGljYXRpb24gUGVyZm9ybWFuY2UgTWFuYWdlbWVudDErMCkGA1UEAwwiY2FfZWNf
Mzg0LmFwbS5pYm1zZXJ2aWNlZW5nYWdlLmNvbTAeFw0xMzEyMDIxNjM2MDlaFw0y
MzEyMDExNjM2MDlaMIGGMQswCQYDVQQGDAJVUzEbMBkGA1UECgwSSUJNIFNlcnZp
Y2UgRW5nYWdlMSswKQYDVQQLDCJBcHBsaWNhdGlvbiBQZXJmb3JtYW5jZSBNYW5h
Z2VtZW50MS0wKwYDVQQDDCQqLmFnZW50cy5uYS5hcG0uaWJtc2VydmljZWVuZ2Fn
ZS5jb20wdjAQBgcqhkjOPQIBBgUrgQQAIgNiAAQmrGoCkAMoNAC3F6MIo1zR8fcO
mczYXtUux2bhlOibn3jQdxamhDR91nr2RBerGjMIITKNXd2MaOr3b6m8euk1BAL3
KsbN9lqvw94kXg0BTO1IHAcdsZQB+AuEVVhmDVGjUDBOMAwGA1UdEwEB/wQCMAAw
HwYDVR0jBBgwFoAU/zpE5TOnQ8LSuvbSWRfpbiGea08wHQYDVR0OBBYEFHL0At4O
GUdcOHVGg4Tfo4hl7LLGMAwGCCqGSM49BAMDBQADZwAwZAIwDWPHo5I04ZFVrkfk
St6gwH2UNF37jBscRN1lOE4SIwezZAqVs42BNMkWRjJBgiHzAjBm4m3zOjsXzNL8
+u8ALjQQCpBDT6dUHujzY5CRxG0xEHi5IXsXf4QwbctnjjvTeYA=
-----END CERTIFICATE-----
subject=/C=US/O=IBM Service Engage/OU=Application Performance
Management/CN=*.agents.na.apm.ibmserviceengage.com
issuer=/C=US/O=IBM Service Engage/OU=Application Performance
Management/CN=ca_ec_384.apm.ibmserviceengage.com
---
Acceptable client certificate CA names
/C=US/O=IBM Service Engage/CN=ca_ec_384.ibmserviceengage.com
/C=US/O=IBM Service Engage/OU=Application Performance
Management/CN=ca_ec_384.apm.ibmserviceengage.com
/C=US/O=DigiCert Inc/OU=www.digicert.com/CN=DigiCert
Global Root CA/C=US/O=IBM Service Engage/OU=Application Performance
Management/CN=*.agents.na.apm.ibmserviceengage.com
Server Temp Key: ECDH, prime256v1, 256 bits
---
SSL handshake has read 2659 bytes and written 261 bytes
---
New, TLSv1/SSLv3, Cipher is ECDHE-ECDSA-AES128-GCM-SHA256
Server public key is 384 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
SSL-Session:
Protocol : TLSv1.2
Cipher : ECDHE-ECDSA-AES128-GCM-SHA256
Session-ID:
A18C31D0B45A1166357C917E1CFCD86A9FBEDB4A0EB768EF5390AC28C95CB7EF
Session-ID-ctx:
Master-Key:
252B8FE2731E51AC0B79A27C7BED33CA8B15AF4CFD015C98DBACA46EA01DC40B
9E6B56E62E0F332FF6B56266B5ADD7B0
Key-Arg : None
Krb5 Principal: None
PSK identity: None
PSK identity hint: None
Start Time: 1510772474
Timeout : 7200 (sec)
Verify return code: 19 (self signed certificate in certificate chain)
---
DONE
SSL3 alert write:warning:close notify
```

If the computer system does not have connectivity to the Cloud APM server, messages similar to the following example are displayed:

```
getaddrinfo: Name or service not known
connect:errno=2
```

If the computer system cannot obtain the server certificate, because the cipher suites are being blocked somewhere in the network, messages similar to the following example are displayed:

```
SSL_connect:failed
---
no peer certificate available
---
No client certificate CA names sent
```

About this task

When a forward proxy is used, the data collector first opens a TCP connection with the proxy. The data collector sends a connection request and the target endpoint (Cloud APM server) URL to the forward proxy. Then, the forward proxy establishes a TCP connection with the target endpoint and sets up an HTTPS tunneling session between the data collector and the Cloud APM server.

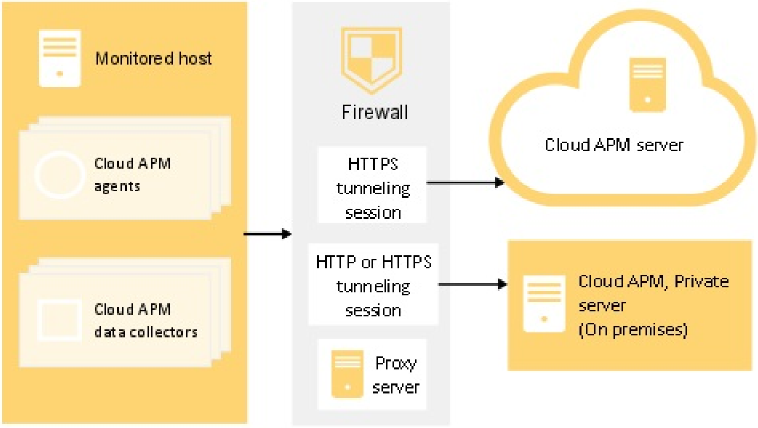

Monitored host
Cloud APM agents
Cloud APM data collectors
Firewall
HTTPS tunneling session
HTTP or HTTPS tunneling session
Proxy server
Cloud APM server
Cloud APM, Private server (On premises)

Figure 2. Connection diagram for using a forward proxy

Some data collectors support authenticating proxies, for example Node.js and Liberty data collectors. These data collectors support logging on to a forward proxy by using a configured proxy user ID and password.

Procedure

1. To configure forward proxy communication for Python data collectors, complete one of the following steps:
   - Open the *<dc home>*/config.properties data collector properties file in a text editor, where *<dc home>* is the installation home directory of the data collectors, for example, /usr/lib/python2.7/site-packages/ibm_python_dc. Update the variable with the proxy host and port number, for example, APM_GW_PROXY_CONNECTION =http://9.181.138.247:8085. Editing the variable in this file impacts all applications with the Python data collector enabled.

    Note: To configure forward proxy communication for a single application, copy the*<dc home>*/`config.properties` file to the directory of the single application. Update the variable in the application directory.

  - Run the following command on Linux systems:

    ```
    export APM_GW_PROXY_CONNECTION =http://<http proxy host>:<http proxy port>
    ```

    for example,

    ```
    export APM_GW_PROXY_CONNECTION =http://9.181.138.247:8085
    ```

2. To configure forward proxy communication for Node.js data collectors, complete one of the following steps:

   - Run the following command on Linux systems:

     ```
     export APM_GW_PROXY_CONNECTION =http://<http proxy host>:<http proxy port>
     ```

     for example,

     ```
     export APM_GW_PROXY_CONNECTION =http://9.181.138.247:8085
     ```

   - If a user name and password is required to access the forward proxy server for Node.js data collectors, run the following command on Linux systems:

     ```
     export APM_GW_PROXY_CONNECTION =http://<http proxy user>:
     <http proxy password>@<http proxy host>:<http proxy port>
     ```

     for example,

     ```
     export APM_GW_PROXY_CONNECTION =http://Joe:passw0rd@9.181.138.247:8085
     ```

3. To configure forward proxy communication for Liberty data collectors, edit the *<Liberty server home>*/`jvm.options` file, where *<Liberty server home>* is the Liberty server home directory, for example: `/opt/ibm/wlp/usr/servers/defaultServer/jvm.options`. Complete one of the following steps:

   - If authentication is not required, add the following code to the `jvm.options` file:

     ```
     -Dhttp.proxyHost=<http proxy host>
     -Dhttp.proxyPort=<http proxy port>
     -Dhttps.proxyHost=<https proxy host>
     -Dhttps.proxyPort=<https proxy port>
     -Djava.net.useSystemProxies=true
     ```

   - If a user name and password is required to access the forward proxy server, add the following code to the `jvm.options` file:

     ```
     -Dhttp.proxyHost=<http proxy host>
     -Dhttp.proxyPort=<http proxy port>
     -Dhttp.proxyUser=<http proxy user>
     -Dhttp.proxyPassword=<http proxy password>
     -Dhttps.proxyHost=<https proxy host>
     -Dhttps.proxyPort=<https proxy port>
     -Dhttps.proxyUser=<https proxy user>
     -Dhttps.proxyPassword=<https proxy password>
     -Djava.net.useSystemProxies=true
     ```

4. Restart the local application to implement your changes.

Results

You configured your data collectors to communicate through a forward proxy.

Validating connectivity to the data center

Successful configuration of your Cloud APM environment includes confirming that your monitoring agents and console users can connect the Cloud APM server.

Procedure

Determine the IP address of the Cloud APM data center where your subscription is based:

1. Find the geography location code embedded inside the APM subscription URL where *<geo>* is the code:

   and console users

   ```
   https://.customers.<geo>.apm.ibmserviceengage.com
   ```

   Example: *`https://1234567890abcdef1234567890abcdef.customers.na.apm.ibmserviceengage.com`*

Geography location code	Data center	Region
na	Washington, D.C.	North America
eu	Amsterdam	Europe
ap	Sydney	Asia Pacific
in-ap	Chennai	Asia Pacific

2. Test connectivity to ensure that your firewall rules allow return traffic from all three IP addresses for your data center through port 443.

   a) To test connectivity for downloading agent packages, point your browser to the IP address for your location:

na	`https://169.47.30.166`
eu	`https://159.8.20.242`
ap	`https://130.198.88.182`
in-ap	`https://169.38.69.54`

   If you have connectivity, you are prompted to validate the certificate. After you accept the certificate, the ibm.com page is displayed.

   b) To test connectivity for agent communication, point your browser to the IP address for your location:

na	`https://169.47.30.165`
eu	`https://159.8.20.241`
ap	`https://130.198.88.181`
in-ap	`https://169.38.69.53`

   If you have connectivity, you are prompted to validate the certificate. After you accept the certificate, a forbidden error is displayed.

   c) To test connectivity for Cloud APM console communication, point your browser to the IP address for your location:

na	`https://169.47.30.164`
eu	`https://159.8.20.240`

ap	`https://130.198.88.180`
in-ap	`https://169.38.69.52`

If you have connectivity, you are prompted to validate the certificate.

Results

After you accept the certificate, the IBM HTTP Server welcome page is displayed.

Managed system names

The managed system name (MSN) is used to uniquely identify each Cloud APM agent within your environment. It is also the instance name that you see on the Application Performance Dashboard when you select a group for each managed system from the navigator **Groups** section. To avoid conflicts in your environment, assign unique MSNs to your agents.

The agent MSN format differs, depending on your agent type. It falls into one of the following categories:

- "Common MSN format for single-instance agents" on page 168
- "Common MSN format for multi-instance agents" on page 168
- "Special MSN format" on page 169

Common MSN format for single-instance agents

For most single-instance agents, the common form of the MSN follows this format:

```
hostname:pc
```

where:

- *hostname* is the name of the computer where the agent is installed. This part can be changed if needed.
- *pc* is the uppercase two-character agent code, which cannot be changed. For more information about agent codes, see "Using agent commands" on page 177.
- `:` is the separator, which cannot be changed.

Example: `linuxhost01:LZ` is the MSN of the Linux OS agent.

Some single-instance agents that do not follow this MSN format are listed in Table 9 on page 169.

The MSN is limited to 32 characters. For this MSN category, 29 characters are available for the host name because the agent code and separator cannot be changed.

Important: If the length of the MSN exceeds 32 characters, part of the MSN is truncated and it does not display correctly in the Cloud APM console. For example, if your host name is `VeryLongSalesDivisionServerName03`, your managed system name should be `VeryLongSalesDivisionServerName03:`*PC*. However, it is truncated to `VeryLongSalesDivisionServerName0`.

Common MSN format for multi-instance agents

For most multi-instance agents, the common form of the MSN follows this format:

```
instancename:hostname:pc
```

where:

- *instancename* is the agent instance name that you specify during agent configuration. Use this variable to ensure a unique MSN for each instance of each agent type on each agent host computer.

  Remember:

  - Letters from the Latin alphabet (a-z, A-Z), Arabic numerals (0-9), and the hyphen-minus character (-) can be used to create agent instance names.

  - The underscore character (_) is not allowed in agent instance names.
  - The instance name that you specify is limited as follows:
    - Linux AIX 28 characters minus the length of your host name on Linux or AIX systems.
    - Windows 28 characters minus the length of your host name when using the silent response file for configuration on Windows systems. Example, `Server-Name` is 11 characters long. So agent instances on the `Server-Name` host must be less than or equal to 17 characters in length.
    - Windows 20 characters minus the length by which your host name exceeds 8 characters when using the Cloud APM console configuration on Windows systems. Example, `TestServer` is 10 characters long, which exceeds 8 by 2. So agent instances on the `TestServer` host must be less than or equal to 18 characters in length.
- *hostname* is the name of the computer where the agent is installed. The host name component of the MSN can be changed if necessary.
- *pc* is the uppercase two-character agent code, which cannot be changed. For more information about agent codes, see "Using agent commands" on page 177.
- `:` is the separator, which cannot be changed.

Example: `jboss1:win2016:JE` is the MSN for the JBoss agent.

Some multi-instance agents that do not follow this MSN format are listed in Table 9 on page 169.

The MSN is limited to 32 characters. For this MSN category, 28 characters are available between the instance name and the host name because the agent code and separators cannot be changed.

Important: If the length of the MSN exceeds 32 characters, part of the MSN is truncated and it does not display correctly in the Cloud APM console. For example, if you specify `VeryLongInstanceName` as your instance name, and your server name is `Production09`, your managed system name should be `VeryLongInstanceName:Production09:`*PC*. However, it is truncated to `VeryLongInstanceName:Production0`.

Special MSN format

Special MSN format applies to the agents whose MSNs do not follow the above two common MSN formats. These agents are listed in Table 9 on page 169.

The special MSN is limited to 32 characters. In Table 9 on page 169, only the italic strings in the MSN format column can be changed.

Table 9. Special MSN format

Agents	MSN format	MSN example
Amazon EC2 agent	B5:*ec2subnodename*:INS	B5:sales:INS
Amazon ELB agent	• AL:*instancename*A:APP • AL:*instancename*C:CLA • AL:*instancename*N:NET	• AL:elb-inst3A:APP • AL:elb-inst3C:CLA • AL:elb-inst3N:NET
Azure Compute agent	AK:*azure_compute_subnode_name*:AVM	AK:azc-inst3:AVM
Citrix VDI agent	VD:*citrixsitename*:XDS	VD:xds1:XDS
DataPower agent	BN:*datapowersystemname*:DPS	BN:datapower23:DPS
HTTP Server agent	HU:*hostname_alias*:HUS	HU:docker-ihs_httpd:HUS
IBM Integration Bus agent	*monitoredbrokername*:*agentID*:KQIB	TRADEBRK:AGT1:KQIB

Table 9. Special MSN format (continued)

Agents	**MSN format**	**MSN example**
MQ Appliance agent	MK:*hostname_sectionname*:ARM	MK:bvtmin_linux150:ARM
Node.js agent	NJ:*hostname_port*:NJA	NJ:KVM-014179_3000:NJA
Oracle Database agent	• RZ:*dbconnection-instancename-hostname*:ASM • RZ:*dbconnection-instancename-hostname*:DG • RZ:*dbconnection-instancename-hostname*:RDB	RZ:11g-oracledbdemo-GVT-1BL:RDB
Ruby agent	KM:*hostname_appname*:RAP	KM:nc9098036112_Blog:RAP
SAP agent	• SAP Instance: *instancename-hostname_sid_instancenumber*:Ins • SAP Process Integration: *instancename-hostname*:PI • SAP Solution Manager: *instancename-hostname*:Slm • SAP System: *instancename-hostname*:Sys	• PS5-IBMSAP3V1_PS5_11:Ins • PS5-IBMSAP3V1:PI • PS8-IBMSAP3V3:Slm • PS5-IBMSAP3V1:Sys
SAP HANA Database agent	• SAP Hana Database: S7:*dbname-systemsid*:HDB • SAP Hana System: *instancename*:*hostname*:S7	• S7:HNA-HNA:HDB • HNA:PS8760:S7
SAP NetWeaver Java Stack agent	• SAP NW Java AS Cluster: *instancename*:*hostname*:SV • SAP NW Java AS Instance: SV:*systemsid-jvmid*:NWJ	• J01:VPT02F17:SV • SV:J01-83309750:NWJ
UNIX OS agent	*hostname*:KUX	worklight17:KUX
WebLogic agent	WB:*instancename*:WLS	WB:Server1:WLS
WebSphere Applications agent	• WebSphere Application Server: *serveralias*:*hostname*:KYNS • WebSphere Portal Server: *serveralias*:*hostname*:KYNR • WebSphere Process Server: *serveralias*:*hostname*:KYNP	simpletrade:worklight17:KYNS
WebSphere MQ agent	*monitoredqmgrname*:*agentname*:MQ	TRADEQM:PoC:MQ
Windows OS agent	Primary:*hostname*:NT	Primary:TRADEIIS1:NT

Changing the agent managed system name

Different procedures apply to change the managed system name for different Cloud APM agents. For some agents, changing the managed system name means changing the host name or agent instance name (or both) in the managed system name. For other agents, specific procedures are required to change the managed system name.

Before you begin

Get familiar with the managed system name formats and naming restrictions as described in "Managed system names" on page 168.

About this task

For most Cloud APM agents, you can use the **CTIRA_HOSTNAME** parameter to change the host name used in the managed system name. To change the agent instance name in the managed system name for multi-instance agents, you can use the agent configuration parameter. If you have configured the agent, you must reconfigure it to assign a different agent instance name. After you reconfigure the agent, you will not be able to retrieve the data collected by the previous agent instance.

You might not be able to change the managed system name in one single procedure, depending on which part of the managed system name you want to change.

To find out the managed system name change method for the agent of your interest, refer to Table 10 on page 171.

Exception: Changing the managed system name is not supported by the HTTP Server agent, Node.js agent, or the Synthetic Playback agent

Table 10. Changing managed system name methods for Cloud APM agents

Agent	managed system name change method
Amazon EC2 agent	Use agent configuration parameter to change the EC2 subnode name in the managed system name, see "Configuration parameters for the Amazon EC2 agent" on page 195.
Amazon ELB agent	Create a new agent instance with a new instance name to change the managed system name.
Azure Compute agent	Use agent configuration parameter to change the subnode name in the managed system name, see "Configuration parameters for the Azure Compute agent" on page 209.
Cassandra agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
Cisco UCS agent	Use agent configuration parameter to change the agent instance name, see "Configuration parameters for the agent" on page 217.
Citrix VDI agent	Use agent configuration parameter to change the Citrix site name, see "Configuration parameters for the Citrix VDI agent" on page 227.
Db2 agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.

Table 10. Changing managed system name methods for Cloud APM agents (continued)

Agent	managed system name change method
DataPower agent	Use agent configuration parameter to change the managed system name, see "Configuring the DataPower agent" on page 238.
DataStage agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
Hadoop agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175.
HMC Base agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
IBM Cloud agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
IBM Integration Bus agent	"Specifying unique managed system name for IBM Integration Bus agent" on page 285
JBoss agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
Linux KVM agent	Use agent configuration parameters, see "Configuring Linux KVM monitoring" on page 474.
Linux OS agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175.
Microsoft .NET agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175.
Microsoft Active Directory agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175.
Microsoft Exchange Server agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175.

Table 10. Changing managed system name methods for Cloud APM agents (continued)

Agent	managed system name change method
Microsoft Hyper-V Server agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175.
Microsoft IIS agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175.
Microsoft Office 365 agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175.
Microsoft SQL Server agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
Microsoft SharePoint Server agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175.
MongoDB agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
MySQL agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
NetApp Storage agent	Use agent configuration parameters, see "Configuring NetApp Storage monitoring" on page 577.
OpenStack agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
Oracle Database agent	Use agent configuration parameters, see "Configuring Oracle Database monitoring" on page 610.
PHP agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.

Table 10. Changing managed system name methods for Cloud APM agents (continued)

Agent	**managed system name change method**
PostgreSQL agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
RabbitMQ agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
Response Time Monitoring agent	"Specifying unique managed system name for the Response Time Monitoring agent" on page 707
Ruby agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175.
SAP agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
SAP HANA Database agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
SAP NetWeaver Java Stack agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
Siebel agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
Skype for Business Server agent (formerly known as Microsoft Lync Server agent)	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175.

Table 10. Changing managed system name methods for Cloud APM agents (continued)

Agent	**managed system name change method**
Sterling File Gateway agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
Sterling Connect Direct agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
Tomcat agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
UNIX OS agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175.
WebSphere Applications agent	To change the host name in the managed system name, see How to change the host name used in managed system name for the WAS agent instance?. To change the server alias name in the managed system name, reconfigure the agent, see "Reconfiguring the data collector interactively" on page 839.
WebSphere Infrastructure Manager agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175. Use agent configuration parameter to change the instance name in the managed system name.
WebSphere MQ agent	"Specifying unique managed system names for multiple queue managers" on page 918
Windows OS agent	Use **CTIRA_HOSTNAME** parameter to change the host name in the managed system name. See "Changing the host name in managed system name" on page 175.

Changing the host name in managed system name

About this task

It is not a common practice to change the host name in the managed system name. The host name is automatically detected and set during agent configuration. Change the host name in the managed system name only when necessary and make sure the value that you specify does not cause any truncations due to managed system name naming restrictions.

Procedure

1. Stop all existing instances of the agent and wait for the Cloud APM console to show that the agent or its subnodes are offline. If you do not have any existing agent instances, proceed to the next step.

   For more information about stopping agent instances, see “Using agent commands” on page 177.
2. If the agent is a single-instance agent, complete the following steps to change the **CTIRA_HOSTNAME** parameter. The value that you specify for the **CTIRA_HOSTNAME** parameter is the value that is applied to all new agent instances.
   a) Make a backup copy of the following file:
      - Linux AIX *install_dir*/config/*pc*.environment
      - Windows *install_dir*/TMAITM6_x64/k*pc*cma.ini

      where:
      - *install_dir* is the agent installation directory.
      - *pc* is the two character agent code. See Agent names and agent codes table.

   b) Edit the file by changing the **CTIRA_HOSTNAME** parameter value as follows, where *newhostname* is the custom string that is used instead of the actual host name of the computer where the agent is installed.
      - Linux AIX **CTIRA_HOSTNAME**=*newhostname*
      - Windows **CTIRA_HOSTNAME**=*newhostname* .TYPE=REG_EXPAND_SZ

   c) Save your changes.
3. If the agent is a multi-instance agent, complete the following steps to change the **CTIRA_HOSTNAME** parameter. Normally all agent instances on a computer use the same host name value. If you need agent instances to use differing values, vary the value that you assign to **CTIRA_HOSTNAME** when you perform this step.
   a) Make a backup copy of the following files:
      - Linux AIX *install_dir*/config/*pc*_*instance*.environment
      - Windows *install_dir*/TMAITM6_x64/k*pc*cma_*instance*.ini

   b) Edit the file the change the **CTIRA_HOSTNAME** parameter value as follows:
      - Linux AIX **CTIRA_HOSTNAME**=*newhostname*
      - Windows **CTIRA_HOSTNAME**=*newhostname* .TYPE=REG_EXPAND_SZ

   c) Save your changes.
4. Windows

   Reconfigure existing agent instances. You can either reconfigure the agent using the procedure in Using the IBM Cloud Application Performance Management window on Windows systems or use the agent .bat script to configure the agent. See “Using agent commands” on page 177 for more details on using the agent .bat script. If the agent .bat script does not provide a config option, use the IBM Cloud Application Performance Management GUI to reconfigure the agent.
5. Start all agent instances.

What to do next

After you change the agent managed system name, start the Cloud APM console and modify your applications by removing the old managed system name from applications and adding the new managed system name in its place.

Configuring agents

After installation, some agents are configured and started automatically, while some agents require manual configuration but start automatically. Some agents must be configured and started manually. Multiple instance agents require creating a first instance and starting manually.

Before you begin

When you install an agent, a sample silent configuration file is placed in the `/opt/ibm/apm/agent/samples` directory, for example, `ynv_silent_config_agent.txt` and `datapower_silent_config.txt`.

Note: Some agents, for example Monitoring Agent for WebSphere Applications, have multiple silent configuration files for different tasks such as configuring the data collector.

About this task

For specific deployment details for agents, see Chapter 5, "Agent and data collector deployment," on page 113.

To configure an agent, you can use the command line or a silent response file as described in this procedure.

Configuration methods vary across agents; use the procedure that is provided for your agent.

Procedure

- Run the `*agent-name*.sh config` command.

  For more commands, see Table 12 on page 180 and Table 13 on page 181.
- Edit the silent response file and then run one of the following commands:
  - For single instance agents, run the following command:

    ```
    agent-name.sh config response_file
    ```
  - For multiple-instance agents, run the following command:

    ```
    agent-name.sh config instance_name response_file
    ```
  - where *instance_name* is the instance name, which can be assigned to indicate what you are monitoring.
- Windows

  For agents that are supported on Windows systems, you can perform certain configuration tasks by using the IBM Cloud Application Performance Management window. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Cloud Application Performance Management**. For more information, see "Using the IBM Cloud Application Performance Management window on Windows systems" on page 182.
- To perform advanced configuration for certain agents such as configuring transaction tracking or data collection, and enabling diagnostic data, use the **Agent Configuration** window. For more information, see "Agent Configuration page" on page 182.

Using agent commands

The same scripts that you use to install monitoring agents are also used to check the status of an installed agent, stop or start it, or uninstall the agent.

About this task

The agent name and agent codes are provided for your reference.

Use the agent name in the following commands:

Linux AIX *name*-`agent.sh`

Windows *name*-agent.bat

Where *name* is the name of the agent that is specified in Table 11 on page 178.

Table 11. Agent names and agent codes

Monitoring agent	***name***	**Two letter agent code**
Monitoring Agent for Amazon EC2	amazon_ec2	b5
Monitoring Agent for Azure Compute	azure_compute	ak
Monitoring Agent for Cassandra	cassandra	zc
Monitoring Agent for Cisco UCS	cisco_ucs	v6
Monitoring Agent for Citrix Virtual Desktop Infrastructure	citrix_vdi	vd
Monitoring Agent for DataPower	datapower	bn
Monitoring Agent for Db2	db2	ud
Monitoring Agent for Hadoop	hadoop	h8
Monitoring Agent for HMC Base	hmc_base	ph
Monitoring Agent for HTTP Server	http_server	hu
Monitoring Agent for IBM Cloud	ibm_cloud	fs
Monitoring Agent for IBM Integration Bus	iib	qi
Monitoring Agent for MQ Appliance	ibm_mq_appliances	mk
Monitoring Agent for InfoSphere DataStage	datastage	dt
Monitoring Agent for JBoss	jboss	je
Monitoring Agent for Linux KVM	linux_kvm	v1
Monitoring Agent for Linux OS	os	lz
Monitoring Agent for MariaDB	mariadb	mj
Monitoring Agent for Microsoft Active Directory	msad	3z
Monitoring Agent for Microsoft Cluster Server	mscs	q5
Monitoring Agent for Microsoft Exchange Server	msexch	ex
Monitoring Agent for Microsoft Hyper-V Server	microsoft_hyper-v_server	hv
Monitoring Agent for Microsoft Internet Information Services	msiis	q7
Monitoring Agent for Skype for Business Server (formerly known as Microsoft Lync Server)	skype_for_business_server	ql
Monitoring Agent for Microsoft .NET	dotnet	qe
Monitoring Agent for Microsoft Office 365	microsoft_office365	mo
Monitoring Agent for Microsoft SharePoint Server	ms_sharepoint_server	qp
Monitoring Agent for Microsoft SQL Server	mssql	oq
Monitoring Agent for MongoDB	mongodb	kj

Table 11. Agent names and agent codes (continued)

Monitoring agent	***name***	**Two letter agent code**
Monitoring Agent for MySQL	mysql	se
Monitoring Agent for NetApp Storage	netapp_storage	nu
Monitoring Agent for Node.js	nodejs	nj
Monitoring Agent for OpenStack	openstack	sg
Monitoring Agent for Oracle Database	oracle_database	rz
Monitoring Agent for PHP	php	pj
Monitoring Agent for PostgreSQL	postgresql	pn
Monitoring Agent for Python	python	pg
Monitoring Agent for RabbitMQ	rabbitMQ	zr
Monitoring Agent for Ruby	ruby	km
Monitoring Agent for SAP Applications	sap	sa
Monitoring Agent for SAP HANA Database	sap_hana_database	s7
Monitoring Agent for SAP NetWeaver Java Stack	sap_netweaver_java_stack	sv
Monitoring Agent for Siebel	siebel	uy
Monitoring Agent for Sterling Connect Direct	sterling_connect_direct-agent	FC
Monitoring Agent for Sterling File Gateway	file_gateway	fg
Monitoring Agent for Sybase Server	sybase	oy
Monitoring Agent for Synthetic Playback	synthetic_transactions	sn
Monitoring Agent for Tomcat	tomcat	ot
Monitoring Agent for UNIX OS	os	ux
Monitoring Agent for VMware VI	vmware_vi	vm
Monitoring Agent for WebLogic	oracle_weblogic	wb
Monitoring Agent for WebSphere Applications	was	yn
Monitoring Agent for WebSphere Infrastructure Manager	wim	d0
Monitoring Agent for WebSphere MQ	mq	mq
Monitoring Agent for Windows OS	os	nt
Response Time Monitoring Agent	rt	t5

Procedure

- Linux AIX

  On the system where you want to send a command to the monitoring agent, change to the `install_dir/bin` directory. Enter any of the commands in Table 12 on page 180 where *name* is the agent name that is specified in Table 11 on page 178.

Table 12. Commands for UNIX and Linux systems

Command	**Description**
./*name*-agent.sh status	Checks the agent status. Status can be either running or not running. When the agent is running, the connection status between the agent and the Cloud APM server is also checked. Possible negative connection statuses are: Connection failed, Error detected, Disconnected-error. The positive status is Connected, this is the expected status. The transitional status is Connecting. A status of Unknown means that the agent status cannot be recognized, possible due to errors in the file system or in the agent log file.
./*name*-agent.sh start	Starts the monitoring agent. If the agent has instances, enter an instance name after the command.
./*name*-agent.sh stop	Stops the agent. If the agent has instances, enter an instance name after the command.
./*name*-agent.sh prereqcheck	Runs a prerequisite scan. This command option is available for most agents.
./*name*-agent.sh install	Installs the monitoring agent. For more information, see "Installing agents on UNIX systems" on page 122 and "Installing agents on Linux systems" on page 128.
./*name*-agent.sh config *instance_name path_to_silent_config_file*	Configures the monitoring agent. Run the command from the *install_dir*/bin directory and add the response file path if required. If the agent has instances, enter an instance name. For more information about which agents are multiple instance agents, see the Table 7 on page 114 . The *silent_config_file* is optional. If you do not specify a file for silent configuration, you can configure the monitoring agent interactively by following the prompts.
./*name*-agent.sh uninstall	Uninstalls the monitoring agent. For more information, see "Uninstalling your agents" on page 147.
./smai-agent.sh uninstall_all	Uninstalls all the monitoring agents on the managed system.
./*name*-agent.sh remove *instance_name*	Removes an instance of a multiple instance agent.
./*name*-agent.sh	View a description of the functions that are available with the script.

- Windows

  On the system where you want to send a command to the monitoring agent, change to the *install_dir*\BIN directory at the command prompt, for example: C:\IBM\APM\bin. Enter any of the commands in Table 13 on page 181 where *name* is the agent name that is specified in Table 11 on page 178.

Table 13. Commands for Windows systems

Command	Description
name`-agent.bat status`	Checks the agent status. Checks the connection status between the agent and the Cloud APM server. Possible negative connection statuses are: Connection failed, Error detected, Disconnected-error. The positive status is Connected, this is the expected status. The transitional status is Connecting. A status of Unknown means that the agent status cannot be recognized, possible due to errors in the file system or in the agent log file.
name`-agent.bat start`	Starts the monitoring agent. If the agent has instances, enter an instance name after the command.
name`-agent.bat stop`	Stops the agent. If the agent has instances, enter an instance name after the command.
name`-agent.bat prereqcheck`	Runs a prerequisite scan. This command option is available for most agents.
name`-agent.bat install`	Installs the monitoring agent. For more information, see "Installing agents" on page 140.
name`-agent.bat config` *instance_name path_to_silent_config_file*	Configures the monitoring agent. Run the command from *install_dir*`\bin` directory and add the response file path if required. If the agent has instances, enter an instance name. For more information about which agents are multiple instance agents, see the Table 7 on page 114 . The *silent_config_file* is optional. If you do not specify a file for silent configuration, you can configure the monitoring agent interactively by following the prompts.
name`-agent.bat uninstall`	Uninstalls the monitoring agent. For more information, see "Uninstalling your agents" on page 147.
`smai-agent.bat uninstall_all`	Uninstalls all monitoring agents on the managed system.
name`-agent.bat remove` *instance_name*	Removes an instance of a multiple instance agent.
name`-agent.bat`	View a description of the functions that are available with the script.

Agent version command

- To see the version of an agent in your environment, run the following commands:
  - Linux AIX

```
install_dir/bin/cinfo
```

Enter 1 to show the versions.

- Windows

```
install_dir/InstallITM/kincinfo
```

Related tasks

"Using the IBM Cloud Application Performance Management window on Windows systems" on page 182

Using the IBM Cloud Application Performance Management window on Windows systems

Windows supported agents have a GUI utility that you can use to perform agent configuration and check the connection status.

The GUI configuration utility is not available for the Monitoring Agent for WebSphere MQ or Monitoring Agent for IBM Integration Bus.

Procedure

- Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Cloud Application Performance Management**.

Results

The IBM Cloud Application Performance Management window is displayed. Each installed agent component is listed with its configuration status, whether it is started or stopped, the connection status, the version number, and other information.

What to do next

Start or stop an agent or configure the parameters by right-clicking the agent and selecting an option.

Agent Configuration page

Use the **Agent Configuration** page to centrally configure settings for such agents as the Response Time Monitoring Agent and WebSphere Applications agent.

General usage

After you click **System Configuration** > **Agent Configuration** from the navigation bar, a tabbed dashboard is displayed with one tab for each configurable monitoring agent. The table shows columns of configuration information, such as the name and IP address for each managed system, one row for each managed system.

Actions

Use the Actions options to enable or disable such functions as transaction tracking or data collection.

Column resize

Drag a column heading border to adjust the column width.

Column sort

Click inside a column heading to sort by that column. Click the same column heading again to switch between ascending and descending sort order.

Table filter

Click inside the filter text box and type the beginning of the value to filter the table by. As you type, the table rows that do not fit the criteria are filtered out and the row ***Total*** is updated for the number of rows found.

Click the "x" in the filter box or press the Backspace key to clear the filter.

Agent configuration

For more information about the settings for the specific agents, see the following topics:

- DataPower agent: “Configuring DataPower monitoring” on page 230
- IBM Integration Bus agent: “Configuring transaction tracking for the IBM Integration Bus agent” on page 285
- Internet Service Monitoring“Configuring the agent on Windows systems” on page 438
- JBoss agent: “Setup the JBoss agent transaction tracking or diagnostics data collector” on page 460
- Microsoft .NET agent: “Enabling collection of transaction tracking and diagnostics data” on page 525
- OS agent log file monitoring: “Adding or removing log file monitoring configuration for the OS agents” on page 626
- Response Time Monitoring Agent: “Configuring using the Agent Configuration Page” on page 683
- Geolocation: “Customizing End User Transaction location values” on page 704
- Ruby agent: “Disabling or enabling diagnostics data for Ruby applications” on page 715
- SAP NetWeaver Java Stack agent: “Enabling the collection of transaction tracking and diagnostics data” on page 763
- Tomcat agent: “Enabling the collection of transaction tracking and diagnostics data” on page 799
- WebLogic agent: “Configuring transaction tracking for the WebLogic agent” on page 822
- WebSphere Applications agent: “Dynamically configuring data collection on Agent Configuration page” on page 869
- WebSphere MQ agent: “Configuring transaction tracking for the WebSphere MQ agent” on page 920

Configuring agents as a non-root user

If you want to configure your agent as a non-root user, create a common group on the system and make each user a member of this group.

Before you begin

If you installed your agent as a root or non-root user and you want to configure the agent as the same user, no special action is required.

If you installed your agent as a selected user and want to configure the agent as a different user, create a common group on the system. Make all agent management users members of this common group. Transfer ownership of all agent files and directories to this group.

Note:

- For the HTTP Server agent, if you configure the agent as a non-root user, the non-root user must have the same user ID as the user who started the IBM HTTP Server. Otherwise, the agent has problems with discovering the IBM HTTP Server.
- For the IBM Integration Bus agent, if IBM Integration Bus installation is a single-user deployment, use the same user ID as the user who installed IBM Integration Bus to configure the agent. Before you configure the agent, complete the following steps for this user ID.

Procedure

1. Install your monitoring agents on Linux or UNIX as described in “Installing agents on Linux systems” on page 128 and “Installing agents on UNIX systems” on page 122.
2. Run the `./secure.sh` script with the group name of the non-root user to secure the files and set the file group ownership to the files.
   For example: **`./secure.sh -g db2iadm1`**
3. Configure your monitoring agents on Linux or AIX as necessary, see Chapter 7, “Configuring your environment,” on page 159.
4. To update the system startup scripts, run the following script with root user or sudo user access: `*install_dir*/bin/UpdateAutoRun.sh`

What to do next

For more information about the **./secure.sh** script, see Securing the agent installation files.

Use the same user ID for agent installation and upgrades.

Disabling automatic agent start on UNIX and Linux systems

On the UNIX or Linux system, an agent can automatically start after an operating system restart. If you don’t want the agent to start automatically after system restart, you can disable automatic agent start.

About this task

If you install an agent as root user on the UNIX or Linux system, the agent can automatically start after system restart. Or, if you install an agent as non-root user but run the **UpdateAutoRun.sh** script as root after installation, the agent can automatically start after system restart.

Procedure

1. Complete the following steps to disable the automatic start on some agents:

   a. For the Linux OS agent and the WebSphere® Applications agent, add the following code to the `agent_install_dir/registry/kcirunas.cfg` file:

   ```
   <productCode>lz</productCode>
    <default>
      <autoStart>no</autoStart>
    </default>
    <productCode>yn</productCode>
    <default>
      <autoStart>no</autoStart>
    </default>
   ```

   b. Run the **agent_install_dir/bin/UpdateAutoRun.sh** command.

2. Complete the following steps to enable the automatic start on some agents:

   a. For the Linux OS agent and the WebSphere® Applications agent, in the `agent_install_dir/registry/kcirunas.cfg` file, change the value of the *<autoStart>* tag to **yes**.

   b. Open the `agent_install_dir/registry/AutoStart` file and check the content.

   c. Delete the `/etc/init.d/ITMAgents{$Num}` file, where *{$Num}* is a positive number in the `agent_install_dir/registry/AutoStart` file. If the value is 1, you must delete the `/etc/init.d/ITMAgents1` file.

   d. Run the **agent_install_dir/bin/UpdateAutoRun.sh** command.

Results

After system restart, an agent script will not automatically run to start the agent.

General procedure for configuring data collectors

To use a data collector to view monitoring data in the Cloud APM console for your applications, you must complete several configuration tasks.

About this task

This procedure is a roadmap for configuring the monitoring for your applications, which includes both required, conditional, and optional steps. Complete the necessary steps according to your needs.

Procedure

1. Download and extract the data collector package. For instructions, see “Downloading your agents and data collectors” on page 105.

2. Configure the data collector to collect monitoring data about IBM Cloud and on-premises applications and send it to the Cloud APM server. Complete one or more of the following tasks according to the type of your application:

   Liberty applications

   - "Configuring the Liberty data collector in on-premises environments (Liberty V18.* and older versions)" on page 464
   - "Configuring the Liberty data collector in IBM Cloud environment (Liberty V18.* and older versions)" on page 468

   Node.js applications

   - "Configuring the stand-alone Node.js data collector for IBM Cloud(formerly Bluemix) applications" on page 590
   - "Configuring the stand-alone Node.js data collector for on-premises applications" on page 595

   Python applications

   - "Configuring the Python data collector for IBM Cloud applications" on page 664
   - "Configuring the Python data collector for on-premises applications" on page 669

   Ruby applications

   - "Configuring the Ruby data collector for IBM Cloud applications" on page 715

   Java applications

   - "Configuring J2SE monitoring" on page 442
3. If the key file or the Cloud APM server changes, reconnect the data collector to the Cloud APM server. For instructions, see "Reconnecting the data collector to the Cloud APM server" on page 185.

What to do next

After you complete all necessary configuration tasks, you can verify that the monitoring data for your IBM Cloud application is displayed in the Cloud APM console.

Reconnecting the data collector to the Cloud APM server

If the Cloud APM server, key file, or the key file password is changed, you must set several environment variables to reconnect the data collector to the Cloud APM server.

Before you begin

If the key file is changed, encrypt the plain text password of your key file by using Base64 first. If you are a Linux user, run the following command:

```
echo -n keyfile_password | base64
```

The command output is your encrypted password. For example, if your plain text password is `password`, the command output `cGFzc3dvcmQ=` is your encrypted password. You then use the encrypted password to set `APM_KEYFILE_PSWD: `*`encrypted_keyfile_password`* and `APM_KEYFILE_PSWD=`*`encrypted_keyfile_password`* in the following configurations.

Procedure

- To reconnect the data collectors to the Cloud APM server for IBM Cloud applications, see "Reconnecting the data collectors for IBM Cloud applications" on page 186.
- To reconnect the data collectors to the Cloud APM server for on-premises, see "Reconnecting the data collector for on-premises applications" on page 187.

Reconnecting the data collectors for IBM Cloud applications

About this task

You have the following two options to reconnect the data collector to the Cloud APM server:

- Edit the `manifest.yml` of your application to set the variables.
- Set the variables on the IBM Cloud UI.

Procedure

- To use the `manifest.yml` file of your IBM Cloud application to reconnect the data collector, complete the following steps:
  a) Edit the variables in the `manifest.yml` file of your IBM Cloud application according to the changes.
    - To configure the Gateway to use *HTTP*, set the following variable:

      ```
      APM_BM_GATEWAY_URL: http://server_ip_or_hostname:80
      ```

    - To configure the Gateway to use *HTTPS*, set the following three variables:

      ```
      APM_BM_GATEWAY_URL: https://server_ip_or_hostname:443
      APM_KEYFILE_PSWD: encrypted_keyfile_password
      APM_KEYFILE_URL: http://hosted_http_server:port/keyfile_name
      ```

      Tip: The key file for the Liberty data collector is a `.jks` file. For the Python, Node.js, and Liberty data collectors, the key files are `.p12` files.

  b) Change to the directory of your IBM Cloud application, and run the following command:

    ```
    cf push
    ```

- To use the IBM Cloud UI to reconnect the data collector, complete the following steps:
  a) Log in to the IBM Cloud UI.
  b) Click the IBM Cloud application.
  c) Click **Runtime** on the left panel.
  d) Switch to the **Environment variable** tab.
  e) In the **user-defined** section, use one of the following methods to define the variables according to your needs:
    - To configure the Gateway to use *HTTP*, set the following variable:

      ```
      APM_BM_GATEWAY_URL: http://server_ip_or_hostname:80
      ```

    - To configure the Gateway to use *HTTPS*, set the following three variables:

      ```
      APM_BM_GATEWAY_URL: https://server_ip_or_hostname:443
      APM_KEYFILE_PSWD: encrypted_keyfile_password
      APM_KEYFILE_URL: http://hosted_http_server:port/keyfile_name
      ```

      Tip: The key file for the Liberty data collector is a `.jks` file. For the Python, Node.js, and Liberty data collectors, the key files are `.p12` files.

  f) From the directory where you run the **`cf push`** command to push your application, run the following command for your changes to take effect:

    ```
    cf restage <app_name>
    ```

Results

The values of the variables are properly set to connect the data collector to the Cloud APM server.

Reconnecting the data collector for on-premises applications

About this task

By modifying the `global.environment` or the `dc.java.properties` file, you can customize the connection between the data collector and the Cloud APM server.

Procedure

1. Find the corresponding file that contains the connection variables.
   a) For the Liberty data collector, Node.js data collector, and Python data collector, find the `global.environment` file according to the information in the following table:

Data collector name	Directory to the `global.environment` file
Liberty data collector	The `itcamdc/etc/global.environment` folder where your Liberty data collector is installed.
Node.js data collector	The `ibmapm/etc` folder where your Node.js data collector is installed.
Python data collector	The `etc` folder where your Python data collector is installed.

   b) For the J2SE data collector, find the `dc.java.properties` file in the *`DC_HOME`*`/itcamdc/etc` folder. *DC_HOME* is the directory where your J2SE data collector is installed.
2. Edit the variables in the corresponding file according to the changes.
   a) For the Liberty data collector, Node.js data collector, and Python data collector, edit the `global.environment` file according to the following instruction:
   - To configure the Gateway to use *HTTP*, set the following variable:

```
APM_BM_GATEWAY_URL=http://server_ip_or_hostname:80
```

   - To configure the Gateway to use *HTTPS*, set the following variables:

```
APM_BM_GATEWAY_URL=https://server_ip_or_hostname:443
APM_KEYFILE_PSWD=encrypted_keyfile_password
APM_KEYFILE_URL=http://hosted_http_server_:port/keyfile_name
```

   Tip: The key file for the Liberty data collector is a `.jks` file. For the Python, Node.js, and Liberty data collectors, the key files are `.p12` files.

   b) For the J2SE data collector, edit the `dc.java.properties` file according to the following instruction:
   - To configure the Gateway to use *HTTP*, set the following variable:

```
apm.http.type=http
```

   If the value of this variable is left empty, `http` is the default value
   - To configure the Gateway to use *HTTPS*, set the following variables:

```
apm.ssl.password=encrypted_keyfile_password
apm.http.type=https
```

   Important: If the password is changed, replace the *DC_HOME*`/itcamdc/etc/keyfile.jks` file with the `/opt/ibm/ccm/keyfiles/default.agent/keyfiles/keyfile.jks` file from the Cloud APM server, where *DC_HOME* is the home directory of your J2SE data collector.

3. Optional: If you do not use the default key file for your Node.js data collector, set the following variable:

   ```
   APM_SNI=owner_host_in_the_key_file
   ```

   Tip: To find out the value of the *owner host* variable, open the key file that you use and search for `owner`. And then set the *APM_SNI* variable to the same value of *owner*.
4. Restart the application for the change to take effect.

Results

The values of the variables are properly set to connect the data collector to the Cloud APM server.

Sample `manifest.yml` file

Refer to the following lines for the content of the `manifest.yml` file of an IBM Cloud application:

```
applications:

- disk_quota: 1024M
  host: myBluemixApp
  name: myBluemixApp
  path: .
  domain: mybluemix.net

  instances: 1
  memory: 512M

  env:
    KNJ_ENABLE_TT: "true"
    KNJ_SAMPLING: 1
```

Removing data collectors from Cloud APM console

After you unconfigure a data collector, you should also remove the data collector from the applications and from the resource groups that it was added to. Otherwise, the Cloud APM console will show that no data is available for the application and will not indicate that the data collector is offline.

Procedure

1. Remove the data collector from any applications that you added it to by manually editing the applications.

   It is similar to removing offline agents from your application, see "Viewing and removing offline agents" on page 1078.
2. Remove the data collector from any custom resource groups that you added it to.

   For more information, see "Resource Group Manager" on page 962.
3. Open a ticket for the Cloud APM Operations team to perform the following steps on the Cloud APM server:

   a) Edit the *install_dir*`/serveragents/config/hostname_bi.cfg` file to remove the lines for the data collector that has been unconfigured.

   b) Restart the server component for data collectors by running the following command as root user:

   ```
   apm restart biagent
   ```

Results

After a few minutes, the Cloud APM console will indicate that the data collector is offline in the **My Components** application and in the **Resource Group Manager** UI when you select the system resource group for the data collector.

After the interval specified by the **Remove Offline System Delay** configuration property in the **Advanced Configuration** page, the data collector will automatically be removed from **My Components** and from its system resource group.

Tip: You can adjust the **Remove Offline System Delay** setting in the **Advanced Configuration** page to increase or decrease the wait time before the offline agent is removed from view. For more information, see“Agent Subscription Facility” on page 1051.

Remember: If the data collector provided transaction tracking data to the Cloud APM server, the Cloud APM console might continue to display the data collector in the **My Components** application and display the message of `The agent is invalid` for the data collector after the time period specified by the **Remove Offline System Delay** setting has expired. If you have installed Cloud APM 8.1.4.0 Server Interim Fix 3 or later, an invalid data collector will eventually be removed from the **My Components** application 8 days after transaction tracking data stops being received from the data collector.

Configuring Amazon EC2 monitoring

The Amazon EC2 agent provides you with a central point of monitoring for the health, availability, and performance of your Amazon Elastic Compute Cloud (EC2) Instances. The agent displays a comprehensive set of metrics to help you make informed decisions about your EC2 environment. These metrics include CPU usage, Elastic Block Store (EBS) usage, network usage, Amazon Web Services (AWS) maintenance updates, and disk performance.

Before you begin

- Read the entire “Configuring Amazon EC2 monitoring” on page 189 topic to determine what is needed to complete the configuration.
- The directions here are for the most current release of the agent, except as indicated.
- Make sure that the system requirements for the Amazon EC2 agent are met in your environment. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Amazon EC2 agent.
- Ensure that the following information is available:
  - A list of the AWS region names that contain EC2 instances to monitor.
  - The AWS security credentials (Access Key ID and Secret Access Key) with permission to access each AWS region.
- Ensure that the AWS security credentials that are used for each AWS region are a member of a group that includes at least the *AmazonEC2ReadOnlyAccess* policy.

About this task

The Amazon EC2 agent is both a multiple instance agent and also a subnode agent. You can create one agent instance with multiple subnodes – one for each Amazon EC2 region, or you can create an agent instance for each Amazon EC2 region with one subnode for that region. Or you can create a combination of each type of configuration. After you configure agent instances, you must start each agent instance manually. If you have more than 50 resources per Amazon EC2 region, it is suggested that you create an agent instance per region or use tagging on your EC2 instances and filter agent instances by the tags you create by using the agent's filtering condition parameter.

Procedure

1. Configure the agent on Windows systems with the **IBM Performance Management** window or the silent response file.
   - “Configuring the agent on Windows systems” on page 190.
   - “Configuring the agent by using the silent response file” on page 194.
2. Configure the agent on Linux systems with the script that prompts for responses or the silent response file.

- “Configuring the agent by responding to prompts” on page 193.
- “Configuring the agent by using the silent response file” on page 194.

What to do next

In the Cloud APM console, go to your Application Performance Dashboard to view the data that was collected. For more information about using the Cloud APM console, see “Starting the Cloud APM console” on page 957.

If you are unable to view the data in the agent dashboards, first check the server connection logs and then the data provider logs. The default paths to these logs are listed here:

- Linux `/opt/ibm/apm/agent/logs`
- Windows `C:\IBM\APM\TMAITM6_x64\logs`

For help with troubleshooting, see the Cloud Application Performance Management Forum.

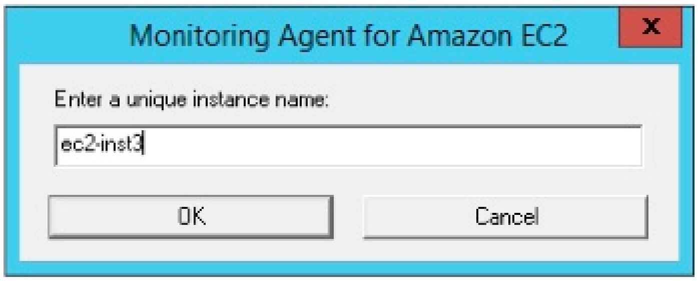
Configuring the agent on Windows systems

You can configure the Amazon EC2 agent on Windows operating systems by using the IBM Cloud Application Performance Management window. After you update the configuration values, you must start the agent to save the updated values.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Cloud Application Performance Management**.
2. In the **IBM Performance Management** window, right-click the **Monitoring Agent for Amazon EC2** template, and then click **Configure agent**.

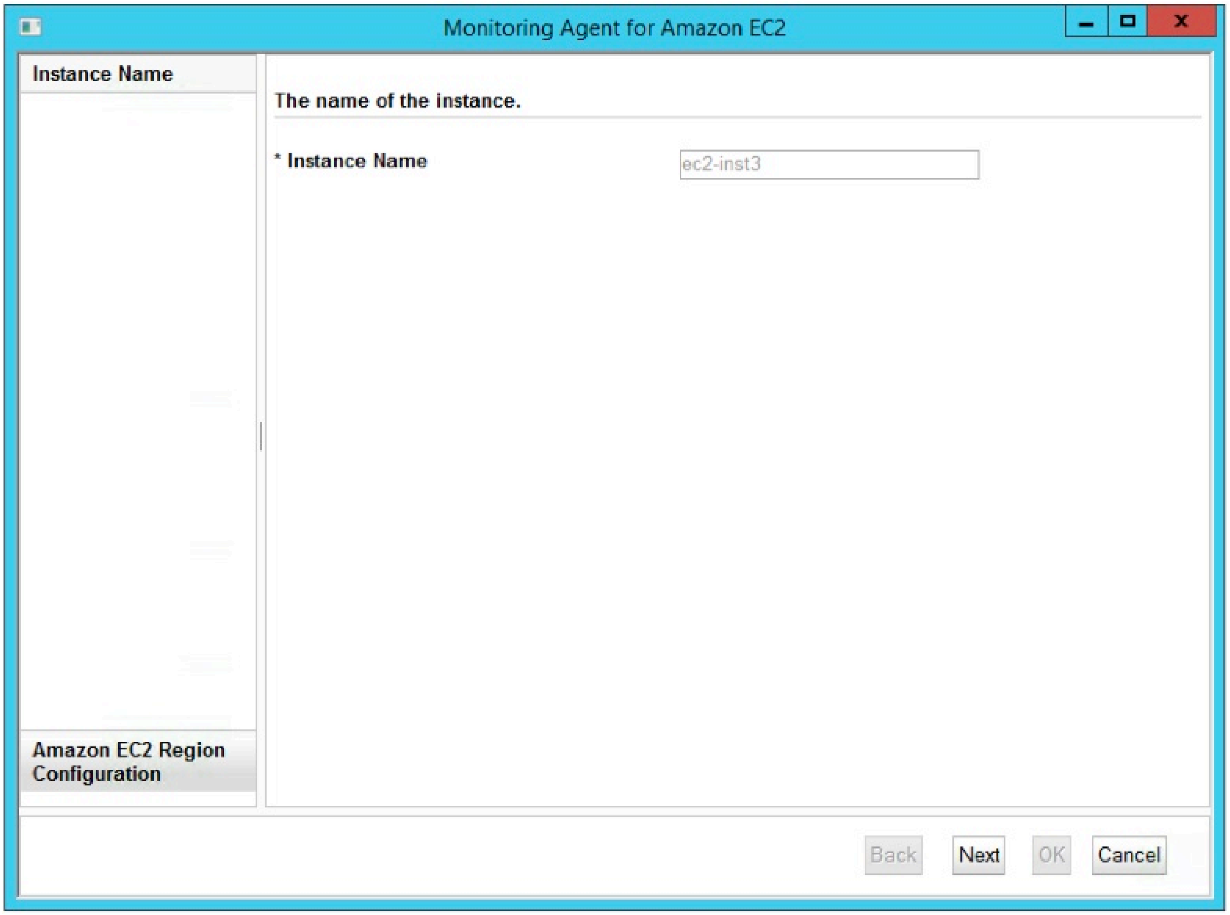
   Remember: After you configure an agent instance for the first time, the **Configure agent** option is disabled. To configure the agent instance again, right-click on it and then click **Reconfigure**.
3. Enter a unique instance name then click **OK**. Use only Latin letters, Arabic numerals, and the hyphen-minus character in the instance name. Example, `ec2-inst3`.

   Monitoring Agent for Amazon EC2

   Enter a unique instance name: ec2-inst3

   Figure 3. The window to enter a unique instance name.
4. Click **Next** on the agent instance name window.


Monitoring Agent for Amazon EC2
Instance Name
Amazon EC2 Region Configuration
Back
Next
OK
Cancel

The name of the instance.

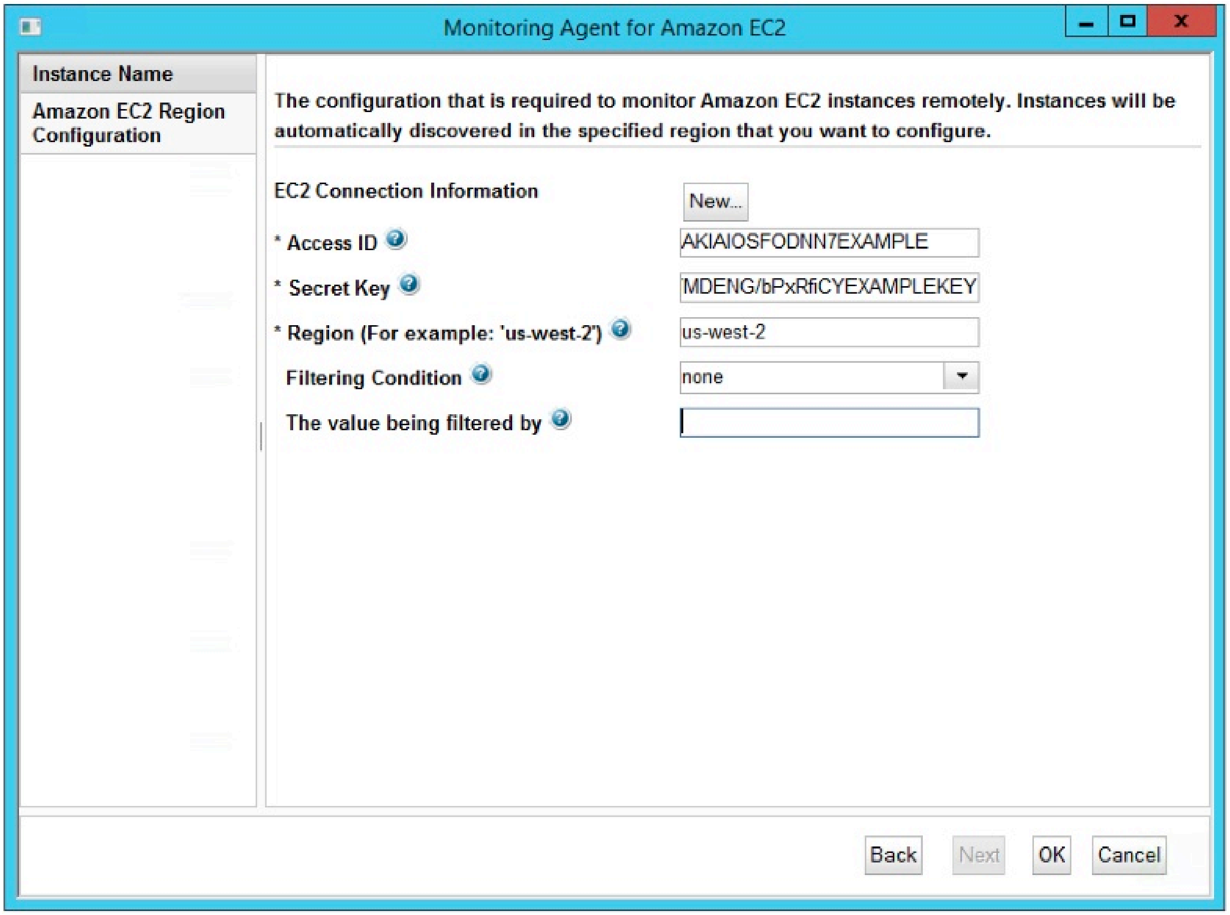
* Instance Name ec2-inst3

Figure 4. The agent instance name window.

5. Enter the **Amazon EC2 Region Configuration** instance template settings.

   Note: This section is not the Amazon EC2 region instance configuration. It is a template section for settings that are used as the default values when you add the actual Amazon EC2 region instance configurations in step 6.

   See Table 14 on page 195 for an explanation of each of the configuration parameters.



Monitoring Agent for Amazon EC2
Instance Name
Amazon EC2 Region Configuration
New...
Back
Next
OK
Cancel

The configuration that is required to monitor Amazon EC2 instances remotely. Instances will be automatically discovered in the specified region that you want to configure.

EC2 Connection Information

* Access ID: AKIAIOSFODNN7EXAMPLE

* Secret Key: MDENG/bPxRfiCYEXAMPLEKEY

* Region (For example: 'us-west-2'): us-west-2

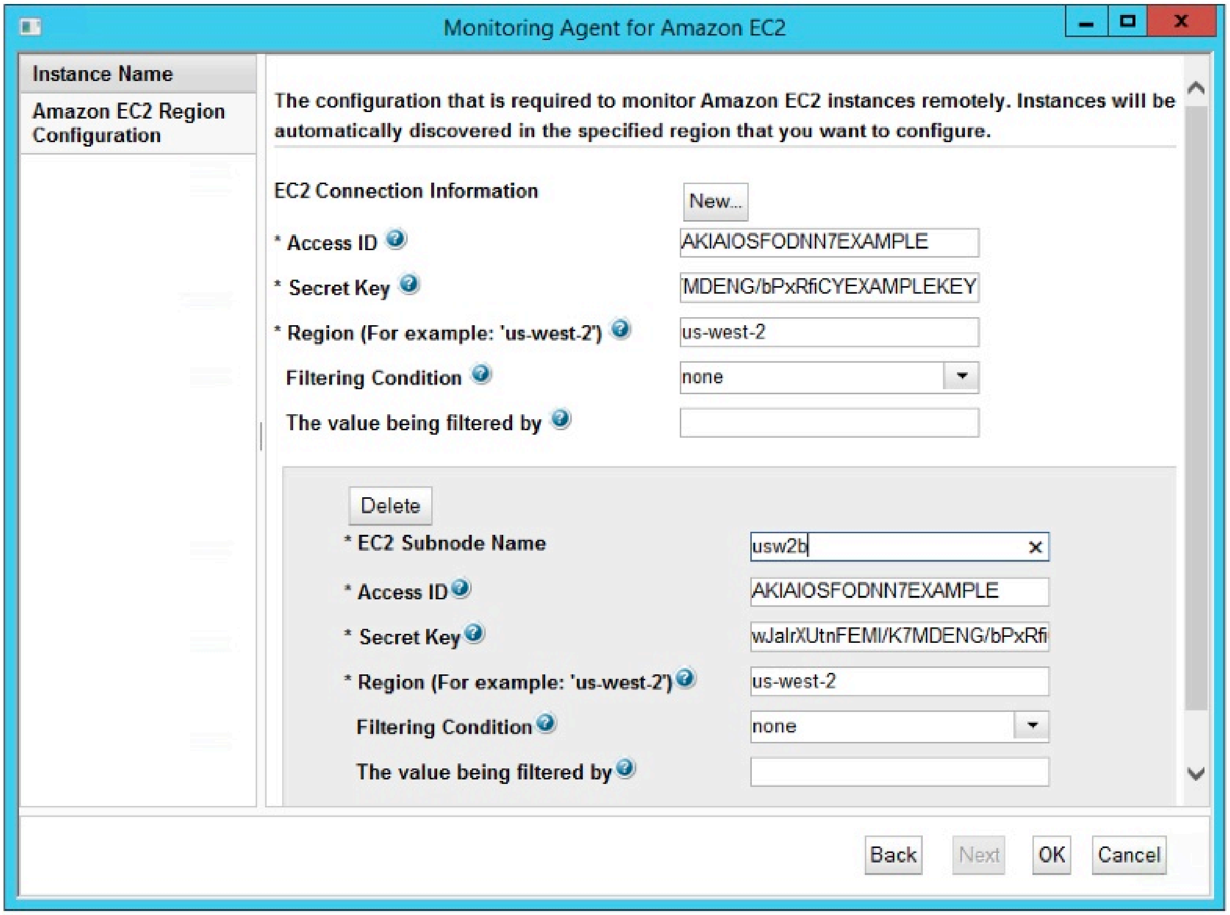
Filtering Condition: none

The value being filtered by:

Figure 5. The window to specify Amazon EC2 region instance template settings.

6. Press **New** and enter Amazon EC2 region instance settings, then click **Next**.

   See Table 14 on page 195 for an explanation of each of the configuration parameters.


Monitoring Agent for Amazon EC2
Instance Name
Amazon EC2 Region Configuration
New...
Delete
Back
Next
OK
Cancel

The configuration that is required to monitor Amazon EC2 instances remotely. Instances will be automatically discovered in the specified region that you want to configure.

EC2 Connection Information

* Access ID: AKIAIOSFODNN7EXAMPLE

* Secret Key: MDENG/bPxRfiCYEXAMPLEKEY

* Region (For example: 'us-west-2'): us-west-2

Filtering Condition: none

The value being filtered by:

* EC2 Subnode Name: usw2b

* Access ID: AKIAIOSFODNN7EXAMPLE

* Secret Key: wJalrXUtnFEMI/K7MDENG/bPxRfi

* Region (For example: 'us-west-2'): us-west-2

Filtering Condition: none

The value being filtered by:

Figure 6. The window to specify Amazon EC2 region instance settings.

7. Click **OK** to complete the configuration.
8. In the IBM Cloud Application Performance Management window, right-click the instance that you configured, and then click **Start**.

Configuring the agent by responding to prompts

After installation of the Amazon EC2 agent, you must configure it before you start the agent. If the Amazon EC2 agent is installed on a local Linux computer, you can follow these instructions to configure it interactively through command line prompts.

About this task

Remember: If you are reconfiguring a configured agent instance, the value that is set in the last configuration is displayed for each setting. If you want to clear an existing value, press the space key when the setting is displayed.

Procedure

Follow these steps to configure the Amazon EC2 agent by running a script and responding to prompts.

1. Run the following command:

   ```
   install_dir/bin/amazonec2-agent.sh config instance_name
   ```

   where *install_dir* is the path where the agent is installed and *instance_name* is the name that you want to give to the agent instance.

   Example

```
/opt/ibm/apm/agent/bin/amazonec2-agent.sh config ec2-inst3
```

2. Respond to the prompts to set configuration values for the agent.

   See "Configuration parameters for the Amazon EC2 agent" on page 195 for an explanation of each of the configuration parameters.

3. Run the following command to start the agent:

   ```
   install_dir/bin/amazonec2-agent.sh start instance_name
   ```

   where *install_dir* is the path where the agent is installed and *instance_name* is the name of the agent instance.

   Example

   ```
   /opt/ibm/apm/agent/bin/amazonec2-agent.sh start ec2-inst3
   ```

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

The silent response file contains the agent configuration parameters with default values that are defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

- Configure the Amazon EC2 agent in the silent mode:

  a) Open the `amazonec2_silent_config.txt` file at one of the following paths in a text editor.

  – Linux *install_dir*`/samples/amazonec2_silent_config.txt`

    Example, `/opt/ibm/apm/agent/samples/amazonec2_silent_config.txt`

  – Windows *install_dir*`\samples\amazonec2_silent_config.txt`

    Example, `C:\IBM\APM\samples\amazonec2_silent_config.txt`

  where *install_dir* is the path where the agent is installed.

  b) In the `amazonec2_silent_config.txt` file, specify values for all mandatory parameters and modify the default values of other parameters as needed.

  See "Configuration parameters for the Amazon EC2 agent" on page 195 for an explanation of each of the configuration parameters.

  c) Save and close the `amazonec2_silent_config.txt` file, and run the following command:

  – Linux *install_dir*`/bin/amazonec2-agent.sh config` **`instance_name`** *install_dir*`/samples/amazonec2_silent_config.txt`

    Example, **`/opt/ibm/apm/agent/bin/amazonec2-agent.sh config ec2-inst3 /opt/ibm/apm/agent/samples/amazonec2_silent_config.txt`**

  – Windows *install_dir*`\bin\amazonec2-agent.bat config` **`instance_name`** *install_dir*`\samples\amazonec2_silent_config.txt`

    Example, **`C:\IBM\APM\bin\amazonec2-agent.bat config ec2-inst3 C:\IBM\APM\samples\amazonec2_silent_config.txt`** where *install_dir* is the path where the agent is installed and *instance_name* is the name that you want to give to the agent instance.

Important: Ensure that you include the absolute path to the silent response file. Otherwise, the agent data is not shown in the dashboards.

d) Run the following command to start the agent:

- Linux *`install_dir`*`/bin/amazonec2-agent.sh start` **`instance_name`**

  Example, **`/opt/ibm/apm/agent/bin/amazonec2-agent.sh start ec2-inst3`**

- Windows *`install_dir`*`\bin\amazonec2-agent.bat start` **`instance_name`**

  Example, **`C:\IBM\APM\bin\amazonec2-agent.bat start ec2-inst3`**

where *install_dir* is the path where the agent is installed and *instance_name* is the name of the agent instance.

Configuration parameters for the Amazon EC2 agent

The configuration parameters for the Amazon EC2 agent are displayed in a table.

1. Amazon EC2 Region Configuration - Settings to monitor Amazon EC2 instances remotely. Instances are automatically discovered in the specified region that you want to configure.

Table 14. Amazon EC2 Region Configuration

Parameter name	Description	Silent configuration file parameter name
EC2 Subnode Name	Name of the EC2 Subnode for collection of data. Example, *usw2a*. This alias is part of the managed system name (MSN) and it is used to visually identify the monitored environment in the Cloud APM console. **Note:** This alias can be anything that you choose to represent the Amazon EC2 subnode instance with the following restrictions. Letters from the Latin alphabet (a-z, A-Z), Arabic numerals (0-9), the hyphen-minus character (-), and the underscore character (_) can be used to create agent subnode instance names. The maximum length of an EC2 subnode name is 25 characters.	Each of the following parameters must have an agent subnode name suffix that is the same for each parameter of an agent subnode instance. New agent subnode instances must use a unique name for its set of parameters. For example, one agent subnode instance can use *.usw2a* and another agent subnode instance can use *.usw2b* in place of *.subnode_name* in the parameter names that follow.
Access ID	AWS security credentials Access Key ID that is used to authenticate with the specified Amazon Region. For example, 'AKIAxxxxxxxxxxxxxxx'.	**`KB5_INS_ACCESS_ID.subnode_name`**
Secret Key	AWS security credentials Secret Access Key that is used to authenticate with the specified Amazon Region. For example, 'kK7txxxxxxxxxxxxxxxxxxxxxxxxx'.	**`KB5_INS_SECRET_KEY.subnode_name`**
Region	AWS Region to monitor. For example, 'us-west-2'.	**`KB5_INS_REGION.subnode_name`**

Table 14. Amazon EC2 Region Configuration (continued)

Parameter name	Description	Silent configuration file parameter name
Filtering Condition	The type of filtering that is being done. You can use custom tags on EC2 instances to limit which EC2 instances are monitored by the agent. For more information, see Tagging Your Amazon EC2 Resources. Filtering options, **none** All EC2 instances within the region are monitored. **Filter Value** is ignored. **tagName** EC2 instances with the tag key that is specified in **Filter Value** are monitored, regardless of the actual value in the corresponding EC2 instance tag value. For example, to monitor all EC2 instances that have the tag key *Stack*, regardless of the value in its tag value, specify Stack in **Filter Value**. **tagName|tagValue** EC2 instances with the tag key and tag value pair that is separated with a vertical bar (|), and that is specified in **Filter Value** are monitored. For example, to monitor all EC2 instances that have the tag key *Stack* and the tag value *Production*, specify Stack|Production in **Filter Value**. **monitoring-tag** EC2 instances that have at least one tag are monitored. **Filter Value** is ignored.	**FILTER_CONDITION.subnode_name** Valid values, **none** none **tagName** tagName **tagName|tagValue** tagName|tagValue **monitoring-tag** monitoring-tag
Filter Value	The value of the tag by which the EC2 instances are filtered when either tagName or tagName|tagValue are selected for **Filtering Condition**.	**FILTER_VALUE.subnode_name**

Configuring AWS Elastic Load Balancer monitoring

The Amazon ELB agent provides you with a central point of monitoring for the health, availability, and performance of your AWS Elastic Load Balancers. The agent displays a comprehensive set of metrics for each load balancer type-application, network and classic-to help you make informed decisions about your AWS Elastic Load Balancer environment.

Before you begin

- Read the entire "Configuring AWS Elastic Load Balancer monitoring" on page 196 topic to determine what is needed to complete the configuration.
- The directions here are for the most current release of the agent, except as indicated.

- Make sure that the system requirements for the Amazon ELB agent are met in your environment. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Amazon ELB agent.
- Ensure that the following information is available:
  - The AWS security credentials (Access Key ID and Secret Access Key) with permission to access each AWS region with Elastic Load Balancers.

About this task

The Amazon ELB agent is both a multiple instance agent and also a subnode agent. The subnodes are created automatically for each type of Elastic Load Balancer that is available in your AWS environment.

Procedure

1. Configure the agent on Windows systems with the **IBM Performance Management** window or the silent response file.
   - "Configuring the agent on Windows systems" on page 197.
   - "Configuring the agent by using the silent response file" on page 199.
2. Configure the agent on Linux systems with the script that prompts for responses or the silent response file.
   - "Configuring the agent by responding to prompts" on page 198.
   - "Configuring the agent by using the silent response file" on page 199.

What to do next

In the Cloud APM console, go to your Application Performance Dashboard to view the data that was collected. For more information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you are unable to view the data in the agent dashboards, first check the server connection logs and then the data provider logs. The default paths to these logs are listed here:

- Linux `/opt/ibm/apm/agent/logs`
- Windows `C:\IBM\APM\TMAITM6_x64\logs`

For help with troubleshooting, see the Cloud Application Performance Management Forum.

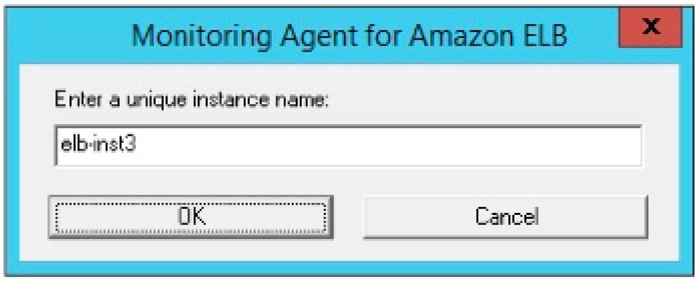
Configuring the agent on Windows systems

You can configure the Amazon ELB agent on Windows operating systems by using the IBM Cloud Application Performance Management window. After you update the configuration values, you must start the agent to save the updated values.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Cloud Application Performance Management**.
2. In the **IBM Performance Management** window, right-click the **Monitoring Agent for AWS Elastic Load Balancer** template, and then click **Configure agent**.

   Remember: After you configure an agent instance for the first time, the **Configure agent** option is disabled. To configure the agent instance again, right-click on it and then click **Reconfigure**.
3. Enter a unique instance name then click **OK**. Use only Latin letters, Arabic numerals, and the hyphen-minus character in the instance name. Example, `elb-inst3`. For more information, see *instancename* under "Common MSN format for multi-instance agents" on page 168.



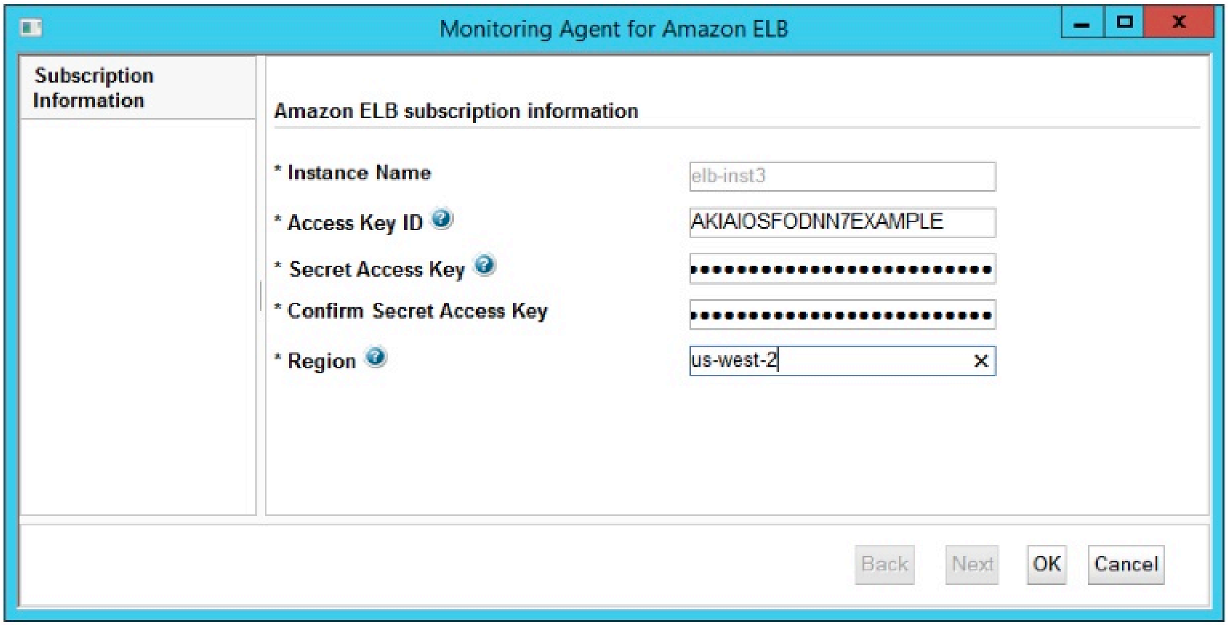
Enter a unique instance name: elb-inst3

Figure 7. The window to enter a unique agent instance name.

4. Enter the **Amazon ELB Subscription Credentials**, then click **Next**.

   See "Configuration parameters for the Amazon ELB agent" on page 201 for an explanation of each of the configuration parameters.

   Important: Windows If your **`Secret Key/Password`** contains an equal sign (=), you must reenter it each time that you reconfigure the agent.



Amazon ELB subscription information

* Instance Name: elb-inst3
* Access Key ID: AKIAIOSFODNN7EXAMPLE
* Secret Access Key: ••••••••••••••••••••••••••
* Confirm Secret Access Key: ••••••••••••••••••••••••••
* Region: us-west-2

Figure 8. The Amazon ELB subscription credentials window.

5. Click **OK** to complete the configuration.
6. In the IBM Cloud Application Performance Management window, right-click the instance that you configured, and then click **Start**.

Configuring the agent by responding to prompts

After installation of the Amazon ELB agent, you must configure it before you start the agent. If the Amazon ELB agent is installed on a local Linux computer, you can follow these instructions to configure it interactively through command line prompts.

About this task

Remember: If you are reconfiguring a configured agent instance, the value that is set in the last configuration is displayed for each setting. If you want to clear an existing value, press the space key when the setting is displayed.

Procedure

Follow these steps to configure the Amazon ELB agent by running a script and responding to prompts.

1. Run the following command:

```
install_dir/bin/amazon_elb-agent.sh config instance-name
```

Where *install_dir* is the path where the agent is installed and *instance-name* is the name that you want to give to the agent instance. Use only Latin letters, Arabic numerals, and the hyphen-minus character in the *instance-name*. For more information, see *instancename* under "Common MSN format for multi-instance agents" on page 168.

Example

```
/opt/ibm/apm/agent/bin/amazon_elb-agent.sh config elb-inst3
```

2. Respond to the prompts to set configuration values for the agent.

   See "Configuration parameters for the Amazon ELB agent" on page 201 for an explanation of each of the configuration parameters.

3. Run the following command to start the agent:

   ```
   install_dir/bin/amazon_elb-agent.sh start instance-name
   ```

   Where *install_dir* is the path where the agent is installed and *instance-name* is the name of the agent instance.

   Example

   ```
   /opt/ibm/apm/agent/bin/amazon_elb-agent.sh start elb-inst3
   ```

Example

Creating an agent instance that is named `elb-inst3`.

```
# ./amazon_elb-agent.sh config elb-inst3
Configuring Monitoring Agent for Amazon ELB

Edit 'Monitoring Agent for Amazon ELB' settings? [ 1=Yes, 2=No ]  (default is: 1): 1

Subscription Information :
Amazon ELB subscription information

The access ID that is used to authenticate with the specified Amazon Region.
For example, 'AKIAxxxxxxxxxxxxxxx'.
Access Key ID (default is: ): AKIAIOSFODNN7EXAMPLE

The secret access key that is used to authenticate with the specified Amazon
Region. For example, 'kK7txxxxxxxxxxxxxxxxxxxxxxxxx'.
Enter Secret Access Key (default is: ): hidden

Re-type : Secret Access Key (default is: ): hidden

The Amazon region where the load balancers are located. For example, 'us-west-2'.
Region (default is: ): us-west-2

Configuration completed successfully.
Automatic start at system initialization has been configured.
Automatic stop at system shutdown has been configured.
```

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

The silent response file contains the agent configuration parameters with default values that are defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

Follow these steps to configure the Amazon ELB agent in the silent mode.

1. Open the `amazon_elb_silent_config.txt` file at one of the following paths in a text editor.
   - Linux *`install_dir`*`/samples/amazon_elb_silent_config.txt`

     Example, `/opt/ibm/apm/agent/samples/amazon_elb_silent_config.txt`
   - Windows *`install_dir`*`\samples\amazon_elb_silent_config.txt`

     Example, `C:\IBM\APM\samples\amazon_elb_silent_config.txt`

   Where *install_dir* is the path where the agent is installed.
2. In the `amazon_elb_silent_config.txt` file, specify values for all mandatory parameters and modify the default values of other parameters as needed.

   See "Configuration parameters for the Amazon ELB agent" on page 201 for an explanation of each of the configuration parameters.
3. Save and close the `amazon_elb_silent_config.txt` file, and run the following command:
   - Linux *`install_dir`*`/bin/amazon_elb-agent.sh config` **`instance-name`** *`install_dir`*`/samples/amazon_elb_silent_config.txt`

     Example, **`/opt/ibm/apm/agent/bin/amazon_elb-agent.sh config elb-inst3 /opt/ibm/apm/agent/samples/amazon_elb_silent_config.txt`**
   - Windows *`install_dir`*`\bin\amazon_elb-agent.bat config` **`instance-name`** *`install_dir`*`\samples\amazon_elb_silent_config.txt`

     Example, **`C:\IBM\APM\bin\amazon_elb-agent.bat config elb-inst3 C:\IBM\APM\samples\amazon_elb_silent_config.txt`**

   Where *install_dir* is the path where the agent is installed and *instance-name* is the name that you want to give to the agent instance. Use only Latin letters, Arabic numerals, and the hyphen-minus character in the *instance-name*. For more information, see *instancename* under "Common MSN format for multi-instance agents" on page 168.

   Important: Ensure that you include the absolute path to the silent response file. Otherwise, the agent data is not shown in the dashboards.
4. Run the following command to start the agent:
   - Linux *`install_dir`*`/bin/amazon_elb-agent.sh start` **`instance-name`**

     Example, **`/opt/ibm/apm/agent/bin/amazon_elb-agent.sh start elb-inst3`**
   - Windows *`install_dir`*`\bin\amazon_elb-agent.bat start` **`instance-name`**

     Example, **`C:\IBM\APM\bin\amazon_elb-agent.bat start elb-inst3`**

   Where *install_dir* is the path where the agent is installed and *instance-name* is the name of the agent instance.

Example

Edited `amazon_elb_silent_config.txt`.

```
#
# This is a sample configuration response file for agent Amazon ELB.
#
# It contains an entry for every configuration property.
# Entries for optional properties that have no default value are included
# in comments.
# Ensure that all uncommented properties have a value before configuring
# the agent.
#

# Access Key ID: The access ID that is used to authenticate with the
# specified Amazon Region. For example, 'AKIAxxxxxxxxxxxxxxx'.
KAL_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE

# Secret Access Key: The secret access key that is used to authenticate with
# the specified Amazon Region. For example, 'kK7txxxxxxxxxxxxxxxxxxxxxxxxxxx'.
KAL_SECRET_ACCESS_KEY_PASSWORD=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

# Region: The Amazon region where the load balancers are located. For
# example, 'us-west-2'.
KAL_REGION=us-west-2
```

Configuration parameters for the Amazon ELB agent

The configuration parameters for the Amazon ELB agent are displayed in a table.

1. Table 15 on page 201 - Credentials that are required for access to the Amazon Subscription that contains the AWS Elastic Load Balancers to monitor.

Table 15. Subscription information

Parameter name	Description	Silent configuration file parameter name
Access Key ID	The access ID that is used to authenticate with the specified Amazon Region. For example, 'AKIAxxxxxxxxxxxxxxx'.	**KAL_ACCESS_KEY_ID**
Secret Access Key	The secret access key that is used to authenticate with the specified Amazon Region. For example, 'kK7txxxxxxxxxxxxxxxxxxxxxxxxxxx'.	**KAL_SECRET_ACCESS_KEY_PASSWORD**
Region	The Amazon region where the load balancers are located. For example, 'us-west-2'.	**KAL_REGION**

Configuring Azure Compute monitoring

The Azure Compute agent provides you with a central point of monitoring for the health, availability, and performance of your Azure Compute instances. The agent displays a comprehensive set of metrics to help you make informed decisions about your Azure Compute environment. These metrics include CPU usage, network usage, and disk performance.

Before you begin

- Read the entire "Configuring Azure Compute monitoring" on page 201 topic to determine what is needed to complete the configuration.
- The directions here are for the most current release of the agent, except as indicated.
- Make sure that the system requirements for the Azure Compute agent are met in your environment. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Azure Compute agent.
- Ensure that the following information is available:
  - The Azure Subscription Credentials with permission to access the Azure Compute instances to monitor. See "Azure Compute Configuration Information" on page 202 for more details.

About this task

The Azure Compute agent is both a multiple instance agent and also a subnode agent. Each Azure Compute agent subnode monitors a grouping of Azure Compute virtual machines according to a filter you define. You can create one agent instance with multiple subnodes – one for each virtual machine grouping, or you can create an agent instance for each virtual machine grouping with one subnode for that grouping. Or you can create a combination of each type of configuration. After you configure agent instances, you must start each agent instance manually. It is suggested that you have no more than 50 resources per Azure Compute virtual machine grouping. Each Azure Compute agent subnode name must be unique within your environment.

Procedure

1. Configure the agent on Windows systems with the **IBM Performance Management** window or the silent response file.
   - "Configuring the agent on Windows systems" on page 203.
   - "Configuring the agent by using the silent response file" on page 207.
2. Configure the agent on Linux systems with the script that prompts for responses or the silent response file.
   - "Configuring the agent by responding to prompts" on page 205.
   - "Configuring the agent by using the silent response file" on page 207.

What to do next

In the Cloud APM console, go to your Application Performance Dashboard to view the data that was collected. For more information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you are unable to view the data in the agent dashboards, first check the server connection logs and then the data provider logs. The default paths to these logs are listed here:

- Linux `/opt/ibm/apm/agent/logs`
- Windows `C:\IBM\APM\TMAITM6_x64\logs`

For help with troubleshooting, see the Cloud Application Performance Management Forum.

Azure Compute Configuration Information

The Azure Compute agent requires the some additional setup in the Azure Compute environment.

About this task

To run these steps, you must login to the Microsoft Azure console.

Procedure

1. Subscription ID
   - On the left pane, select "Subscriptions" and chose the subscription you want to use for this agent.
   - Select "Overview", then copy the Subscription ID. This will be used as one of the Agent's configuration parameters.
2. Tenant ID
   - Navigate to "Azure Active Directory".
   - Select "Properties", then copy the Tenant ID.
3. Register an Application
   - Go to "All services" and search for "App registrations".
   - Click "New Application Registration".
   - Fill out a name, select Application Type "Web App/API", and a sign-on URL (this URL will not be used so chose whatever you'd like).
   - Click "Create"

- Copy the Application ID - This will be used in the Agent's "Client ID" field.

4. Create Application Key
   - Click on the App you just created, then go to "Settings" followed by "Keys".
   - Enter a description (e.g., "IBM Key") and duration (e.g., "Never Expires"), then click "Save".
   - Copy the Secret Key and store it somewhere safe - you will only see this key one time and will need to generate a new one if you lose it.
5. Give the Application Permissions
   - Go to "Subscriptions" and select the subscription to be monitored.
   - Go to "Access Control (IAM) and click "Add".
   - Select "Reader" role or higher for monitoring.
   - Under "Select", find the App you just registered and select it, then click "Save".

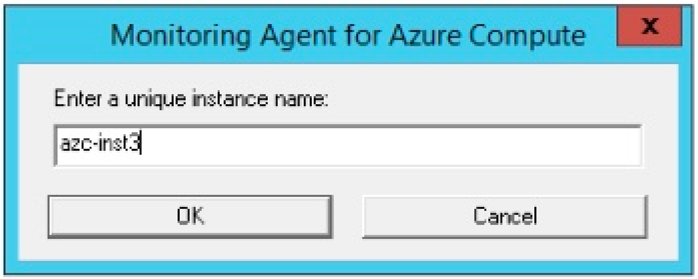
Configuring the agent on Windows systems

You can configure the Azure Compute agent on Windows operating systems by using the IBM Cloud Application Performance Management window. After you update the configuration values, you must start the agent to save the updated values.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Cloud Application Performance Management**.
2. In the **IBM Performance Management** window, right-click the **Monitoring Agent for Azure Compute** template, and then click **Configure agent**.

   Remember: After you configure an agent instance for the first time, the **Configure agent** option is disabled. To configure the agent instance again, right-click on it and then click **Reconfigure**.
3. Enter a unique instance name then click **OK**. Use only Latin letters, Arabic numerals, and the hyphen-minus character in the instance name. Example, `azc-inst3`. For more information, see *instancename* under "Common MSN format for multi-instance agents" on page 168.

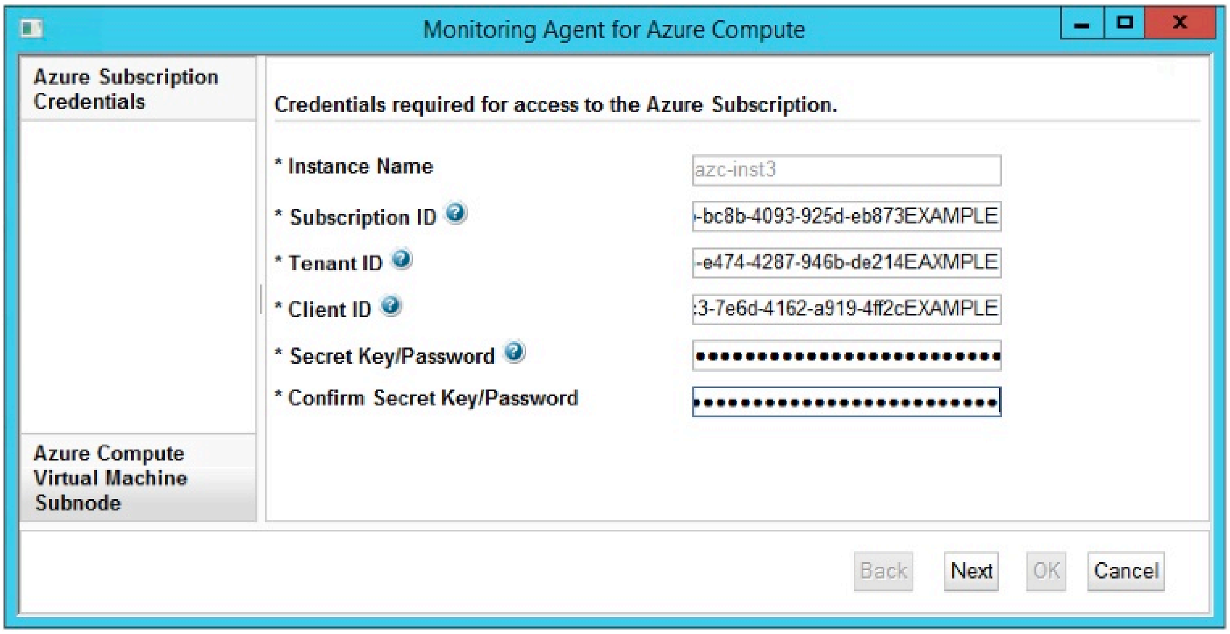
   Monitoring Agent for Azure Compute

   Enter a unique instance name: azc-inst3

   Figure 9. The window to enter a unique agent instance name.
4. Enter the **Azure Subscription Credentials**, then click **Next**.

   See Table 16 on page 209 for an explanation of each of the configuration parameters.

   Important: Windows If your **`Secret Key/Password`** contains an equal sign (=), you must reenter it each time that you reconfigure the agent.


Monitoring Agent for Azure Compute

Azure Subscription Credentials

Azure Compute Virtual Machine Subnode

Credentials required for access to the Azure Subscription.

* Instance Name: azc-inst3

* Subscription ID: -bc8b-4093-925d-eb873EXAMPLE

* Tenant ID: -e474-4287-946b-de214EAXMPLE

* Client ID: :3-7e6d-4162-a919-4ff2cEXAMPLE

* Secret Key/Password: •••••••••••••••••••••••••

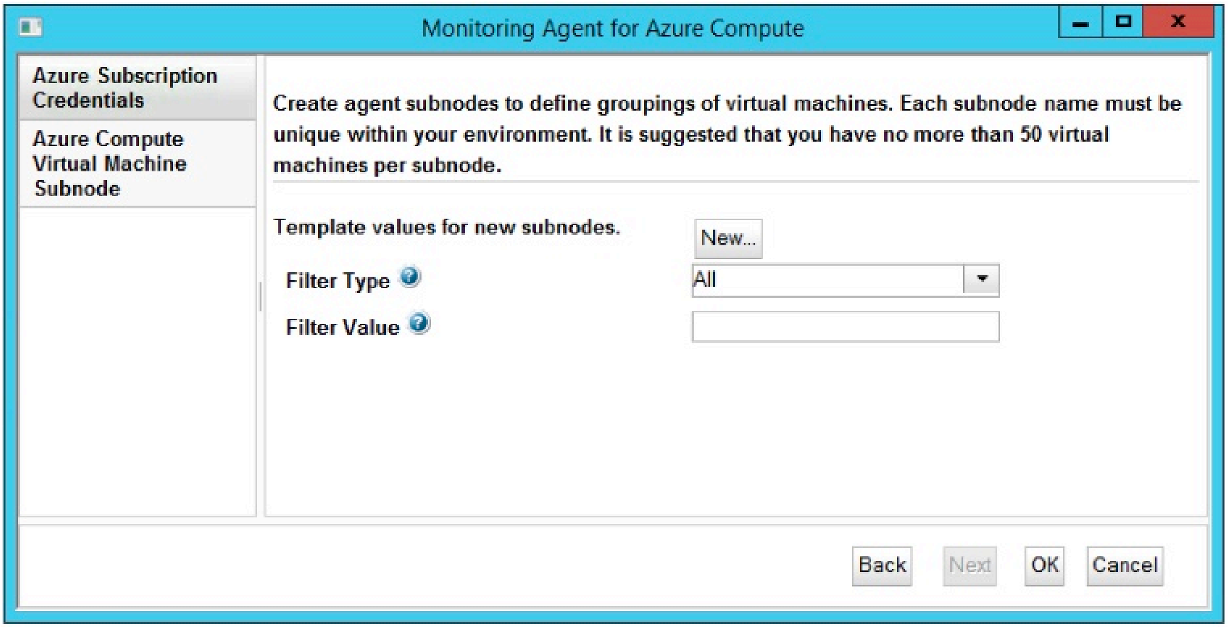
* Confirm Secret Key/Password: •••••••••••••••••••••••••

Figure 10. The Azure subscription credentials window.

5. Enter the **Azure Compute Virtual Machine Subnode** template settings.

   See Table 17 on page 210 for an explanation of each of the configuration parameters.

   Note: This section is not the Azure Compute Virtual Machine Subnode instance configuration. It is a template section for setting what is used as the default values when you add the actual Azure Compute Virtual Machine Subnode instance configurations in step 6.


Monitoring Agent for Azure Compute

Azure Subscription Credentials

Azure Compute Virtual Machine Subnode

Create agent subnodes to define groupings of virtual machines. Each subnode name must be unique within your environment. It is suggested that you have no more than 50 virtual machines per subnode.

Template values for new subnodes.

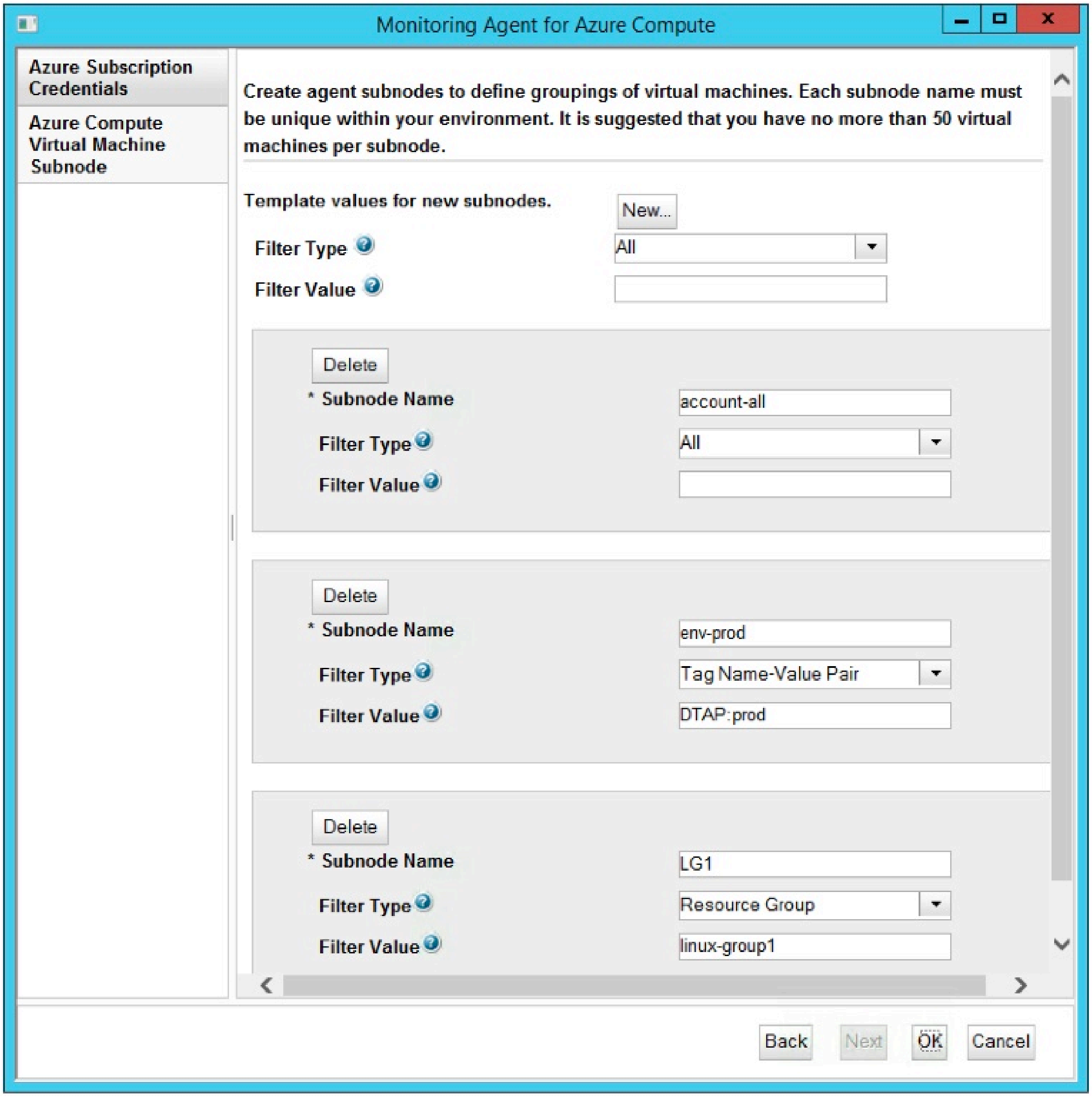
Filter Type: All

Filter Value:

Figure 11. The window to specify Azure Compute virtual machine subnode template settings.

6. Press **New** and enter **Azure Compute Virtual Machine Subnode** instance settings, then click **Next**.

   See Table 17 on page 210 for an explanation of each of the configuration parameters.


Monitoring Agent for Azure Compute
Azure Subscription Credentials
Azure Compute Virtual Machine Subnode
New...
Delete
Back
Next
OK
Cancel

Create agent subnodes to define groupings of virtual machines. Each subnode name must be unique within your environment. It is suggested that you have no more than 50 virtual machines per subnode.

Template values for new subnodes.

Filter Type: All

Filter Value:

* Subnode Name: account-all

Filter Type: All

Filter Value:

* Subnode Name: env-prod

Filter Type: Tag Name-Value Pair

Filter Value: DTAP:prod

* Subnode Name: LG1

Filter Type: Resource Group

Filter Value: linux-group1

Figure 12. The window to specify Azure Compute virtual machine subnode instance settings.

7. Click **OK** to complete the configuration.
8. In the IBM Cloud Application Performance Management window, right-click the instance that you configured, and then click **Start**.

Configuring the agent by responding to prompts

After installation of the Azure Compute agent, you must configure it before you start the agent. If the Azure Compute agent is installed on a local Linux computer, you can follow these instructions to configure it interactively through command line prompts.

About this task

Remember: If you are reconfiguring a configured agent instance, the value that is set in the last configuration is displayed for each setting. If you want to clear an existing value, press the space key when the setting is displayed.

Procedure

Follow these steps to configure the Azure Compute agent by running a script and responding to prompts.

1. Run the following command:

   ```
   install_dir/bin/azure_compute-agent.sh config instance-name
   ```

   Where *install_dir* is the path where the agent is installed and *instance-name* is the name that you want to give to the agent instance. Use only Latin letters, Arabic numerals, and the hyphen-minus character in the *instance-name*. For more information, see *instancename* under "Common MSN format for multi-instance agents" on page 168.

   Example

   ```
   /opt/ibm/apm/agent/bin/azure_compute-agent.sh config azc-inst3
   ```

2. Respond to the prompts to set configuration values for the agent.

   See "Configuration parameters for the Azure Compute agent" on page 209 for an explanation of each of the configuration parameters.

   Remember: When you first configure an agent instance, you must add at least one subnode when prompted to **Edit 'Azure Compute Virtual Machine Subnode' settings**.

3. Run the following command to start the agent:

   ```
   install_dir/bin/azure_compute-agent.sh start instance-name
   ```

   Where *install_dir* is the path where the agent is installed and *instance-name* is the name of the agent instance.

   Example

   ```
   /opt/ibm/apm/agent/bin/azure_compute-agent.sh start azc-inst3
   ```

Example

Creating an agent instance that is named `azc-inst3` and has one subnode instance that is named `azc1`.

```
# ./azure_compute-agent.sh config azc-inst3
Configuring Monitoring Agent for Azure Compute

Edit 'Monitoring Agent for Azure Compute' settings? [ 1=Yes, 2=No ]  (default is: 1): 1

Azure Subscription Credentials :
Credentials required for access to the Azure Subscription.

The ID assigned by Azure for the Subscription that is monitored.
Subscription ID (default is: ): 09x73b6b-bcxb-40x3-92xd-ebx7-EXAMPLE

The tenant ID that is assigned by Azure. Used to log in to the Azure service API.
Tenant ID (default is: ): 75x2e745-e4x4-42x7-94xb-dex1-EXAMPLE

The client ID that is assigned by Azure to identify this agent as an external
application that monitors the Azure compute services.
Client ID (default is: ): 79x2e6c3-7exd-41x2-a9x9-4fx2-EXAMPLE

The secret access key or password that is created by Azure for the client application.
Enter Secret Key/Password (default is: ): hidden

Re-type : Secret Key/Password (default is: ): hidden

Azure Compute Virtual Machine Subnode :

Create agent subnodes to define groupings of virtual machines. Each subnode name
must be unique within your environment. It is suggested that you have no more
than 50 virtual machines per subnode.

No 'Azure Compute Virtual Machine Subnode' settings available.
Edit 'Azure Compute Virtual Machine Subnode' settings, [1=Add, 2=Edit, 3=Del,
4=Next, 5=Exit] (default is: 5): 1
Subnode Name (default is: ): azc1
```

```
The type of filter to be applied.
Filter Type [ 1=All, 2=Tag Name-Value Pair, 3=Resource Group ] (default is: 1): 2

The filter value corresponding to the selected Filter Type. This value can be a
Resource Group or Tag Name-Value Pair, for example Environment\:Production.
A backslash might appear in the example, do not enter a backslash in the value
you provide.
Filter Value (default is: ): Environment:Production



Azure Compute Virtual Machine Subnode settings: Subnode Name=azc1
Edit 'Azure Compute Virtual Machine Subnode' settings, [1=Add, 2=Edit, 3=Del,
4=Next, 5=Exit] (default is: 5): 5
Configuration completed successfully.
Automatic start at system initialization has been configured.
Automatic stop at system shutdown has been configured.
You have new mail in /var/spool/mail/root
```

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

The silent response file contains the agent configuration parameters with default values that are defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

Follow these steps to configure the Azure Compute agent in the silent mode.

1. Open the `azure_compute_silent_config.txt` file at one of the following paths in a text editor.

   - Linux *`install_dir`*`/samples/azure_compute_silent_config.txt`

     Example, `/opt/ibm/apm/agent/samples/azure_compute_silent_config.txt`

   - Windows *`install_dir`*`\samples\azure_compute_silent_config.txt`

     Example, `C:\IBM\APM\samples\azure_compute_silent_config.txt`

   Where *install_dir* is the path where the agent is installed.

2. In the `azure_compute_silent_config.txt` file, specify values for all mandatory parameters and modify the default values of other parameters as needed.

   See "Configuration parameters for the Azure Compute agent" on page 209 for an explanation of each of the configuration parameters.

   Important: You must enable and specify the Filter Type and Filter Value parameters for at least one subnode name.

3. Save and close the `azure_compute_silent_config.txt` file, and run the following command:

   - Linux *`install_dir`*`/bin/azure_compute-agent.sh config` **`instance-name`** *`install_dir`*`/samples/azure_compute_silent_config.txt`

     Example, **`/opt/ibm/apm/agent/bin/azure_compute-agent.sh config azc-inst3 /opt/ibm/apm/agent/samples/azure_compute_silent_config.txt`**

   - Windows *`install_dir`*`\bin\azure_compute-agent.bat config` **`instance-name`** *`install_dir`*`\samples\azure_compute_silent_config.txt`

Example, **C:\IBM\APM\bin\azure_compute-agent.bat config azc-inst3 C:\IBM\APM\samples\azure_compute_silent_config.txt**

Where *install_dir* is the path where the agent is installed and *instance-name* is the name that you want to give to the agent instance. Use only Latin letters, Arabic numerals, and the hyphen-minus character in the *instance-name*. For more information, see *instancename* under "Common MSN format for multi-instance agents" on page 168.

Important: Ensure that you include the absolute path to the silent response file. Otherwise, the agent data is not shown in the dashboards.

4. Run the following command to start the agent:

   - Linux `install_dir/bin/azure_compute-agent.sh start` **`instance-name`**

     Example, **/opt/ibm/apm/agent/bin/azure_compute-agent.sh start azc-inst3**

   - Windows `install_dir\bin\azure_compute-agent.bat start` **`instance-name`**

     Example, **C:\IBM\APM\bin\azure_compute-agent.bat start azc-inst3**

   Where *install_dir* is the path where the agent is installed and *instance-name* is the name of the agent instance.

Example

Edited `azure_compute_silent_config.txt` with three subnodes that are named `account-all`, `env-prod`, and LG1.

```
#
# This is a sample configuration response file for agent Azure Compute.
#
# It contains an entry for every configuration property.
# Entries for optional properties that have no default value are included in
# comments.
# Entries for subnode AVM are given a sample subnode instance name of avm1.
# Ensure that all uncommented properties have a value before configuring the
# agent.
#

# Subscription ID: The ID assigned by Azure for the Subscription that is
# monitored.
KAK_SUBSCRIPTION_ID=09x73b6b-bcxb-40x3-92xd-ebx7-EXAMPLE
# Tenant ID: The tenant ID that is assigned by Azure. Used to log in to the
# Azure service API.
KAK_TENANT_ID=75x2e745-e4x4-42x7-94xb-dex1-EXAMPLE
# Client ID: The client ID that is assigned by Azure to identify this agent
# as an external
# application that monitors the Azure compute services.
KAK_CLIENT_ID=79x2e6c3-7exd-41x2-a9x9-4fx2-EXAMPLE
# Secret Key/Password: The secret access key or password that is created by
# Azure for the client application.
KAK_SECRET_PASSWORD=hZxWPq/IOxlnvg/wdxLwTf2Fs3x2sWQV/sCE-EXAMPLE

# Filter Type: The type of filter to be applied.
# Valid values: ALL (All), TAG_NAME_VALUE (Tag Name-Value Pair),
# RESOURCE_GROUP (Resource Group)
#KAK_FILTER_TYPE.avm1=ALL
# Filter Value: The filter value corresponding to the selected Filter Type.
# This value can be a Resource Group or Tag Name-Value Pair, for example
# Environment:Production. A backslash might appear in the example, do not
# enter a backslash in the value you provide.
#KAK_FILTER_VALUE.avm1=

# Filter Type: The type of filter to be applied.
# Valid values: ALL (All), TAG_NAME_VALUE (Tag Name-Value Pair),
# RESOURCE_GROUP (Resource Group)
KAK_FILTER_TYPE.account-all=ALL
# Filter Value: The filter value corresponding to the selected Filter Type.
# This value can be a Resource Group or Tag Name-Value Pair, for example
# Environment:Production. A backslash might appear in the example, do not
# enter a backslash in the value you provide.
KAK_FILTER_VALUE.account-all=

# Filter Type: The type of filter to be applied.
# Valid values: ALL (All), TAG_NAME_VALUE (Tag Name-Value Pair),
# RESOURCE_GROUP (Resource Group)
KAK_FILTER_TYPE.env-prod=TAG_NAME_VALUE
# Filter Value: The filter value corresponding to the selected Filter Type.
# This value can be a Resource Group or Tag Name-Value Pair, for example
# Environment:Production. A backslash might appear in the example, do not
# enter a backslash in the value you provide.
KAK_FILTER_VALUE.env-prod=DTAP:prod

# Filter Type: The type of filter to be applied.
# Valid values: ALL (All), TAG_NAME_VALUE (Tag Name-Value Pair),
# RESOURCE_GROUP (Resource Group)
KAK_FILTER_TYPE.LG1=RESOURCE_GROUP
# Filter Value: The filter value corresponding to the selected Filter Type.
# This value can be a Resource Group or Tag Name-Value Pair, for example
# Environment:Production. A backslash might appear in the example, do not
# enter a backslash in the value you provide.
KAK_FILTER_VALUE.LG1=linux-group1
```

Configuration parameters for the Azure Compute agent

The configuration parameters for the Azure Compute agent are displayed in tables which group them according to sections.

1. Table 16 on page 209 - Credentials that are required for access to the Azure Subscription that contains the Azure Compute resources to monitor.
2. Table 17 on page 210 - Create agent subnodes to define groupings of virtual machines. Each subnode name must be unique within your environment. It is suggested that you have no more than 50 virtual machines per subnode.

Table 16. Azure Subscription Credentials

Parameter name	Description	Silent configuration file parameter name
Subscription ID	The ID assigned by Azure for the Subscription that is monitored.	**KAK_SUBSCRIPTION_ID**
Tenant ID	The tenant ID that is assigned by Azure. Used to log in to the Azure service API.	**KAK_TENANT_ID**
Client ID	The client ID that is assigned by Azure to identify this agent as an external application that monitors the Azure compute services.	**KAK_CLIENT_ID**
Secret Key/ Password	The secret access key or password that is created by Azure for the client application. **Important:** Windows If your **Secret Key/Password** contains an equal sign (=), you must reenter it each time that you reconfigure the agent.	**KAK_SECRET_PASSWORD**

Table 17. Azure Compute Virtual Machine Subnode		
Parameter name	**Description**	**Silent configuration file parameter name**
Subnode Name	Name of the Azure Compute Subnode for collection of data. Example, *azc1*. Subnode name must be unique in your environment. This alias is part of the managed system name (MSN) and it is used to visually identify the monitored environment in the Cloud APM console. **Note:** This alias can be anything that you choose to represent the Azure Compute subnode instance with the following restrictions. Letters from the Latin alphabet (a-z, A-Z), Arabic numerals (0-9), the hyphen-minus character (-), and the underscore character (_) can be used to create agent subnode instance names. The maximum length of an Azure Compute subnode name is 25 characters.	Each of the following parameters must use a period (.) followed by an agent **Subnode Name** as a suffix. The **Subnode Name** must be the same for each subnode parameter. New agent subnode instances must use a unique **Subnode Name** for its set of parameters. For example, one agent subnode instance can use .*azc1* and another agent subnode instance can use .*azc2* in place of .*subnode_name* in the parameter names that follow.
Filter Type	The type of filter to be applied.	**KAK_FILTER_TYPE.subnode_name** Valid values, **ALL** All **TAG_NAME_VALUE** Tag Name-Value Pair **RESOURCE_GROUP** Resource Group

Table 17. Azure Compute Virtual Machine Subnode (continued)

Parameter name	Description	Silent configuration file parameter name
Filter Value	The filter value corresponding to the selected **`Filter` Type**. This value can be a **Resource Group** or **Tag Name-Value Pair**. Leave it empty for **`Filter` Type All**. For command line configuration, a backslash might appear in the example displayed. Do not enter a backslash in the value you provide. Examples of filter type and filter value pairs: • Azure Compute Subnode to monitor all virtual machines. Leave the filter value empty. Filter value is not needed and it is ignored for filter type **All**. – Filter Type: **All** – Filter Value: • Azure Compute Subnode to monitor all virtual machines with a tag name of `DTAP` and a tag value that matches the string `prod`. – Filter Type: **Tag Name-Value Pair** – Filter Value: `DTAP:prod` • Azure Compute Subnode to monitor all virtual machines with a resource group property that matches the string `linux-group1`. – Filter Type: **Resource Group** – Filter Value: `linux-group1`	**`KAK_FILTER_VALUE.subnode_name`**

Configuring Cassandra monitoring

You must configure the Cassandra agent so that the agent can collect data from the nodes within the cluster to monitor the health of the Cassandra Database.

Before you begin

Review the hardware and software prerequisites, see Software Product Compatibility Reports for Cassandra agent

About this task

The Cassandra agent is a multiple instance agent. You must create the first instance and start the agent manually.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the "Change history" on page 54.

- To configure the agent on Windows systems, you can use the IBM Cloud Application Performance Management window or the silent response file.
- To configure the agent on Linux systems, you can run the script and respond to prompts, or use the silent response file.

Configuring the agent on Windows systems

You can use the IBM Cloud Application Performance Management window to configure the agent on Windows systems.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Template** in the **Task/SubSystem** column, and click **Configure Using Defaults**.

   The **Monitoring Agent for Cassandra** window opens.
3. In the **Enter a unique instance name** field, type an agent instance name and click **OK**.
4. In the **Monitoring Agent for Cassandra** window, specify values for the configuration parameters and click **OK**.

   For information about the configuration parameters, see “Configuration parameters of the agent” on page 213.
5. In the **IBM Performance Management** window, right-click the agent instance that you created and click **Start**.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the console, see “Starting the Cloud APM console” on page 957.

For help with troubleshooting, see the Troubleshooting section.

Configuring the agent on Linux systems

To configure the agent on Linux operating systems, you must run the script and respond to prompts.

Procedure

1. On the command line, change the path to the agent installation directory.
   Example: `/opt/ibm/apm/agent/bin`
2. Run the following command where instance_name is the name that you want to give to the instance:

   ```
   ./cassandra-agent.sh config instance_name
   ```
3. When the command line displays the following message, type 1 and enter:

   ```
   Edit ‘Monitoring Agent for Cassandra’ setting? [1=Yes, 2=No]
   ```
4. Specify values for the configuration parameters when you are prompted.

   For information about the configuration parameters, see “Configuration parameters of the agent” on page 213.
5. Run the following command to start the agent:

   ```
   ./cassandra-agent.sh start instance_name
   ```

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see “Starting the Cloud APM console” on page 957.

For help with troubleshooting, see the Troubleshooting section.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

You can use the silent response file to configure the Cassandra agent on Linux and Windows systems. After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

1. In a text editor, open the silent config file that is available at the following location and specify values for all the parameters:

   Windows `install_dir\samples\cassandra_silent_config_windows.txt`
   Linux `install_dir\samples\cassandra_silent_config_UNIX.txt`

   Windows `C:\IBM\APM\samples`

   Linux `/opt/ibm/apm/agent/samples`

   For information about the configuration parameters, see "Configuration parameters of the agent" on page 213.
2. On the command line, change the path to `install_dir\bin`.
3. Run the following command:

   Windows `cassandra-agent.bat config instance_name install_dir\samples\cassandra_silent_config_windows.txt`
   Linux `cassandra-agent.sh config instance_name install_dir\samples\cassandra_silent_config_UNIX.txt`
4. Start the agent.

   Windows In the **IBM Performance Management** window, right-click the agent instance that you created, and click **Start**.
   Linux Run the following command: `./cassandra-agent.sh start instance_name`

What to do next

Log in to the Cloud APM console to view data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

For help with troubleshooting, see the Troubleshooting section.

Configuration parameters of the agent

While configuring the Cassandra agent, you can change the default value of the parameters, such as IP Address and JMX_PORT.

The following table contains detailed descriptions of the configuration parameters of the Cassandra agent.

Table 18. Names and descriptions of the configuration parameters

Parameter name	Description	Mandatory field
Instance Name	The default value for this field is identical to the value that you specify in the **Enter a unique instance name** field.	Yes
IP Address	The IP address of any node in the cluster.	Yes

Table 18. Names and descriptions of the configuration parameters (continued)

Parameter name	Description	Mandatory field
JMX_PORT	The JMX Port number to enable monitoring. **Important:** Ensure that you specify the JMX Port, JMX Username, and JMX Password throughout the cluster. If the node through which the agent is connected to the cluster, is not working, then the agent can collect data though a different node in the cluster by using the same parameters.	Yes
JMX_Username	The user name for accessing JMX.	No
JMX_Password	The password for accessing the JMX.	No

Configuring Cisco UCS monitoring

The Monitoring Agent for Cisco UCS monitors the Cisco UCS Virtual Infrastructure by connecting to the Cisco UCSM. You must configure the Cisco UCS agent so that the agent can collect the Cisco UCS data.

Before you begin

- Review the hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Cisco UCS agent.
- Ensure that the user, who connects to the Cisco UCSM infrastructure, has aaa or administrator privileges. Use an existing user ID, which has aaa or administrator privileges, or create a new user ID.
- If the Cisco UCS agent is configured to communicate with its Cisco UCS data sources that use the SSL agent, add the SSL certificate of each data source to the certificate truststore of the agent. For more information about enabling SSL communication with Cisco UCS data sources, see “Enabling SSL communication with Cisco UCS data sources” on page 219.

About this task

The Cisco UCS agent is a multiple instance agent. You must create the first instance, and start the agent manually.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version. To access the documentation for earlier agent releases, see the following table:

Table 19. Agent versions and documentation

Cisco UCS agent version	Documentation
7.2.0.4, 7.2.0.3	IBM Cloud Application Performance Management **Note:** The link opens an on-premises Knowledge Center topic.
7.2.0.2	IBM Performance Management 8.1.3 **Note:** The link opens an on-premises Knowledge Center topic.
7.2.0.1	IBM Performance Management 8.1.2 **Note:** The link opens an on-premises Knowledge Center topic.

The configuration attributes define which Cisco UCS Infrastructure is monitored. The attributes define a connection to Cisco UCSM 1.4, or later. You can configure more than one instance of the monitoring agent on a remote monitoring host system. You can also create separate instances to monitor specific Cisco UCS Infrastructure.

After the Cisco UCS agent is installed, you can start the agent. However, you must manually configure the agent to view data for all the agent attributes.

- To configure the agent on Windows systems, you can use the **IBM Performance Management** window or the silent response file.
- To configure the agent on Linux systems, you can run the script and respond to prompts, or use the silent response file.

Configuring the agent on Windows systems

You can configure the agent on Windows operating systems by using the **IBM Performance Management** window. After you update the configuration values, you must start the agent to save the updated values.

About this task

The Cisco UCS agent provides default values for some parameters. You can specify different values for these parameters.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Monitoring Agent for Cisco UCS**, and then click **Configure agent**.

   Remember: After you configure the agent for the first time, the **Configure agent** option is disabled. To configure the agent again, click **Reconfigure**.
3. In the Monitoring Agent for Cisco UCS window, complete the following steps:
   a) Enter a unique name for the Cisco UCS agent instance, and click **OK**.
   b) On the **CONFIG** tab, specify values for the configuration parameters, and then click **Next**.
   c) On the **LOG_CONFIG** tab, specify values for the configuration parameters, and then click **Next**.

   For information about the configuration parameters in each tab of the Monitoring Agent for Cisco UCS window, see the following topics:

   - "Configuration parameters for the agent" on page 217
   - "Configuration parameters for the data provider" on page 218
4. In the **IBM Performance Management** window, right-click **Monitoring Agent for Cisco UCS**, and then click **Start**.

What to do next

- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

  If you need help with troubleshooting, see the Troubleshooting section.
- If you are monitoring a large Cisco UCS environment, you might need to increase the heap size for the Java™ data provider. For more information, see "Increasing the Java heap size" on page 220.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

The silent response file contains the agent configuration parameters with default values that are defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

- To configure the Cisco UCS agent in the silent mode, complete the following steps:

  a) In a text editor, open the `cisco_ucs_silent_config.txt` file that is available at the following path:

  - Linux *install_dir*`/samples/cisco_ucs_silent_config.txt`

    Example `/opt/ibm/apm/agent/samples/cisco_ucs_silent_config.txt`

  - Windows *install_dir*`\samples\cisco_ucs_silent_config.txt`

    Example `C:\IBM\APM\samples\cisco_ucs_silent_config.txt`

  b) In the `cisco_ucs_silent_config.txt` file, specify values for all mandatory parameters. You can also modify the default values of other parameters.

  For information about the configuration parameters, see the following topics:

  - "Configuration parameters for the agent" on page 217
  - "Configuration parameters for the data provider" on page 218

  c) Save and close the `cisco_ucs_silent_config.txt` file, and run the following command:

  - Linux *install_dir*`/bin/cisco_ucs-agent.sh config` *instance_name* *install_dir*`/samples/cisco_ucs_silent_config.txt`

    Example **`/opt/ibm/apm/agent/bin/cisco_ucs-agent.sh config instance_name /opt/ibm/apm/agent/samples/cisco_ucs_silent_config.txt`**

  - Windows *install_dir*`\bin\cisco_ucs-agent.bat config` *instance_name* *install_dir*`\samples\cisco_ucs_silent_config.txt`

    Example **`C:\IBM\APM\bin\cisco_ucs-agent.bat config instance_name C:\IBM\APM\samples\cisco_ucs_silent_config.txt`**

    Where

    instance_name
    Name that you want to give to the instance.

    install_dir
    Path where the agent is installed.

  Important: Ensure that you include the absolute path to the silent response file. Otherwise, the agent data is not shown in the dashboards.

  d) Run the following command to start the agent:

  - Linux *install_dir*`/bin/cisco_ucs-agent.sh start` *instance_name*

    Example **`/opt/ibm/apm/agent/bin/cisco_ucs-agent.sh start instance_name`**

- Windows `install_dir\bin\cisco_ucs-agent.bat start instance_name`

  Example **C:\IBM\APM\bin\cisco_ucs-agent.bat start instance_name**

What to do next

- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

  If you need help with troubleshooting, see the Troubleshooting section.
- If you are monitoring a large Cisco UCS environment, you might need to increase the heap size for the Java™ data provider. For more information, see "Increasing the Java heap size" on page 220.

Configuring the agent by responding to prompts

To configure the agent on Linux operating systems, you must run the script and respond to prompts.

Procedure

- To configure the agent by running the script and responding to prompts, complete the following steps:
  a) On the command line, enter the following command:

  ```
  install_dir/bin/cisco_ucs-agent.sh config instance_name
  ```

  Example **/opt/ibm/apm/agent/bin/cisco_ucs-agent.sh config instance_name**

  Where

  instance_name
  Name that you want to give to the instance.

  install_dir
  Path where the agent is installed.

  b) Respond to the prompts by referring to the following topics:
  - "Configuration parameters for the agent" on page 217
  - "Configuration parameters for the data provider" on page 218

  c) Run the following command to start the agent:

  ```
  install_dir/bin/cisco_ucs-agent.sh start instance_name
  ```

  Example **/opt/ibm/apm/agent/bin/cisco_ucs-agent.sh start instance_name**

What to do next

- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

  If you need help with troubleshooting, see the Troubleshooting section.
- If you are monitoring a large Cisco UCS environment, you might need to increase the heap size for the Java™ data provider. For more information, see "Increasing the Java heap size" on page 220.

Configuration parameters for the agent

When you configure the Cisco UCS agent, you can change the default values of the parameters, such as the instance name and the SSL validation certificates.

The following table contains detailed descriptions of the configuration parameters for the Cisco UCS agent.

Table 20. Names and descriptions of the configuration parameters for the Cisco UCS agent

Parameter name	Description	Mandatory field
Instance Name	The name of the instance. **Restriction:** The **Instance Name** field displays the name of the instance that you specify when you configure the agent for the first time. When you configure the agent again, you cannot change the instance name of the agent.	Yes
URL	The URL of the Cisco UCS Manager. To set the URL of the Cisco UCS manager, enter the URL in the http://ip_address/nuova format.	Yes
Username	The administrator user name of the Cisco UCS Manager.	Yes
Password	The administrator password of the Cisco UCS Manager.	Yes
Confirm Password	The same password that you entered in the **Password** field.	Yes
SSL truststore filepath	The path of the secure socket layer (SSL) truststore file. If you want the agent to validate SSL certificates when using SSL to communicate over the network, then specify the location where the secure socket layer (SSL) truststore file is located.	Yes
Validate SSL Certificates	A Boolean value that indicates whether the agent validates SSL certificates when the agent uses SSL to communicate over the network. Set the value to Yes if you want the agent to validate SSL certificates when the agent uses SSL to communicate over the network. Set the value to No to prevent the agent from validating SSL certificates. **Tip:** For more information about enabling SSL communication with Cisco UCS data sources, see "Enabling SSL communication with Cisco UCS data sources" on page 219.	Yes

Configuration parameters for the data provider

When you configure the Cisco UCS agent, you can change the default values of the parameters for the data provider, such as the maximum number of data provider log files, the maximum size of the log file, and the level of detail that is included in the log file.

The following table contains detailed descriptions of the configuration parameters for the data provider.

Table 21. Names and descriptions of the configuration parameters for the data provider

Parameter name	Description	Mandatory field
Maximum number of Data Provider Log Files	The maximum number of log files that the data provider creates before it overwrites the previous log files. The default value is 10.	Yes
Maximum Size in KB of Each Data Provider Log	The maximum size in KB that a data provider log file must reach before the data provider creates a new log file. The default value is 5190 KB.	Yes

Table 21. Names and descriptions of the configuration parameters for the data provider (continued)

Parameter name	Description	Mandatory field
Level of Detail in Data Provider Log	The level of detail that can be included in the log file that the data provider creates. The default value is INFO. The following values are valid: OFF, SEVERE, WARNING, INFO, FINE, FINER, FINEST, and ALL.	Yes

Enabling SSL communication with Cisco UCS data sources

The Cisco UCS agent can be configured to securely communicate with its Cisco UCS data sources by using SSL. In this configuration, you must add a data source SSL certificate to the certificate truststore of the agent.

About this task

Important: The following information applies only if the agent is configured to validate SSL certificates.

If SSL certificate validation is turned off, the Cisco UCS agent connects to Cisco UCS data sources even if their SSL certificates are expired, untrusted, or invalid. However, turning off SSL certificate validation is potentially not secure and must be done with care.

If a Cisco UCS data source uses an SSL certificate that is signed by a common Certificate Authority (for example, Verisign, Entrust, or Thawte), then it is not necessary to add certificates to the Cisco UCS agent certificate truststore. However, if the data source uses a certificate that is not signed by a common Certificate Authority, as is the case by default, the certificate must be added to the truststore to allow the agent to successfully connect and collect data.

Procedure

1. Copy the certificate file from your data source to the agent computer.
2. On the agent computer, place the certificate file in a directory of your choice. Do not overwrite the certificate files. Use a unique file name and label for each certificate that you add.
3. Use the `keytool` command to add the data source certificate to the certificate truststore of the agent:

```
keytool -import -noprompt -trustcacerts -alias CertificateAlias -file
CertificateFile -keystore Truststore -storepass TruststorePassword
```

Where

CertificateAlias

Unique reference for each certificate added to the certificate truststore of the agent, for example, an appropriate alias for the certificate from *datasource.example.com* is *datasource*.

CertificateFile

Complete path and file name to the Cisco UCS data source certificate to add to the truststore.

Truststore

Complete path and file name to the Cisco UCS agent certificate database. Use the following path and file name:

- Windows (64 bit) `*install_dir*\tmaitm6_x64\kv6.truststore`
- Linux (64 bit) `install_dir/lx8266/vm/etc/kv6.truststore`

TruststorePassword

ITMFORVE is the default password for the Cisco UCS agent truststore. To change this password, consult the Java Runtime documentation for information about the tools to use.

Important: To use the `keytool` command, the Java Runtime bin directory must be in your path. Use the following commands:

- Windows (64 bit) `set PATH=%PATH%;install_dir\java\java70_x64\jre\bin`
- Linux (64 bit) `PATH="$PATH":/opt/ibm/apm/agent/JRE/lx8266/bin`

4. After you add all the data source certificates, start the monitoring agent.

Increasing the Java heap size

After you configure the Cisco UCS agent, if you are monitoring a large Cisco UCS environment, then you might need to increase the heap size for the Java™ data provider.

About this task

The default heap size for the Java data provider is 256 megabytes. In large Cisco UCS environments, if the following problems arise, then you might need to increase the heap size:

- The Java data provider stops because of a `javacore` problem, and creates a file that is named `javacore.`*`date.time.number`*`.txt` in the `CANDLEHOME\tmaitm6_x64` directory.
- The `javacore.`*`date.time.number`*`.txt` file contains the string `java/lang/OutOfMemoryError`.

Procedure

- Windows

  Complete the following steps to set a value of 1 GB as heap size:

  1. Open the `%CANDLE_HOME%\TMAITM6_x64\kv6_data_provider.bat` file.
  2. Add the following line before the line that starts with `KV6_JVM_ARGS="$KV6_CUSTOM_JVM_ARGS...`:

     ```
     SET KV6_CUSTOM_JVM_ARGS=-Xmx1024m
     ```

  3. Restart the agent.

- Linux

  Complete the following steps to set a value of 1 GB as heap size:

  1. Open the `$CANDLEHOME/lx8266/vm/bin/kv6_data_provider.sh` file.
  2. Add the following line before the line that starts with `KV6_JVM_ARGS="$KV6_CUSTOM_JVM_ARGS...`:

     ```
     KV6_CUSTOM_JVM_ARGS=-Xmx1024m
     ```

  3. Restart the agent.

Configuring Citrix Virtual Desktop Infrastructure monitoring

The Citrix VDI agent provides a central point of monitoring for your Citrix XenDesktop or XenApp resources, including delivery groups, catalogs, applications, desktops, users, and sessions. Before the agent can be used, you must configure the agent to collect data through the delivery controller.

Before you begin

- The directions here are for the most current release of the agent, except as indicated.
- Make sure that the system requirements for the Citrix VDI agent are met in your environment. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Citrix VDI agent.
- Ensure that the following information is available:
  - Host name of the delivery controller to which you plan to connect.

  - OData user name, password, and domain.
  - PowerShell user name, password, domain, PowerShell port, SSL verification type, and authentication mechanism if you enable Windows Event Log Event and PowerShell metric retrieval.
- Ensure that an agent operator user account has at least Citrix read-only administrator privileges. See Enabling Citrix read-only administrator privileges.
- Starting with Citrix VDI agent version 8.1.3.1, the ability to retrieve Windows Event Log Events became available. To retrieve Windows Event Log Events from all Desktop Delivery Controller (DDC) and Virtual Delivery Agent (VDA) machines, remote PowerShell access needs to be enabled for the user account that is specified during the agent instance configuration. Follow these steps to ensure that the agent can perform this function:
  1. Log in to a Windows computer as the user specified in the agent instance configuration.
  2. Run the following PowerShell command, where *vda_system* is the name of a VDA machine that is powered on:
     ```
     Get-WinEvent -FilterHashtable
     @{ProviderName='Citrix*';LogName='Citrix*';StartTime=((Get-
     Date).AddDays(-10))} -ComputerName vda_system
     ```
- Ensure that the following load balancing policies are enabled for the monitored environment:
  - CPU Usage
  - Disk Usage
  - Memory Usage

  These policies can be configured through the Citrix Studio application.

About this task

The Citrix VDI agent is a multiple instance agent. You must create at least one instance, and start the agent instance manually.

The configuration for XenApp servers is the same as for XenDesktop servers. If a configuration parameter name or description mentions only "XenDesktop", it is also for XenApp.

Procedure

1. Configure the agent on Windows systems with the **IBM Performance Management** window or the silent response file.
   - “Configuring the agent on Windows systems” on page 222.
   - “Configuring the agent by using the silent response file” on page 226.
2. Configure the agent on Linux systems with the script that prompts for responses or the silent response file.
   - “Configuring the agent by responding to prompts” on page 225.
   - “Configuring the agent by using the silent response file” on page 226.

What to do next

In the Cloud APM console, go to your Application Performance Dashboard to view the data that was collected. For information about using the Cloud APM console, see “Starting the Cloud APM console” on page 957.

If you are unable to view the data in the agent dashboards, first check the server connection logs and then the data provider logs. The default paths to these logs are listed here:

- Linux `/opt/ibm/apm/agent/logs`
- Windows `C:\IBM\APM\TMAITM6_x64\logs`

For help with troubleshooting, see the Cloud Application Performance Management Forum.

Enabling Citrix read-only administrator privileges

The Citrix VDI agent requires the agent operator user account have at least Citrix read-only administrator privileges.

About this task

To run these steps remotely from a computer that has the Citrix Delegated Admin PowerShell Snap-in installed, use the `AdminAddress` parameter. For example, the command in step 2 would become `New-AdminAdministrator -Name "`*`YOURDOMAIN`*`\`*`NewAdmin`*`" -AdminAddress "`*`controller1.YOURDOMAIN.com`*`"`. Where *YOURDOMAIN* is the name of the network domain, *NewAdmin* is the user account that is being given Citrix administration privileges, and *controller1.YOURDOMAIN.com* is the fully qualified domain name of the Citrix site server.

Procedure

1. Start a PowerShell session with an existing Citrix administrator account.
2. Load the Delegated Admin PowerShell Snap-in to manage the Citrix XenApp or XenDesktop site.

   `(Add-PSSnapin Citrix.DelegatedAdmin.Admin.V1)`
3. Add the agent operator user account as a Citrix site administrator.

   `New-AdminAdministrator -Name "`*`YOURDOMAIN`*`\`*`NewAdmin`*`"`

   Where *YOURDOMAIN* is the name of the network domain and *NewAdmin* is the user account that is being given Citrix administration privileges.
4. Query for the available roles and scopes to assign to *NewAdmin*.

   ```
   Get-AdminRole
   Get-AdminScope
   ```
5. Assign roles to the agent operator user account, including read-only administrator permissions.

   `Add-AdminRight -Administrator "`*`YOURDOMAIN`*`\`*`NewAdmin`*`" -Role "`*`Read Only Administrator`*`" -Scope "`*`All`*`"`

   Where

   - *YOURDOMAIN* is the name of the network domain.
   - *NewAdmin* is the user account that is being given Citrix administration privileges.
   - *Read Only Administrator* is the Citrix site administrator role that you are assigning.
   - *All* is the Citrix site administrator scope that you are assigning.
6. Confirm the addition and changes of the administrator.

   `Get-AdminAdministrator -Name "`*`YOURDOMAIN`*`\`*`NewAdmin`*`"`

   Where *YOURDOMAIN* is the name of the network domain and *NewAdmin* is the user account that is being given Citrix administration privileges.

Configuring the agent on Windows systems

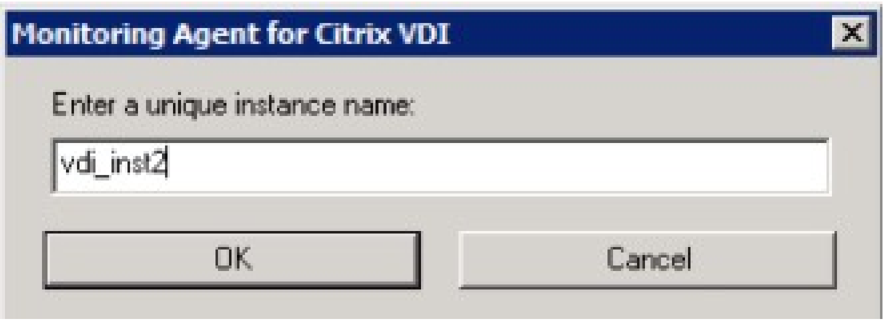
You can configure the Citrix VDI agent on Windows operating systems by using the IBM Cloud Application Performance Management window. After you update the configuration values, you must start the agent to save the updated values.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Cloud Application Performance Management**.
2. In the **IBM Performance Management** window, right-click the **Monitoring Agent for Citrix Virtual Desktop Infrastructure** template, and then click **Configure agent**.

Remember: After you configure an agent instance for the first time, the **Configure agent** option is disabled. To configure the agent instance again, right-click on it and then click **Reconfigure**.

3. Enter a unique instance name then click **OK**. Use only letters, Arabic numerals, the underline character, and the minus character in the instance name. Example, `vdi_inst2`.



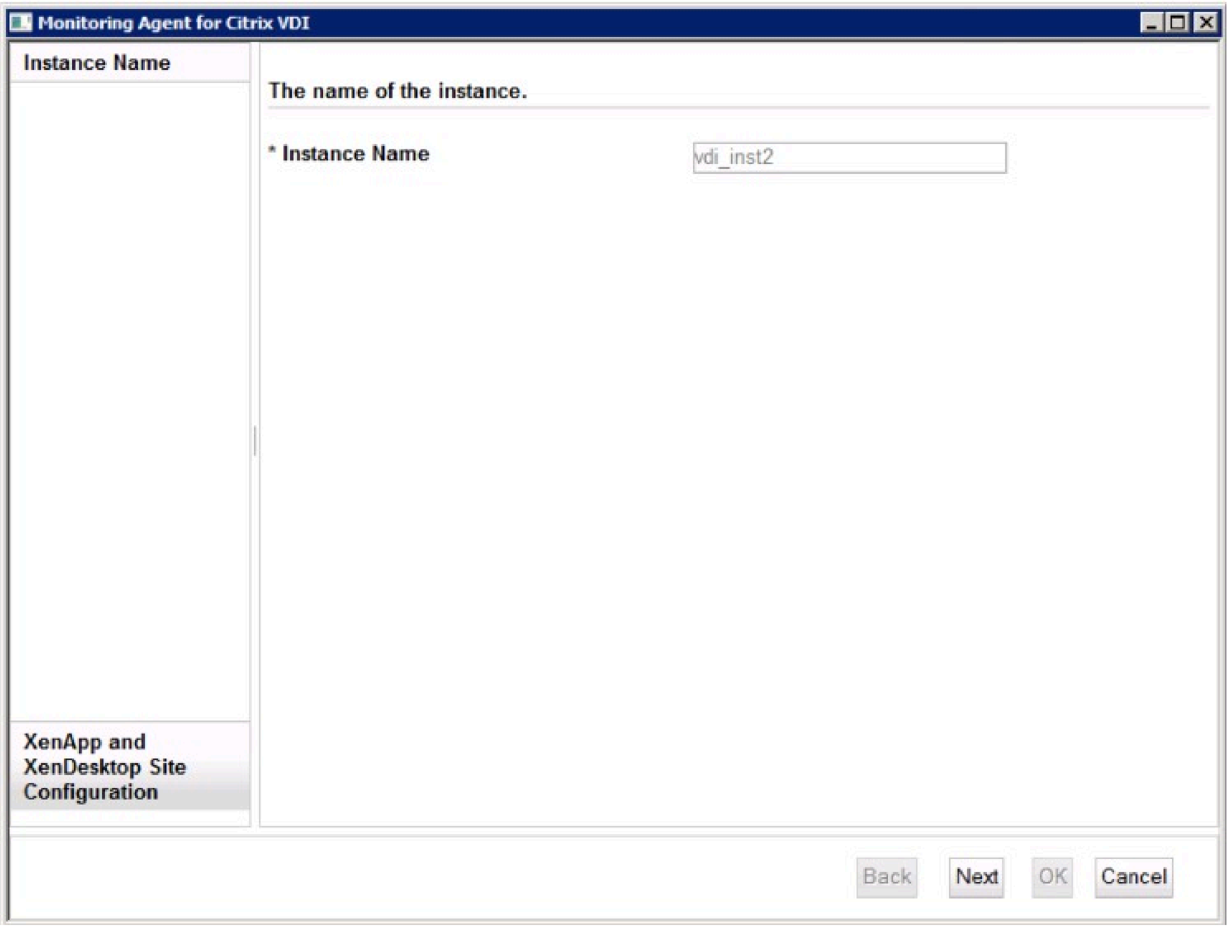
Monitoring Agent for Citrix VDI

Enter a unique instance name:

vdi_inst2

Figure 13. The window to enter a unique instance name.

4. Click **Next** on the agent instance name window.



Monitoring Agent for Citrix VDI

The name of the instance.

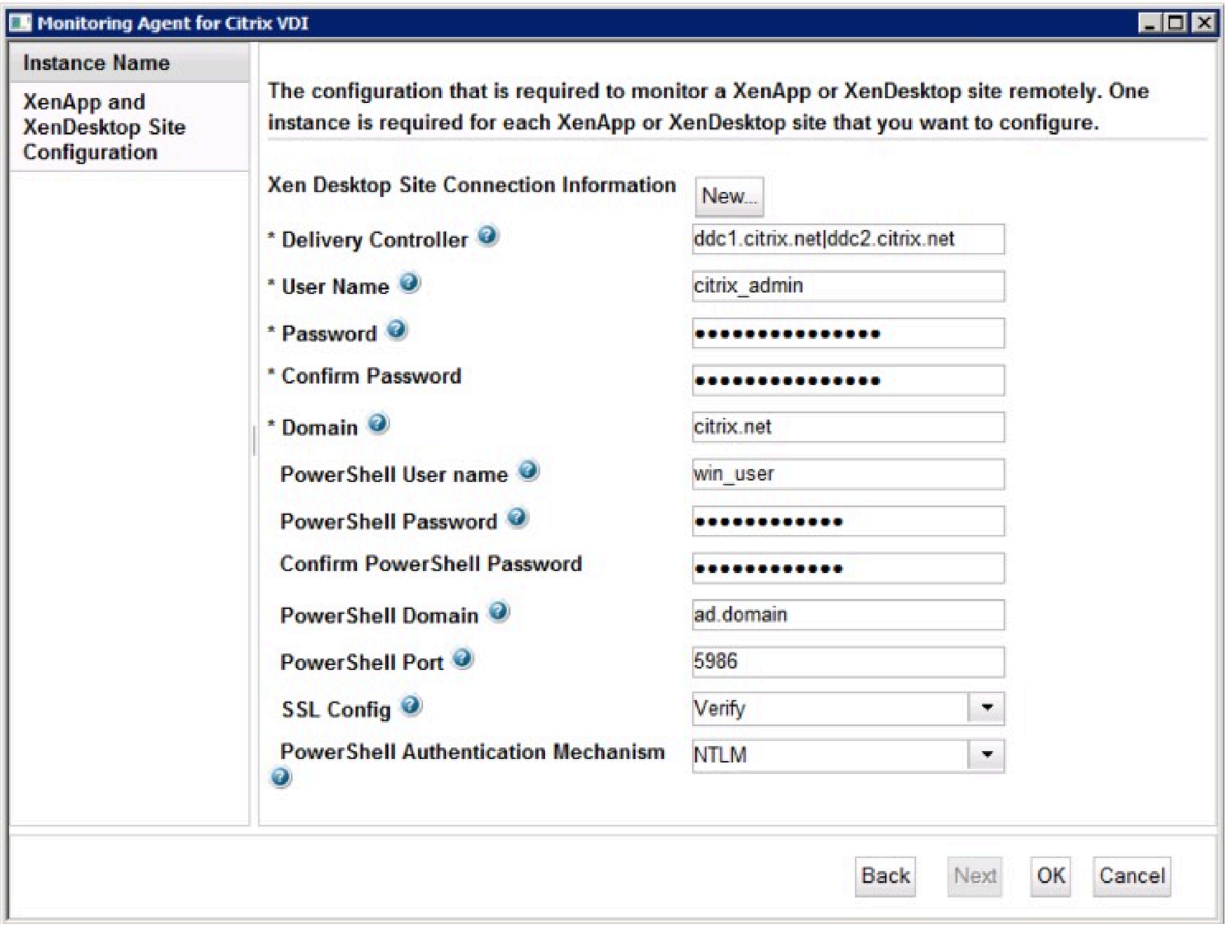
* Instance Name vdi_inst2

Figure 14. The agent instance name window.

5. Enter the **XenApp and XenDesktop Site Configuration** instance template settings.

Note: This section is not the XenApp or XenDesktop site instance configuration. It is a template section for setting what is used as the default values when you add the actual XenApp or XenDesktop site instance configurations in step 6.

See Table 22 on page 227 for an explanation of each of the configuration parameters.



Monitoring Agent for Citrix VDI
Instance Name
XenApp and XenDesktop Site Configuration
New...
Back
Next
OK
Cancel

The configuration that is required to monitor a XenApp or XenDesktop site remotely. One instance is required for each XenApp or XenDesktop site that you want to configure.

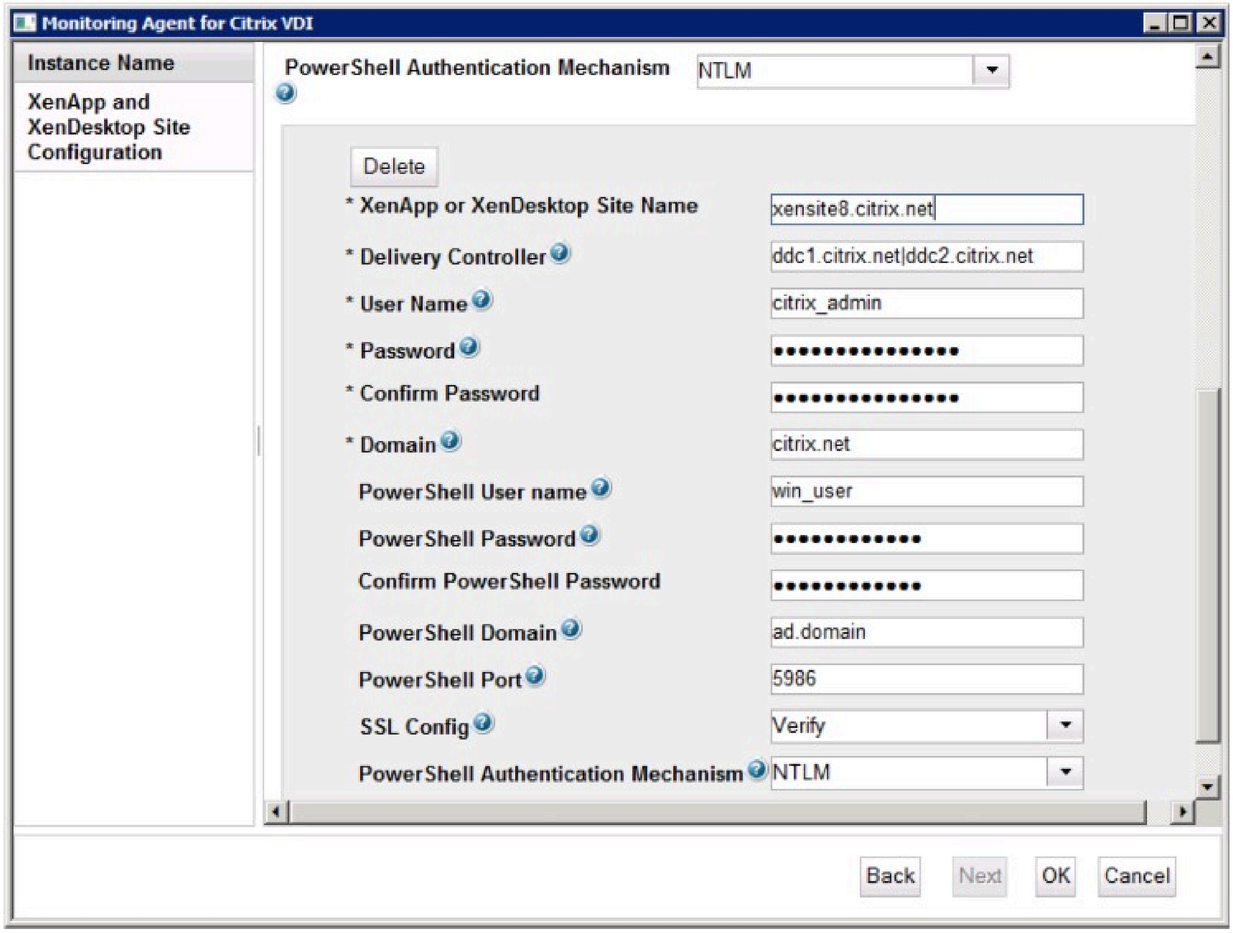
Xen Desktop Site Connection Information

* Delivery Controller	ddc1.citrix.net|ddc2.citrix.net
* User Name	citrix_admin
* Password	••••••••••••••••
* Confirm Password	••••••••••••••••
* Domain	citrix.net
PowerShell User name	win_user
PowerShell Password	•••••••••••••
Confirm PowerShell Password	•••••••••••••
PowerShell Domain	ad.domain
PowerShell Port	5986
SSL Config	Verify
PowerShell Authentication Mechanism	NTLM

Figure 15. The window to specify XenApp or XenDesktop site instance template settings.

6. Press **New** and enter XenApp or XenDesktop site instance settings, then click **Next**.

   See Table 22 on page 227 for an explanation of each of the configuration parameters.


Monitoring Agent for Citrix VDI
Instance Name
XenApp and XenDesktop Site Configuration
Delete
Back
Next
OK
Cancel

PowerShell Authentication Mechanism: NTLM

* XenApp or XenDesktop Site Name: xensite8.citrix.net

* Delivery Controller: ddc1.citrix.net|ddc2.citrix.net

* User Name: citrix_admin

* Password: •••••••••••••••

* Confirm Password: •••••••••••••••

* Domain: citrix.net

PowerShell User name: win_user

PowerShell Password: ••••••••••••

Confirm PowerShell Password: ••••••••••••

PowerShell Domain: ad.domain

PowerShell Port: 5986

SSL Config: Verify

PowerShell Authentication Mechanism: NTLM

Figure 16. The window to specify XenApp or XenDesktop site instance settings.

Note: The **`PowerShell User name`** parameter and all following PowerShell parameters are only needed when "Enabling monitoring of Windows events and PowerShell metrics" on page 229. These advanced environment variables are off by default because of the significant load they put on the monitored system.

Note: Ensure the **`SSL Config`** and **`PowerShell Authentication Mechanism`** parameters are set correctly for each new XenApp or XenDesktop site instance. A defect causes the default values to be set instead of the template values.

7. Click **OK** to complete the configuration.
8. In the IBM Cloud Application Performance Management window, right-click the instance that you configured, and then click **Start**.

Configuring the agent by responding to prompts

After installation of the Citrix VDI agent, you must configure it before you start the agent. If the Citrix VDI agent is installed on a local Linux machine, you can follow these instructions to configure it interactively through command line prompts.

About this task

Remember: If you are reconfiguring a configured agent instance, the value that is set in the last configuration is displayed for each setting. If you want to clear an existing value, press the space key when the setting is displayed.

Procedure

Follow these steps to configure the Citrix VDI agent by running a script and responding to prompts.

1. Run the following command:

   ```
   install_dir/bin/citrixvdi-agent.sh config instance_name
   ```

   where *install_dir* is the path where the agent is installed and *instance_name* is the name that you want to give to the agent instance.

   Example

   ```
   /opt/ibm/apm/agent/bin/citrixvdi-agent.sh config vdi_inst01
   ```

2. Respond to the prompts to set configuration values for the agent.

   See “Configuration parameters for the Citrix VDI agent” on page 227 for an explanation of each of the configuration parameters.

   Note: The **`PowerShell User name`** parameter and all following PowerShell parameters are only needed when “Enabling monitoring of Windows events and PowerShell metrics” on page 229. These advanced environment variables are off by default because of the significant load they put on the monitored system.

3. Run the following command to start the agent:

   ```
   install_dir/bin/citrixvdi-agent.sh start instance_name
   ```

   where *install_dir* is the path where the agent is installed and *instance_name* is the name of the agent instance.

   Example

   ```
   /opt/ibm/apm/agent/bin/citrixvdi-agent.sh start vdi_inst01
   ```

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

The silent response file contains the agent configuration parameters with default values that are defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

- Configure the Citrix VDI agent in the silent mode:

  a) Open the `citrixvdi_silent_config.txt` file at one of the following paths in a text editor.

     – Linux *`install_dir`*`/samples/citrixvdi_silent_config.txt`

       Example, `/opt/ibm/apm/agent/samples/citrixvdi_silent_config.txt`

     – Windows *`install_dir`*`\samples\citrixvdi_silent_config.txt`

       Example, `C:\IBM\APM\samples\citrixvdi_silent_config.txt`

     where *install_dir* is the path where the agent is installed.

  b) In the `citrixvdi_silent_config.txt` file, specify values for all mandatory parameters and modify the default values of other parameters as needed.

See “Configuration parameters for the Citrix VDI agent” on page 227 for an explanation of each of the configuration parameters.

Note: The **PowerShell User name** parameter and all following PowerShell parameters are only needed when “Enabling monitoring of Windows events and PowerShell metrics” on page 229. These advanced environment variables are off by default because of the significant load they put on the monitored system.

c) Save and close the `citrixvdi_silent_config.txt` file, and run the following command:

- Linux *install_dir*/bin/citrixvdi-agent.sh config **instance_name** *install_dir*/samples/citrixvdi_silent_config.txt

  Example, **/opt/ibm/apm/agent/bin/citrixvdi-agent.sh config vdi_inst01 /opt/ibm/apm/agent/samples/citrixvdi_silent_config.txt**

- Windows *install_dir*\bin\citrixvdi-agent.bat config **instance_name** *install_dir*\samples\citrixvdi_silent_config.txt

  Example, **C:\IBM\APM\bin\citrixvdi-agent.bat config vdi_inst01 C:\IBM\APM\samples\citrixvdi_silent_config.txt**

where *install_dir* is the path where the agent is installed and *instance_name* is the name that you want to give to the agent instance.

Important: Ensure that you include the absolute path to the silent response file. Otherwise, the agent data is not shown in the dashboards.

d) Run the following command to start the agent:

- Linux *install_dir*/bin/citrixvdi-agent.sh start **instance_name**

  Example, **/opt/ibm/apm/agent/bin/citrixvdi-agent.sh start vdi_inst01**

- Windows *install_dir*\bin\citrixvdi-agent.bat start **instance_name**

  Example, **C:\IBM\APM\bin\citrixvdi-agent.bat start vdi_inst01**

where *install_dir* is the path where the agent is installed and *instance_name* is the name of the agent instance.

Configuration parameters for the Citrix VDI agent

The configuration parameters for the Citrix VDI agent are displayed in a table.

1. Citrix VDI Agent Settings - Citrix VDI agent environment settings.

Table 22. Citrix VDI Agent Settings

Parameter name	Description	Silent configuration file parameter name
XenApp or XenDesktop Site Name	Provide a name to identify the XenApp or XenDesktop site agent instance. Example, *vdi_inst2* **Note:** This alias can be anything that you choose to represent the WebLogic server agent instance with the following restrictions. Only letters, Arabic numerals, the underline character, and the minus character can be used in the connection name. The maximum length of a connection name is 25 characters.	Each of the following parameters must have an instance name suffix that is the same for each parameter of an agent instance. New agent instances must use a unique instance name for its set of parameters. For example, one agent instance can use *.vdi1* and another agent instance can use *.vdi2* in place of *.instance_name* in the parameter names that follow.

Table 22. Citrix VDI Agent Settings (continued)

Parameter name	Description	Silent configuration file parameter name
Delivery Controller	Host name or IP address of the delivery controller. If multiple DDCs are set up in a cluster, a '|' separated list of delivery controllers can be provided.	**KVD_XDS_DELIVERY_CONTROLLER.instance_name**
User Name	User name that is used to authenticate with the OData API on the specified XenApp or XenDesktop delivery controller.	**KVD_XDS_ODATA_USERNAME.instance_name**
Password	Password that is used to authenticate with the OData API on the specified XenApp or XenDesktop delivery controller.	**KVD_XDS_ODATA_PASSWORD.instance_name**
Domain	Domain that is used to authenticate with the OData API on the specified XenApp or XenDesktop delivery controller.	**KVD_XDS_ODATA_DOMAIN.instance_name**
PowerShell User name	User name that is used to authenticate for PowerShell calls to remote VDA and DDC machines. **Note:** This and all following PowerShell parameters are only needed when "Enabling monitoring of Windows events and PowerShell metrics" on page 229. These advanced environment variables are off by default because of the significant load they put on the monitored system.	**KVD_XDS_POWERSHELL_USERNAME.instance_name**
PowerShell Password	Password that is associated with PowerShell user name provided.	**KVD_XDS_POWERSHELL_PASSWORD.instance_name**
PowerShell Domain	Domain that is associated with PowerShell user name provided.	**KVD_XDS_POWERSHELL_DOMAIN.instance_name**
PowerShell Port	The SSL port that is open for use by WinRm. PowerShell default remote connection ports are 5985 for HTTP and 5986 for HTTPS.	**KVD_XDS_POWERSHELL_PORT.instance_name**
SSL Requirement	Choose the SSL option required for your environment.	**KVD_XDS_SSL_CONFIG.instance_name** Valid values, **KVD_XDS_SSL_CONFIG_VERIFY** Verify **KVD_XDS_SSL_CONFIG_NOVERIFY** No Verify **KVD_XDS_SSL_CONFIG_NOSSL** No SSL

Table 22. Citrix VDI Agent Settings (continued)

Parameter name	Description	Silent configuration file parameter name
PowerShell Authentication Mechanism	Defines the type of authentication that is used to create a credential to retrieve information from remote systems with PowerShell.	**KVD_XDS_POWERSHELL_AUTH_MECH.instance_name** Valid values, **KVD_XDS_POWERSHELL_BASIC** Basic **KVD_XDS_POWERSHELL_CREDSSP** CredSSP **KVD_XDS_POWERSHELL_NTLM** NTLM **KVD_XDS_POWERSHELL_DEFAULT** Default **KVD_XDS_POWERSHELL_DIGEST** Digest **KVD_XDS_POWERSHELL_KERBEROS** Kerberos **KVD_XDS_POWERSHELL_NEGOTIATE** Negotiate

Enabling monitoring of Windows events and PowerShell metrics

Enable monitoring of Windows events and PowerShell metrics with this procedure. Monitoring this data can have a significant performance impact to the monitored system.

Before you begin

Ensure the agent's PowerShell configuration parameters are set.

About this task

One or more of the following advanced environment variables must be enabled for the agent to monitor Windows events and PowerShell metrics.

GET_SESSION_LATENCY
Whether session latency and round-trip time are retrieved remotely from the connected VDA from PowerShell.

GET_VDA_MACHINE_METRICS_REMOTELY
Whether VDA machine metrics are retrieved remotely from PowerShell.

RETRIEVE_WINDOWS_EVENTS
Whether Windows Event Log Events are retrieved from PowerShell from Windows VDAs and DDCs.

Procedure

1. Go to the agent installation directory of the Citrix VDI agent:

   - Linux *install_dir*/config
   - Windows *install_dir*\TMAITM6_x64

   where *install_dir* is the path where the agent is installed.
2. Edit the Citrix VDI agent configuration file to set one or more of the *GET_SESSION_LATENCY*, *GET_VDA_MACHINE_METRICS_REMOTELY*, and *RETRIEVE_WINDOWS_EVENTS* variables to `true`.

- Linux `vd.environment`
- Windows `KVDENV_`*`instance_name`*

where *instance_name* is the name of the agent instance.

3. Restart the agent.

   Important: To make these settings the default for all new agent instances, set them to `true` in the configuration template files:

   - Linux This setting is already made the default for new agents instances by editing `vd.environment` in Step 2.
   - Windows `KVDENV`

Example

```
GET_SESSION_LATENCY=true
GET_VDA_MACHINE_METRICS_REMOTELY=true
RETRIEVE_WINDOWS_EVENTS=true
```

Configuring DataPower monitoring

To monitor DataPower appliances, you need to first complete some configuration tasks on your appliances, and then configure the Monitoring Agent for DataPower.

Tip: Click APM v8: Configuring DataPower monitoring in IBM APM to watch a video that covers the basic configuration process of DataPower monitoring.

Configuring DataPower Appliances

Before you configure the Monitoring Agent for DataPower, you must complete some configuration tasks on your appliances.

Tip: For information about the supported DataPower Appliances, see the Prerequisites tab in Software Product Compatibility Reports.

You can monitor DataPower appliances at three different levels. Configure the three levels according to your needs, on each DataPower appliance that you want to monitor to display DataPower appliance data in the Cloud APM console.

1. Resource monitoring

   To see monitoring data, such as resource utilization, throughput, and connection statistics, enable resource monitoring. For instructions, see "Resource monitoring" on page 231.

2. Middleware transaction tracking

   To see monitoring data for transactions, such as transaction detailed information, volume, and dependencies, enable middleware transaction tracking. For instructions, see "Middleware transaction tracking" on page 232.

3. Instance-level transaction tracking of the DataPower appliance

   To display monitoring data for transactions in instance topologies, configure instance-level transaction tracking of the DataPower appliance. For instructions, see "Instance-level transaction tracking of DataPower appliance" on page 233.

Important: Make sure that the user ID has the proper permissions to configure the DataPower appliance. You can enter `*/*/*?Access=r` in the **Access profile** field for the user ID that is used to configure the DataPower appliance. And then use this user ID to configure the DataPower appliance.

Exporting the public certificate

If the XML Management Interface of the DataPower Appliance has the SSL Proxy Profile enabled, you must export the public certificate that is used by the XML Management Interface of the DataPower Appliance to the machine that runs the DataPower agent.

Procedure

1. To download the crypto certificate that is used by the XML Management Interface of the DataPower appliance, for example, `pubcert:///mycert.pem`, click **Administration** > **Main** > **File Management** and save the certificate to the machine that runs the DataPower agent.
2. When you configure the DataPower agent, there is an option to specify the **SSL Proxy Profile** field. Enter the absolute path of the public certificate.

   Note: When more multi-protocol gateways are added, you need to repeat these steps.

Resource monitoring

The first level of monitoring available for a DataPower appliance is to enable resource monitoring, such as SOAP management, statistics, and transaction rates.

The operation on the DataPower Gateway user interface (UI) in the following configuration tasks apply to DataPower Gateway Version 7.5.1 and former versions. If the version of the DataPower Gateway that you use is later than V 7.5.1, you can click the question mark on the upper right in the UI and choose **WebGUI** to return to the UI of the former version. And then follow the instructions to complete DataPower appliance configuration tasks.

Enabling SOAP management

If you want the DataPower agent to collect data from DataPower Appliances, you must configure the XML Management Interface and enable SOAP Management.

Procedure

To enable SOAP:

1. Log on to the WebGUI for the DataPower Appliance that you want to monitor.
2. Click **Objects** > **Device Management** > **XML Management Interface**.

   Note: Ensure that the Administrative state is enabled.
3. For **Port Number**, enter the port number on which the DataPower agent listens for notification reports. The port number is 5550 by default.
4. For **Enabled Services**, ensure that **SOAP Management** is selected.

Enabling Statistics

If you want the DataPower agent to collect data from DataPower Appliances, Statistics must be enabled.

Procedure

To enable Statistics, complete the following steps:

1. Log on to the WebGUI for the DataPower Appliance that you want to monitor.
2. Click **Administration** > **Device** > **Statistics Settings**.
3. Enable **Statistics Settings** and click **Apply**.

Enabling Transaction Rate

If you want the DataPower agent to collect data from DataPower Appliances, the Transaction Rate must be enabled.

Procedure

To enable Transaction Rate, complete the following steps:

1. Log on to the WebGUI for the DataPower Appliance that you want to monitor.

2. Select the `default` domain.
3. Click **Status** > **Connection** > **Transaction Rate**.
4. If **Statistics is currently disabled** is displayed, click **disabled** and in the Statistic Settings, set the **Administrative state** to **enabled**.
5. If you have multiple domains, click **Show All Domains** and repeat steps 3-4 to enable the Transaction Rate for all applicable domains.
6. Click **Apply**.

Middleware transaction tracking
The second level of monitoring available for a DataPower appliance is to display Middleware transaction tracking in the workspaces.

The operation on the DataPower Gateway user interface (UI) in the following configuration tasks apply to DataPower Gateway Version 7.5.1 and former versions. If the version of the DataPower Gateway that you use is later than V 7.5.1, you can click the question mark on the upper right in the UI and choose **WebGUI** to return to the UI of the former version. And then follow the instructions to complete DataPower appliance configuration tasks.

Both transaction tracking of SOAP traffic and REST traffic through the DataPower appliance is supported. DataPower transaction tracking supports SOAP using files `store:///soapreq.xsl`, `store:///soaprsp.xsl`, and `store:///soaperror.xsl`. These XSL files instrument the Web Service Proxy to add and report on `kd4:KD4SoapHeaderV2` in the SOAP Envelope.

In addition to the `soap*.xsl` files, DataPower transaction tracking also includes `apm_req.xsl`, `apm_rsp.xsl`, and `apm_error.xsl`, which support incoming HTTP requests containing an `ARM_CORRELATOR: HTTP Header`, or a SOAP Envelope containing `ITCAMCorrelator` or `kd4:KD4SoapHeaderV2`. The Web Service Proxy updates or sets the outgoing request to contain an `ARM_CORRELATOR: HTTP Header`, and removes the SOAP correlators.

Note: If DataPower appliances are added to a business application, and the appliance carries traffic for multiple applications, after Transaction tracking is enabled, the application topology displayed for those business applications includes paths to nodes for all applications.

Configuring Web Service Management
Complete these steps for each DataPower appliance for which you want to display tracking data.

1. Log on to the WebGUI for the DataPower Appliance that you want to monitor.
2. Select the `default` domain.
3. Search for `XML Management Interface`. Set the following values and click **Apply**.
   - On the **Main** tab, in the **Enabled services** section, enable **WS-Management endpoint**
4. Search for `Web Services Management Agent`. Set the following values and click **Apply**.
   - Set the **Administrative state** to **enabled**
   - Set the **Capture Mode** to **None**
   - Set the **Buffering Mode (deprecated)** to **Discard**
5. Configure the Web Service Proxy or Multi-Protocol Gateway as described in the following topics.

Configuring Web Service Proxy
Complete these steps for each Web Service Proxy for which you want to display tracking data.

Procedure

1. Select the domain of which the Web Service Proxy is a part.
2. On the **Proxy Settings** tab, set the following values and click **Apply**:
   - Set the **Monitor via Web Services Management Agent** to **on**
3. To report SOAP faults, disable error processing and enable error reporting in Cloud APM console: on the **Advanced Proxy Settings** tab, set **Process HTTP Errors** to `off`, and click **Apply**.

Configuring Multi-Protocol Gateway
Complete these steps for each Multi-Protocol Gateway for which you want to display transaction tracking data.

Procedure

1. Select the domain of which the Multi-Protocol Gateway is a part.
2. On the **Advanced** tab for the Multi-Protocol Gateway, set the following values and click **Apply**:
   - Set the **Monitor via Web Services Management Agent** to **on**
   - If the web server uses redirects, set **Follow Redirects** to **off**. Then set the **Rewrite Location URL** to **on**.
3. If you are monitoring a Multi-Protocol Gateway with Response Type or Request Type of `Non-XML`, you must define a Multi-Protocol Gateway Policy with rules covering both Client to Server and Server to Client directions. If a Non-XML Multi-Protocol Gateway does not have rules in its policy, no traffic is captured by either the Web Services Management Agent or a DataPower debug Probe (if enabled).
4. To propagate the HTTP Response Code from the back-end server and to report SOAP faults, on the **Advanced Settings** tab, set **Process Backend Errors** to `off`, and click **Apply**.

Instance-level transaction tracking of DataPower appliance

The third level of monitoring available for a DataPower appliance is to display its data in instance topologies.

The operation on the DataPower Gateway user interface (UI) in the following configuration tasks apply to DataPower Gateway Version 7.5.1 and former versions. If the version of the DataPower Gateway that you use is later than V 7.5.1, you can click the question mark on the upper right in the UI and choose **WebGUI** to return to the UI of the former version. And then follow the instructions to complete DataPower appliance configuration tasks.

Configuring transforms
Complete these steps on each DataPower appliance that you want to display in instance topologies.

About this task

For IBM Performance Management V8.1.2 Fix Pack 1, transaction tracking of SOAP traffic through the DataPower appliance is supported. DataPower transaction tracking supports SOAP using files `store:///soapreq.xsl`, `store:///soaprsp.xsl`, and `store:///soaperror.xsl`. These XSL files instrument the Web Service Proxy to add and report on kd4:`KD4SoapHeaderV2` in the SOAP Envelope.

For IBM Performance Management V8.1.3 and later, transaction tracking of REST traffic through the DataPower appliance is also supported. In addition to the `soap*.xsl` files, DataPower transaction tracking also includes `apm_req.xsl`, `apm_rsp.xsl`, and `apm_error.xsl`, which support incoming HTTP requests containing an `ARM_CORRELATOR: HTTP Header`, or a SOAP Envelope containing `ITCAMCorrelator` or kd4:`KD4SoapHeaderV2`. The Web Service Proxy updates or sets the outgoing request to contain an `ARM_CORRELATOR: HTTP Header`, and removes the SOAP correlators.

The DataPower agent supports transaction tracking for SOAP traffic through the DataPower appliance, REST traffic through the DataPower appliance, and the traffic between DataPower and WebSphere MQ.

- If you want to enable transaction tracking for SOAP and REST traffic through the DataPower appliance, apply `apm_req.xsl`, `apm_rsp.xsl`, and `apm_error.xsl`, which support incoming HTTP requests containing an `ARM_CORRELATOR: HTTP Header`, or a SOAP Envelope containing `ITCAMCorrelator` or kd4:`KD4SoapHeaderV2`. The Web Service Proxy updates or sets the outgoing request to contain an `ARM_CORRELATOR: HTTP Header`, and removes the SOAP correlators.
- In addition to SOAP and REST traffic through the DataPower appliance, if you want to enable transaction tracking between DataPower and WebSphere MQ, apply the `apm_req_MQ.xsl`, `apm_rsp_MQ.xsl`, and `apm_error_MQ.xsl` files. Transaction tracking for SOAP and REST traffic is also enabled automatically after you apply these files.

Procedure

To track REST traffic and enable transaction tracking between DataPower and WebSphere MQ, complete the following steps:

1. Download the files from the following location:
   - For Linux systems, `/opt/ibm/apm/agent/`*`lx8266`*`/bn/bin`
   - For AIX systems, `/opt/ibm/apm/agent/`*`aix536`*`/bn/bin`
2. Upload the XSL files to each DataPower appliance that you want to monitor as part of the IBM integration stack.
3. Configure the Web Service Proxy or Multi-Protocol Gateway as described in the following topics.
4. For each Domain you want to monitor, configure it with the following steps:

   a) Select the Domain from the drop-down list in the DataPower Gateway header.

   b) In the Control Panel navigator, select **Objects** > **Device Management** > **Web Services Management Agent**.

   c) Set the **Buffering Mode (deprecated)** to `Discard`.

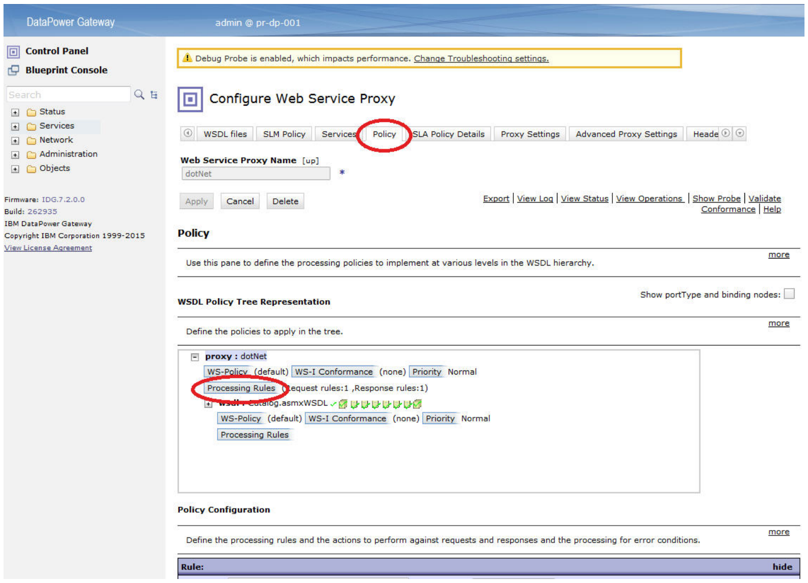
   d) Click **Apply**.

Configuring Web Service Proxy
Complete these steps on each Web Service Proxy that you want to display in instance topologies.

Procedure

In the WebGUI, complete the following steps for each Web Service Proxy that you want to monitor:

1. In the **Configure Web Service Proxy** page, select the name of the Web Service Proxy to configure.
2. On the **Policy** tab, expand **proxy :** ***domain*** and click **Processing Rules**.



Debug Probe is enabled, which impacts performance. Change Troubleshooting settings.

Firmware: IDG.7.2.0.0
Build: 262935
IBM DataPower Gateway
Copyright IBM Corporation 1999-2015
View License Agreement

Web Service Proxy Name [up]
dotNet *

Policy

Use this pane to define the processing policies to implement at various levels in the WSDL hierarchy.

WSDL Policy Tree Representation

Show portType and binding nodes: □

Define the policies to apply in the tree.

proxy : dotNet
WS-Policy (default) WS-I Conformance (none) Priority Normal
Processing Rules (Request rules:1 ,Response rules:1)
wsdl : Catalog.asmxWSDL
WS-Policy (default) WS-I Conformance (none) Priority Normal
Processing Rules

Policy Configuration

Define the processing rules and the actions to perform against requests and responses and the processing for error conditions.

3. In the **Policy Configuration** section, select an existing `Client to Server` rule, or click **New Rule** to create one.

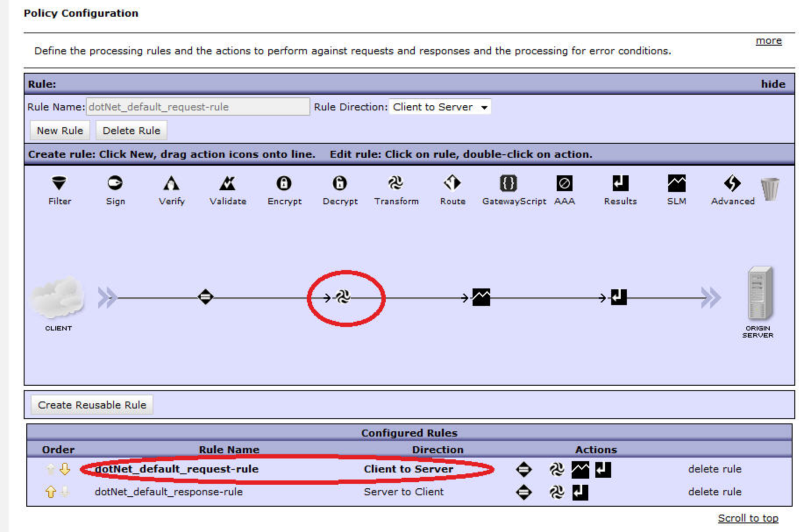
   a. Drag a Transform to the timeline.

   Note:

   1) If a `Client to Server` rule already exists, add the transform node to it.

   2) If the `Client to Server` rule has an Authentication, Authorization, and Audit (AAA) node, ensure that the transform node that includes the DataPower agent xslt file precedes the AAA node.

   b. Double-click the Transform to edit it.



Policy Configuration

Define the processing rules and the actions to perform against requests and responses and the processing for error conditions.

Rule:

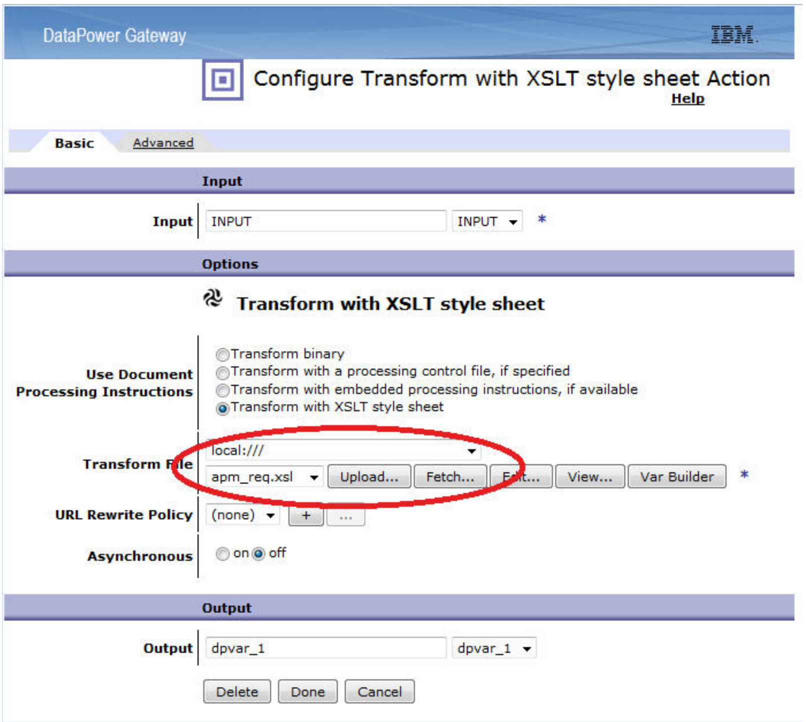
Rule Name: dotNet_default_request-rule   Rule Direction: Client to Server

Create rule: Click New, drag action icons onto line.   Edit rule: Click on rule, double-click on action.

Configured Rules				
Order	Rule Name	Direction	Actions	
	dotNet_default_request-rule	Client to Server		delete rule
	dotNet_default_response-rule	Server to Client		delete rule

c. In the **Configure Transform with XSLT style sheet Action** window, next to Transform File, select `apm_req.xsl` from the data store that you uploaded it to. For example, `local:///`

If the file does not exist, click **Upload** to get it from the installed location.



Configure Transform with XSLT style sheet Action

Input

Input: INPUT INPUT *

Options

Transform with XSLT style sheet

Use Document Processing Instructions:
- ○ Transform binary
- ○ Transform with a processing control file, if specified
- ○ Transform with embedded processing instructions, if available
- ● Transform with XSLT style sheet

Transform File: local:/// apm_req.xsl *

URL Rewrite Policy: (none)

Asynchronous: ○ on ● off

Output

Output: dpvar_1 dpvar_1

Tip: In addition to SOAP and REST traffic through the DataPower appliance, if you want to configure a `Client to Server` rule to monitor the traffic between DataPower and WebSphere MQ, apply the `apm_req_MQ.xsl` file instead of the `apm_req.xsl` file in this step.

d. Click **Done**.

4. Back in the **Policy Configuration** section, repeat Step 3 to configure a `Server to Client` rule, or click **New Rule** to create one.

   a. Drag a Transform to the timeline.

   b. Double-click the Transform to edit it.

   c. In the **Configure Transform with XSLT style sheet Action** window, next to Transform File, select `apm_rsp.xsl` from the data store that you uploaded it to. For example, `local:///`

   If the file does not exist, click **Upload** to get it from the installed location.

Tip: In addition to SOAP and REST traffic through the DataPower appliance, if you want to configure a `Server to Client` rule to monitor the traffic between DataPower and WebSphere MQ, apply the `apm_rsp_MQ.xsl` file instead of the `apm_rsp.xsl` file in this step.

   d. Click **Done**.

5. Back in the **Policy Configuration** section, repeat Step 3 to configure an `Error` rule, or click **New Rule** to create one.

   a. Drag a Transform to the timeline.

   b. Double-click the Transform rule to edit it.

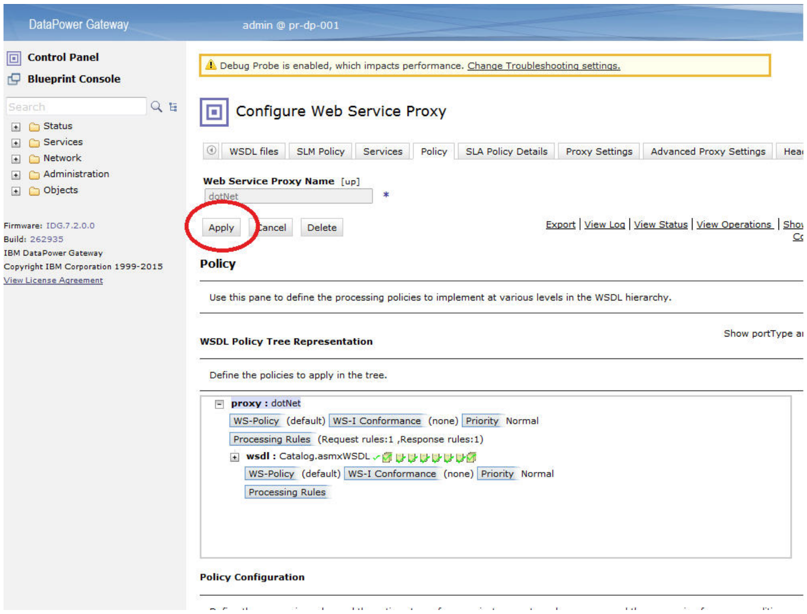
   c. In the **Configure Transform with XSLT style sheet Action** window, next to Transform File, select `apm_error.xsl` from the data store that you uploaded it to. For example, `local:///`

   If the file does not exist, click **Upload** to get it from the installed location.

   Tip: In addition to SOAP and REST traffic through the DataPower appliance, if you want to configure an `error` rule to monitor the traffic between DataPower and WebSphere MQ, apply the `apm_error_mq.xsl` file instead of the `apm_error.xsl` file in this step.

   d. Click **Done**.

6. Back in the **Configure Web Service Proxy** page, click **Apply**.



Debug Probe is enabled, which impacts performance. Change Troubleshooting settings.

Configure Web Service Proxy

Web Service Proxy Name [up]: dotNet

Policy

Use this pane to define the processing policies to implement at various levels in the WSDL hierarchy.

WSDL Policy Tree Representation

Define the policies to apply in the tree.

proxy : dotNet
WS-Policy (default) WS-I Conformance (none) Priority Normal
Processing Rules (Request rules:1 ,Response rules:1)
wsdl : Catalog.asmxWSDL
WS-Policy (default) WS-I Conformance (none) Priority Normal
Processing Rules

Policy Configuration

Firmware: IDG.7.2.0.0
Build: 262935
IBM DataPower Gateway
Copyright IBM Corporation 1999-2015

Configuring Multi-Protocol Gateway
Complete these steps on each Multi-Protocol Gateway that you want to display in instance topologies.

Procedure

In the WebGUI, complete the following steps for each Multi-Protocol Gateway that you want to monitor.

1. In the **Configure Multi-Protocol Gateway** page, click the name of the Multi-Protocol Gateway that you want to configure.
2. On the **Multi-Protocol Gateway Policy** page, configure the policy. Click **...**.
3. On the **Configure Multi-Protocol Gateway Style Policy** page, select an existing `Client to Server` rule, or click **New Rule** to create one.

   a. Drag a Transform to the timeline.

   Note:

   1) If a `Client to Server` rule already exists, add the transform node to it.

   2) If the `Client to Server` rule has an Authentication, Authorization, and Audit (AAA) node, ensure that the transform node that includes the DataPower agent xslt file precedes the AAA node.

   b. Double-click the Transform rule to edit it.

   c. In the **Configure Transform with XSLT style sheet Action** window, next to Transform File, select `apm_req.xsl` from the data store that you uploaded it to. For example, `local:///`

      If the file does not exist, click **Upload** to get it from the installed location.

      Tip: In addition to SOAP and REST traffic through the DataPower appliance, if you want to configure a `Client to Server` rule to monitor the traffic between DataPower and WebSphere MQ, apply the `apm_req_MQ.xsl` file instead of the `apm_req.xsl` file in this step.

   d. Click **Done**.

4. Back on the **Configure Multi-Protocol Gateway Style Policy** page, select an existing `Server to Client` rule, or click **New Rule** to create one.

   a. Drag a Transform to the timeline.

   b. Double-click the Transform rule to edit it.

   c. In the **Configure Transform with XSLT style sheet Action** window, next to Transform File, select `apm_rsp.xsl` from the data store that you uploaded it to. For example, `local:///`

      If the file does not exist, click **Upload** to get it from the installed location.

      Tip: In addition to SOAP and REST traffic through the DataPower appliance, if you want to configure a `Server to Client` rule to monitor the traffic between DataPower and WebSphere MQ, apply the `apm_rsp_MQ.xsl` file instead of the `apm_rsp.xsl` file in this step.

   d. Click **Done**.

5. Back on the **Configure Multi-Protocol Gateway Style Policy** page, select an existing `Error` rule, or click **New Rule** to create one.

   a. Drag a Transform to the timeline.

   b. Double-click the Transform rule to edit it.

   c. In the **Configure Transform with XSLT style sheet Action** window, next to Transform File, select `apm_error.xsl` from the data store that you uploaded it to. For example, `local:///`

      If the file does not exist, click **Upload** to get it from the installed location.

      Tip: In addition to SOAP and REST traffic through the DataPower appliance, if you want to configure an `error` rule to monitor the traffic between DataPower and WebSphere MQ, apply the `apm_error_mq.xsl` file instead of the `apm_error.xsl` file in this step.

   d. Click **Done**.

6. Back on the **Configure Multi-Protocol Gateway Style Policy** page, on the **Advanced** tab, set the **Monitor via Web Services Management Agent** to **on**, and click **Apply**.

7. Click **Apply**.

What to do next

In some cases, adding transforms for Transaction Tracking may result in DataPower changing the value of HTTP Content-Type headers. You may see web pages with images that do not load, or binary files being rendered as garbled HTML text.

The behavior of DataPower changes when comparing a rule with no XSL transforms to a rule with one or more XSL transforms. If the service handles MIME, MTOM, XOP or other encoded messages, this behavior may be desired, otherwise modify your DataPower configuration to prevent the behavior.

To prevent DataPower from modifying the HTTP Content-Type header, set the variable in each affected rule (var://service/mpgw/proxy-content-type):

1. Drag an Advanced object to the rule.

2. Double click the Advanced object to edit it.
3. Select **Set Variable** and click **Next**.
4. Enter the Variable Name `service/mpgw/proxy-content-type` and Variable Value 1 and click **Done**.
5. Apply the policy and service configuration changes.
6. Repeat steps 1-5 for each affected rule.

Configuring the DataPower agent

The Monitoring Agent for DataPower provides a central point of monitoring for the DataPower Appliances in your enterprise environment. You can identify and receive notifications about common problems with the appliances. The agent also provides information about performance, resource, and workload for the appliances.

About this task

The DataPower agent is a multiple instance agent; you must create the first instance and start the agent manually. The Managed System Name includes the instance name that you specify, for example, *`instance_name`*`:`*`host_name`*`:`*pc*, where *pc* is your two character product code. The Managed System Name is limited to 32 characters.

The instance name that you specify is limited to 28 characters, minus the length of your host name. For example, if you specify `DataPower` as your instance name, your managed system name is `DataPower:hostname:BN`.

Important: If you specify a long instance name, the Managed System name is truncated and the agent code does not display correctly.

Note: The DataPower agent's XSLT does not parse BLOB characters that are used for mainframe applications.

For each production DataPower appliance, configure one instance. If your DataPower appliances are non-production or small ones, you can configure only one agent instance to monitor them all. Multiple instances can run on the same machine. You can run the configuration script to create an instance and change any configuration settings. You can edit the agent silent response file before you run the script to bypass the prompts and responses that are required.

Procedure

- To configure the DataPower agent, complete one of the following procedures:
  - Linux AIX To configure the agent by responding to prompts, complete the following steps:
    1. Go to the *`install_dir`*`/bin` directory, where *`install_dir`* is the installation directory for the DataPower agent.
    2. Run the `./datapower-agent.sh config` *`instance_name`* command.

       Choose an *instance_name* that is unique on the server.
    3. When prompted to edit the DataPower agent settings, enter 1 to proceed.
    4. When prompted to edit the **`Managed System Details`**, enter one of the following options:
       - `1=Add`
       - `2=Edit`
       - `3=Del`
       - `4=Next`
       - `5=Exit`

If it is the first time that you configure a DataPower agent instance on your system, the `No 'DataPower Appliances' settings available` message is displayed. Enter 1 to add a DataPower Appliances setting. The default is option 5=`Exit`.

5. Enter the properties for the DataPower appliance:

   Managed System Name
   For **`Managed System Name`**, enter the managed system name of the agent.

   Choose a **`Managed System Name`** that is unique among all instances of the agent and that can be used to easily identify an appliance. The name should contain only alphanumeric characters, for example, the host name of the DataPower appliance.

   Device Host
   For **`Device Host`**, enter the IP address of the monitored DataPower Appliance.

   XML Management Interface Port
   For **`XML Management Interface Port`**, enter the port number for the XML Management Interface. The default number is 5550.

   User ID
   For **`User ID`**, enter the User ID to log in to the monitored DataPower Appliance. The default value is admin.

   Password
   For **`Password`**, enter the password to log in to the monitored DataPower Appliance and then confirm the password.

   SSL Proxy Profile
   For **`SSL Proxy Profile`**, enter the absolute path of the public certificate for your SSL proxy profile, if the XML management interface of the device is configured to use the profile. For example,

   ```
   the location of the .pem file exported from datapower appliances/mycert.pem
   ```

   where *the location of the .pem file exported from datapower appliances* is the absolute path of the public certificate. To export the public certificate, see Exporting public certificate.

   SSL Proxy Option
   For **`SSL Proxy Option`**, set to Yes if the XML management interface of the monitored device is configured to use a custom SSL proxy profile. Otherwise, set it to No.

6. To monitor multiple DataPower appliances, repeat "4" on page 238 and "5" on page 239 to configure one agent instance for each DataPower appliance. Otherwise, type 5 and press **Enter** to complete the configuration.
7. Run the following command to start the agent:

   ```
   ./datapower-agent.sh start instance_name
   ```

- Silent configuration

  1. To configure the agent by editing the silent response file and running the script with no interaction, complete the following steps:

     - Linux AIX Open *`install_dir`*`/samples/datapower_silent_config.txt` in a text editor.
     - Windows Open *`install_dir`*`/samples/datapower_silent_config.txt` in a text editor.

  2. To configure the DataPower agent to monitor an appliance, enter the following properties:

     Device Host
     Enter the host name or IP address of the device. For example, **`SOAP_HOST.ManageSystemName=`** *datapower01*.

XML Management Interface Port
Enter the port number for the XML Management Interface. The default value is 5550. For example, **DP_PORT.ManageSystemName=** *5550*.

User ID
Enter the User ID that is used to connect to the device. The default value is admin. For example, **DP_UID.ManageSystemName=** *admin*.

Password
Enter the password of the User ID. For example, **DP_PASSWORD.ManageSystemName=** *password*.

SSL Proxy Profile
Enter the absolute path of the public certificate for your SSL proxy profile, if the XML management interface of the device is configured to use the profile. For example,

```
the location of the .pem file exported from datapower appliances/mycert.pem
```

where *the location of the .pem file exported from datapower appliances* is the absolute path of the public certificate. To export the public certificate, see Exporting public certificate.

SSL Proxy Option
For **SSL Proxy Option**, set to Yes if the XML management interface of the monitored device is configured to use a custom SSL proxy profile. Otherwise, set it to No. For example, **DP_SSL_OPTION.ManageSystemName1=** Yes.

Important: ManageSystemName is unique. You must replace it with your own system name in all entries. If you want to monitor multiple appliances, copy and repeat the steps that are shown to monitor an appliance. Remember to set the appropriate ManageSystemName and DataPower Appliance parameters.

3. Go to the installation directory for the agent and run the following command to start the agent:

```
./datapower-agent.sh start instance_name
```

What to do next

- To check the names and settings of the configured agent instances, run the **./cinfo -s bn** command.
- You can verify that the DataPower agent data is displayed in the Cloud APM console. For instructions on how to start the Cloud APM console, see Starting the Cloud APM console. For information about using the Applications editor, see Managing applications.
- To display transaction tracking data in the Cloud APM console, configure transaction tracking for the DataPower agent. For instructions, see Configuring transaction tracking for the DataPower agent.
- To display monitoring at different levels, configure the DataPower appliance accordingly. For instructions, see Resource monitoring, Middleware transaction tracking, and Instance-level transaction tracking of DataPower appliances.

Configuring transaction tracking for the DataPower agent

To display transaction tracking data for DataPower appliances in the middleware and topology dashboards, you must enable transaction tracking for the DataPower agent.

Before you begin

- Install the DataPower agent and configure it to connect to the DataPower appliance.
- Enable monitoring for SOAP or ARM in the DataPower appliance.

Procedure

To enable transaction tracking for the DataPower agent, complete the following steps:

1. In the navigation bar, click **System Configuration** > **Agent Configuration**.

2. On the **DataPower** tab, select the agent instances for which you want to enable transaction tracking.
3. Select **Actions** > **Set Transaction Tracking** > **Enabled** to enable transaction tracking. The status of the agent in the **Transaction Tracking** column is updated to **Enabled**.

Results

You have enabled transaction tracking for the selected agent instances.

What to do next

To see data for a DataPower appliance in the middleware and topology dashboards, you must now add the appliances that you want to monitor to the Application Performance Dashboard. For further information about adding a DataPower appliance to the Application Performance Dashboard, see "Adding middleware applications to the Application Performance Dashboard" on page 100.

Note: If you are using Integration Services and you want to monitor the data that is transmitted between IBM Integration Bus and DataPower, additional configuration is required to show an accurate Aggregate Transaction Topology. The IBM Integration Bus agent cannot include correlation support for SOAP messages without a SOAP envelope. `SOAPRequest` nodes, `SOAPAsyncRequest` nodes, and `SOAPReply` nodes may accept messages without SOAP Envelopes as input messages. For these nodes, there is no relationship displayed in the topology view from the mediation to the downstream mediation or the application server. To avoid this problem, insert a `SOAPEnvelope` node immediately before the `SOAPRequest`, `SOAPAsyncRequest`, or `SOAPReply` nodes in your IBM Integration Bus message flow, and select the **Create new envelope** option for the `SOAPEnvelope` node to add a SOAP envelope for the SOAP message.

Configuring Db2 monitoring

The Monitoring Agent for Db2 monitors the availability and performance of the Db2 server. You can monitor multiple servers from the Cloud APM console; each server is monitored by a Db2 agent instance. Remote monitoring is also supported by Db2 agent.

Before you begin

Review hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Db2 agent.

About this task

The Db2 agent is a multiple instance agent, you must first create the instance and then start the agent manually.

The managed system name includes the agent instance name that you specify, for example, *`instance_name`*`:`*`host_name`*`:`*`pc`*.

Where:

- The *pc* is your two character product code.
- The *instance_name* is the agent instance name, and it must be the same as the Db2 instance name that is to be monitored.

The managed system name can contain up to 32 characters. The agent instance name that you specify can contain up to 8 characters, excluding the length of your host name. For example, if you specify `DB2inst1` as your agent instance name, your managed system name is `DB2inst1:hostname:ud`.

Important: If you specify a long agent instance name, the managed system name is truncated and the agent code is not displayed completely.

To avoid permission issues when you configure the agent, be sure to use the same root user or non-root user ID that was used for installing the agent. If you installed your agent as a selected user and want to configure the agent as a different user, see "Configuring agents as a non-root user" on page 183. If you installed and configured your agent as a selected user and want to start the agent as a different user, see "Starting agents as a non-root user" on page 992.

Run the configuration script to create an instance and change the configuration settings. You can edit the Db2 silent response file before you run the configuration script to bypass the prompts and responses that are otherwise necessary.

After you configure the Db2 agent, be sure to start the agent with a user ID that has the Db2 SYSADM authority for the monitored instance. The agent requires the SYSADM authority to turn on all monitor switches and collect the monitoring data. Therefore, a user with the SYSADM authority must start the agent. Use the instance owner user, which has the SYSADM authority, to start the agent.

The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see "Change history" on page 54.

Procedure

To configure the agent with the default settings, complete the following steps:

1. Run the following command where *instance_name* is the name that you want to give to the instance:

   ```
   install_dir/bin/db2-agent.sh config instance_name
    install_dir/samples/db2_silent_config.txt
   ```

   The agent instance name *instance_name* is always the same as the Db2 instance name that is being monitored. For more details about the existing agent instances, refer "Agent Configuration page" on page 182

2. Run the following command to start the Db2 agent:

   ```
   install_dir/bin/db2-agent.sh start instance_name
   ```

What to do next

- Grant privileges to the Db2 user to view data for some attributes of the Db2. For information about granting these privileges, see "Granting privileges for viewing Db2 metrics" on page 246.
- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring the agent on Windows systems

You can use the IBM Cloud Application Performance Management window to configure the agent on Windows systems.

Before you begin

Before you start configuring the Db2 agent for local and remote monitoring, ensure that following task is completed for remote monitoring.

- Set up client/server environment for remote monitoring, refer "Prerequisites for Remote Monitoring" on page 250.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Cloud Application Performance Management**.
2. In the IBM Cloud Application Performance Management window, right-click **Monitoring Agent for DB2**, and then click **Configure agent**.
3. In the **Enter a unique instance name** field, type the agent instance name and click **OK**.

   Important: For local monitoring, the agent instance name must match the name of the Db2 instance that is being monitored.

   For remote monitoring, the agent instance name must be the unique catalog node name.
4. In the **Monitoring Agent for DB2** window, complete these steps:

a) In the **Username**, enter the user name of Db2 instance.

For Local Db2, enter the name of Db2 instance owner.

For Remote Db2, enter Actual Db2 instance owner name from remote Db2 machine.

Important: This parameter is mandatory for monitoring remote Db2 instance.

b) In the **Password**, enter the password of Db2 instance.

For Local Db2, enter the password of Db2 instance owner.

For Remote Db2, enter Actual Db2 instance owner password from remote Db2 machine.

Important: This parameter is mandatory for monitoring remote Db2 instance.

c) In the **DB2Customized SQL Definition File** field, enter the full file path name for the SQL definition file. If the SQL definition file is in the default directory, leave this field blank. Otherwise, enter the full file path name of the file. The default file name with path is as follows:

Linux AIX `CANDLEHOME/config/kudcussql.properties`

Windows `CANDLEHOME\TMAITM6_x64\kudcussql.properties`

d) In the **db2diag Log File Path** field, enter the directory path for the db2diag log file. If the db2diag log file is in the default directory, leave this field blank. Otherwise, enter the path of the directory. The default directory path is as follows:

Linux AIX /home/*DB2owner_home_dir*/sqllib/db2dump

Windows C:\ProgramData\IBM\DB2\DB2COPY\DB2INSTANCENAME

Note: This parameter is not applicable for remote monitoring.

e) In the **MSGID Filter in Regular Expression** field, enter the *MSGID* to filter the diagnostic log. The MSGID is a combination of the message type, message number, and severity level. Use a regular expression to filter the log based on message type, message number, or severity level, for example, `ADM1\d*1E|ADM222\d2W`.

f) From the **Enable Monitoring for Partitions in Remote Hosts** list, select `Yes` to specify that the Db2 agent can monitor partitions in remote hosts.

g) From the **Enable Monitoring All Databases** list, select `Yes` to specify that the Db2 agent can monitor all databases.

h) Click **OK**.

The agent instance is displayed in the IBM Cloud Application Performance Management window.

5. Run following steps to configure remote monitoring.

a) Open *install_dir*\TMAITM6_x64\KUDENV_<instanceName>.

b) Set *KUD_DB2_CLIENT_INST* to Db2 client instance name under which remote Db2 server instance is cataloged.

6. Right-click the **Monitoring Agent for DB2** instance, and click **Start**.

What to do next

- Grant privileges to the Db2 user to view data for some attributes of the Db2 agent. For information about granting these privileges, see "Granting privileges for viewing Db2 metrics" on page 246.
- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring the agent on Linux or UNIX systems

You run the configuration script to configure the agent on Linux systems.

Before you begin

Before you start configuring the Db2 agent for local and remote monitoring, ensure that following task is completed for remote monitoring.

- Set up client/server environment for remote monitoring, refer "Prerequisites for Remote Monitoring" on page 250.

Procedure

1. Run the command `install_dir/bin/db2-agent.sh config instance_name`

   Where *instance_name* is the name that you want to give to the instance:

   Important: For local monitoring, the agent instance name must match the name of the Db2 instance that is being monitored.

   For remote monitoring, the local cataloged node of remote Db2 server instance that is to be monitored.
2. When you are prompted to provide a value for the following parameters, press Enter to accept the default value, or specify a value and then press Enter:
   - Username
   - Password
   - DB2® SQL path
   - Diaglog path
   - Diaglog message ID filter
   - Monitor remote partitions
   - Monitor all databases
3. Run the following command to start the agent:

   For local monitoring run `install_dir/bin/db2-agent.sh start instance_name` by Db2 instance owner user.

   For remote monitoring, run `install_dir/bin/db2-agent.sh start node_name` with the instance owner of Db2 client instance under which remote Db2 server instance is cataloged.

What to do next

- Grant privileges to the Db2 user to view data for some attributes of the Db2 agent. For information about granting these privileges, see "Granting privileges for viewing Db2 metrics" on page 246.
- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring the agent by using the silent response file

Use the silent response file to configure the agent without responding to prompts when you run the configuration script. You can use the silent response file for configuring the agent on both Windows and Linux systems.

Before you begin

Before you start configuring the Db2 agent for local and remote monitoring, ensure that following task is completed for remote monitoring.

- Set up client/server environment for remote monitoring, refer "Prerequisites for Remote Monitoring" on page 250.

About this task

The silent response file contains the configuration parameters. You edit the parameter values in the response file, and run the configuration script to create an agent instance and update the configuration values.

Procedure

1. In a text editor, open the `db2_silent_config.txt` file that is available at the following path:

   Linux AIX *`install_dir`*`/samples/db2_silent_config.txt`

   Windows *`install_dir`*`\tmaitm6_x64\samples\db2_silent_config.txt`

2. In the response file, specify a value for the following parameters:

   - In the **Username**, enter the user name of Db2 instance.

     For Local Db2, enter the name of Db2 instance owner.

     For Remote Db2, enter Actual Db2 instance owner name from remote Db2 machine.

     Important: This parameter is mandatory for monitoring remote Db2 instance.

   - In the **Password**, enter the password of Db2 instance.

     For Local Db2, enter the password of Db2 instance owner.

     For Remote Db2, enter Actual Db2 instance owner password from remote Db2 machine.

     Important: This parameter is mandatory for monitoring remote Db2 instance.

   - For the **`DB2 SQL path`** parameter, leave this field blank if the SQL definition file is available at the default directory. Otherwise, enter the correct directory path. The SQL definition file is available at the following default path:

     Linux AIX `CANDLEHOME/config/kudcussql.properties` For example, **`KUD_DB2_SQL_PATH=`** `/opt/ibm/apm/agent/config/kudcussql.properties`

     Windows `CANDLEHOME\TMAITM6_x64\kudcussql.properties` For example, **`KUD_DB2_SQL_PATH=`** `C:\IBM\ITM\TMAITM6_x64\kudcussql.properties`

   - For the **`dialog path`** parameter, leave this field blank if the db2diag log file is available at the default directory. Otherwise, enter the correct directory path. The log file is available at the following default path:

     Linux AIX `/home/DB2owner_home_dir/sqllib/db2dump`
     For example, **`KUD_DIAGLOG_PATH=`** `/home/db2inst1/sqllib/db2dump`.

     Windows `Windows Install_Driver:\ProgramData\IBM\DB2\DB2COPY`
     `\DB2INSTANCENAME`
     For example, **`KUD_DIAGLOG_PATH=`** `C:\ProgramData\IBM\DB2\DB2COPY1\DB2`

     Note: This parameter is not applicable for remote monitoring.

   - For the **`dialog message ID filter`** parameter, specify the *MSGID* to filter the diagnostic log. The MSGID is a combination of the message type, message number, and severity level. You can also use a regular expression, for example, **`KUD_DIAGLOG_MSGID_FILTER=`** `ADM1\d*1E|ADM222\d2W`.

   - For the **`monitor remote partitions`** parameter, enter Yes to specify that the Db2 agent monitors partitions in remote hosts. For example, **`KUD_MONITOR_REMOTE_PARTITIONS=`** *Yes*.

   - For the **`monitor all databases`** parameter, enter Yes to specify that you want the Db2 agent to monitor all databases. For example, **`KUD_MONITOR_ALL_DATABASES=`** *Yes*.

3. Save and close the `db2_silent_config.txt` file, and run the following command

   Linux AIX *`install_dir`*`/bin/db2-agent.sh config` *`instance_name`*
   `install_dir``/samples/`
   `db2_silent_config.txt`

Windows
```
install_dir\bin\db2-agent.bat configinstance_name
\tmaitm6_x64\samples\db2_silent_config.txt
```

<instance_name> is

- For monitoring Local Db2 server : The Db2 server instance name that you want to monitor.
- For monitoring Remote Db2 server : The catalog node name of remote Db2 server instance.

Important: Ensure that you include the absolute path to the silent response file. Otherwise, agent data is not shown in the dashboards.

4. For Windows, Open the `CANDLEHOME\TMAITM6_x64\KUDENV_<instance_name>` file. And edit the line, `KUD_DB2_CLIENT_INST` as `KUD_DB2_CLIENT_INST=<client instance name under which remote Db2 server instance is cataloged>`
5. Run the following command to start the agent:

   Linux AIX
   ```
   install_dir/bin/db2-agent.sh start instance_name
   ```
   Windows
   ```
   install_dir\bin\db2-agent.bat start instance_name
   ```

   Remember: While monitoring remote Db2 server instance from UNIX or Linux, the command must be executed with the client instance owner under which remote server instance is cataloged.

What to do next

- Grant privileges to the Db2 user to view data for some attributes of the Db2 agent. For information about granting these privileges, see "Granting privileges for viewing Db2 metrics" on page 246.
- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Granting privileges for viewing Db2 metrics

To monitor the Db2 resources, a Db2 user must have the Db2 SYSADM, SYSCTRL, SYSMAINT, and SYSMON authorities for the monitored instance to view data for some attributes of the Db2 agent.

About this task

To view the monitoring data that the agent collects for all the attributes on the dashboard, the Db2 user must have specific privileges. To assign these privileges to the Db2 user, run the script file that is present at the following location:

Linux AIX
```
install_dir/config/KudGrantUserPermissions.sh
```
Windows
```
install_dir\TMAITM6_x64\KudGrantUserPermissions.bat
```

A Db2 user with the SYSADM authority can run the script to grant privileges to itself or to any other Db2 user. For a Db2 instance, use the instance owner, which already has the SYSADM authority, to run the script to grant other permissions to itself or to grant all the permissions to any other Db2 user.

Procedure

1. For local monitoring, refer following steps.
   a) On the system where the Db2 agent in installed, open the Db2 command-line interface.
   b) Run the following command where *instance_name* is the name of the Db2 instance and *username* is the name of the Db2 user:

      Linux AIX
      ```
      install_dir/config/KudGrantUserPermissions.sh
      instance_name username
      ```
      Windows
      ```
      install_dir\TMAITM6_x64\KudGrantUserPermissions.bat
      instance_name username
      ```

      Note: For Windows systems, *username* is optional in the command. If a user name is not specified in the command, the privileges are assigned to the default user (system).

2. For remote monitoring, refer following steps.
   a) Copy `KudGrantUserPermissions.sh` for Unix or Linux and `KudGrantUserPermissions.bat` for Windows from *`install_dir`*`/TMAITM6_x64/` from agent workstation to Db2 server machine.
   b) Run the following command from Db2 instance owner user where *instance_name* is the name of the Db2 instance and *username* is the name of the Db2 user:

      Linux AIX `./KudGrantUserPermissions.sh` *`instance_name username`*

      Windows `KudGrantUserPermissions.bat` *`instance_name username`*

      Remember: For remote Db2 monitoring on Windows, the *username* must be the user name that is provided during the Db2 agent configuration at client workstation.

Configuring local environment variables

You can configure local environment variables to change the behavior of the Db2 agent.

Procedure

1. On Windows systems, click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, from the **Actions** menu, click **Advanced > Edit ENV File**.
3. On Linux or AIX systems, go to the command line and edit the `ud.environment` file from the `install_dir/config` directory. Where, `install_dir` is the agent installation directory.

   Note: The `ud.environment` file is a hidden file.
4. In the environment variable file, enter values for the environment variables.

   For information about the environment variables that you can configure, see "Local environment variables" on page 247.

Local environment variables

You can change the behavior of the Db2 agent by configuring the local environment variables.

Variables for defining the data collection method for the tablespace data set

To set the method for data collection of the tablespace data set, use the following environment variables:

- **`KUD_T1_BY_SQL`**: Use this variable to set the method of data collection for the tablespace data set by using SQL queries. To enable data collection by using SQL queries, set the value of this variable as Y. To collect data for the tablespace data set by using the snapshot method, set the value of this variable as N. The default value of this variable is N.

  Important: To collect data by using SQL queries, the Db2 version must be 9.7, or later. Also, the user who starts the Db2 agent must have the SYSADM authority for all databases.
- **`KUD_T1_DISABLE`**: Use this variable to disable the data collection for the tablespace data set. To enable the data collection for the tablespace data set, set the value of this variable as N. To disable the data collection for the tablespace data set, set the value of this variable as Y. The default value of this variable is N.

Variable for excluding the caching facility (CF) nodes from data collection

To exclude caching facility (CF) nodes from the data collection algorithm in pureScale® environment, use the **`DB2_CF_PARTITION_NUMS`** variable. In the agent environment file, set the **`DB2_CF_PARTITION_NUMS`** variable as `DB2_CF_PARTITION_NUMS=<CF node number>`. For example, `DB2_CF_PARTITION_NUMS=1`. For more than one CF node, set the **`DB2_CF_PARTITION_NUMS`** variable value as a list that uses any special symbol from `# . : , ; |` @ as delimiter. For example, `DB2_CF_PARTITION_NUMS=12,13,23,34`. No default value is set for this variable.

Variable for limiting data collection for the Db2 Table data set

To set the maximum number of rows that the Db2 agent must return, while collecting data for the Db2 Table data set, use the **KUD_TABLE_NUMBER** environment variable. The default value is 10000.

Variable for setting the reload interval of the customized SQL properties file

To set the reload time interval (in seconds) for the customized SQL properties file, use the **KUD_CUS_SQL_INTERVAL** variable. The default value is 20 seconds.

Variable for limiting the rows in the data collection for Agent Event data set

To set the number of rows for data collection of the Agent Event data set, use the **KUD_AGENT_EVENT_CACHE** variable. The Agent Event data set provides detailed information about predefined and triggered events and determines problems with the health of the monitored database. The default value is 50.

Variable for limiting the rows in the data collection for Db2 Log Record data set

To set the number of rows for data collection of the Db2 Log Record data set, use the **KUD_DBHISTORY_MAXROW** variable. The Db2 Log Record data set provides historical information about the Db2 archive log. The default value is 500.

Variables for defining the data collection for the Db2 Diagnostic Log data set

To set the method for data collection of the Db2 Diagnostic Log data set, use the following environment variables:

- **KUD_DIAGLOG_BY_TABLE**: Use this variable to set the method of data collection for the Db2 Diagnostic Log data set. If the value of this variable is set to Y, then data for the Db2 Diagnostic Log data set is collected by using SQL queries. If the value of this variable is set to N, then data for the Db2 Diagnostic Log data set is collected by parsing the db2diag.log. The default value of this variable is Y.

  Important: To collect data by using SQL queries, the Db2 version must be 10, or later.
- **KUD_DIAGLOG_TAILCOUNT**: Use this variable to define the number of lines of the db2diag.log file that the Db2 agent parses for collecting data for the DB2 Diagnostic Log data set. This variable limits the Db2 agent to process the Db2 agent log file so that only the latest messages and events are monitored. The default value of this variable is 1000.
- **KUD_DIAGLOG_CACHE**: Use this variable to limit the number of log records that are displayed on the dashboard for the Db2 Diagnostic Log data set. The default value of this variable is 20.
- **KUD_DIAGLOG_INTERVAL**: Use this variable to define the reload time interval (in seconds) for the db2diag.log file for data collection for the Db2 Diagnostic Log data set. The default value of this variable is 30 seconds.
- **KUD_DISABLE_DIAGLOG**: Use this variable to disable the data collection for the Db2 Diagnostic Log data set. To enable the data collection for the Db2 Diagnostic Log data set, set the value of this variable as N. To disable the data collection for the Db2 Diagnostic Log data set, set the value of this variable as Y. The default value of this variable is N.

Variable for setting the query timeout interval

If an SQL query takes a very long time to complete, it affects the performance of the Db2 agent. To set the query timeout interval for the Db2 agent, use the **KUD_QUERY_TIMEOUT** variable. Use this variable to define the maximum amount of time (in seconds) that the Db2 agent waits to receive a response for a query that is sent to the Db2 server. The value for this variable must be less than 300 seconds. The default value of this variable is 45 seconds.

Variable for defining the data collection for the DB2 Database01 (Superseded) data set

The agent must not trigger ASN queries to collect data for the DB2 Database01 (Superseded) data set when ASN schemas are not present. To enable the execution of the ASN queries, use the **KUD_REPLICATION_ON** variable. If the value of this variable is set to Y, the Db2 agent runs ASN queries even when the ASN schemas are not present. If the value of this variable is set to N, the Db2 agent does not run the ASN queries. The default value of this variable is Y.

Variable for configuring the monitor switches when collecting data by using the snapshot method

If you want to collect the Db2 agent monitoring data by using the snapshot method, enable the Db2 monitor switch for the data set. To enable the Db2 monitor switch, use the **KUD_MON_SWITCH_OVERRIDE** variable. The list of Db2 monitor switches is as follows:

LOCK
Lock Information

SORT
Sorting Information

STATEMENT
SQL Statement Information

TABLE
Table Activity Information

TIMESTAMP
Take Timestamp Information

UOW
Unit of Work Information

If the value of this variable is set to Y, the Db2 agent retains the configuration setting of the Db2 monitor switches. If the value of this variable is set to N, the Db2 agent enables all the monitor switches to collect data. The default value of this variable is N.

Variable for tracing the Db2 snapshot buffer data of an data set

To view the data that is collected for an data set by using the snapshot method, use the **KUD_SNAPSHOT_DUMPOUT** variable. If the value of this variable is set to Y, the Db2 agent dumps the snapshot buffer data for attribute groups in the agent log file. If the value of this variable is set to N, the Db2 agent does not dump the snapshot buffer data in the agent log file. The default value of this variable is N.

Variable for tracing the Db2 agent by using the snapshot buffer data of an data set

To trace the Db2 agent by using the snapshot buffer data that is collected for an data set, use the **KUD_SNAPSHOT_READIN** variable. To enable the tracing of Db2 agent, set the value of this variable as Y. To disable the tracing of Db2 agent, set the value of this variable as N.

Variable for defining the data collection method for the Locking Conflict data set

To set the method of data collection for the Locking Conflict data set, use the **KUD_LOCKCONFLICT_BY_SQL** variable. To collect data for the Locking Conflict data set by using SQL queries, set the value of this variable as Y. To collect data for the Locking Conflict data set by using the snapshot method, set the value of this variable as N. The default value of this variable is Y.

Important: To collect data by using SQL queries, the Db2 version must be 9.7 FP1, or later. Also, the user who starts the Db2 agent must have SYSADM authority for all databases.

Variable to monitor remote Db2 server on Windows

KUD_DB2_CLIENT_INST: Set this variable to Db2 client instance name under which remote Db2 server instance is cataloged. You need to set this variable only if you are using remote monitoring where agent is on Windows.

Prerequisites for Remote Monitoring

You can use Monitoring Agent for Db2 for remote monitoring. Refer the topic for prerequisites of remote monitoring of Db2.

About this task

For remote monitoring of Db2, you must first do the basic Db2 client/server environment setup. Do this setup for Windows and UNIX or Linux.

For this set up a user must have Db2 SYSADM or SYSCTRL authority.

Remember: Run all the steps on agent workstation except for step 2.

Procedure

1. On the Db2 agent workstation, install Db2 client. The version of this client must be greater than or equal to that of Db2 server instance version that is to be monitored.
2. Verify that the communication protocol for Db2 instance is TCPIP.

   a) To verify, run the command **db2set** on the Db2 `command line`.

   b) If it is not set to TCPIP, then run **db2set DB2COMM=tcpip** in Db2 `command line`.

   Important: This step is done at the server side.
3. Catalog the remote server instance at Db2 agent workstation with following command.

   Important: The server instance is to be cataloged under the client instance. So run following command on the client instance.

   db2=>CATALOG TCPIP NODE<node_name> REMOTE <hostname/ip_address> SERVER <service_name/port_number>

   on Db2 where

   a. *<node_name>* represents a local nickname of Db2 instance on client component.

      Note: For UNIX or Linux, *<node_name>* must not be same as of any Db2 client or Db2 server instance name available on the same workstation.

   b. <hostname/ip_address> represents name or IP address of the Db2 server workstation.

   c. <service_name/port_number> at which Db2 TCPIP configured.

   To catalog Db2 server instance running on port number 50000 on remote server "**myserver**" as node "db2node", enter the following command from a Db2 command line

   db2 => CATALOG TCPIP NODE db2node REMOTE myserver SERVER 50000

   For more details on catalog node, refer https://www.ibm.com/support/knowledgecenter/SSEPGG_11.1.0/com.ibm.db2.luw.qb.client.doc/doc/t0005621.html
4. If Db2 agent workstation is UNIX/Linux,

   - Create a user with node name, which is used in cataloging command

     Issue the command

     useradd -g <group> -m -d <home_dir> <user> -p <password>

     where

     – **<group>** represents a group for the DB2 UDB instance owners.

  - **`<user>`** represents a local **`username`** on client workstation. **`Userame`** must be same as node name by which the server instance has been cataloged on agent machine.

- Check the Db2 client instance name under which remote Db2 server instance is cataloged and assign the read, write, execute permissions of the newly created user's home directory to the owner of this instance. This step is necessary to make the client Db2 environment available for operations on remote node

- Issue the command

  ```
  chmod -R 775 /home/<nodename>
  ```

  where

  - **`<nodename>`** represents a local username of Db2 instance on client component

5. Catalog all the databases that you want to monitor on the client instance present at Db2 agent workstation.

   Issue the command in the Db2 CLP to catalog the database.

   ```
   CATALOG DATABASE <db_name> AS <db_alias> AT NODE <node_name>authentication
   server
   ```

   a. <db_name> represents server database name.

   b. <db_alias> represents local nickname for database at Db2 client.

   c. <node_name> represents a local nickname of Db2 instance on client component at which database is cataloged.

   To catalog a database called "sample" on catalog node "db2node" with alias as "dbAlias1", enter the following command from a Db2 prompt.

   ```
   db2 => CATALOG DATABASE sample AS dbAlias1 AT NODE db2node authentication
   server
   ```

Configuring Hadoop monitoring

You must configure the Monitoring Agent for Hadoop so that the agent can collect data of a Hadoop cluster that it monitors. The agent can monitor a single node Hadoop cluster and a multi-node Hadoop cluster.

Before you begin

Review the hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Hadoop agent.

Ensure that the following hosts can be resolved from the computer where the Hadoop agent is installed:

- All the Hadoop hosts that you want to configure, such as NameNode, ResourceManager, and so on
- Hadoop hosts with only the NodeManager role

For example, you can complete these steps to resolve hosts:

- Add the IP address, host name, and fully qualified domain name of all the Hadoop hosts to the `hosts` file that is available at the following path:
  - Windows `C:\Windows\System32\drivers\etc\hosts`
  - Linux AIX `/etc/hosts`
- Add the computer where the Hadoop agent is installed in the same domain as that of Hadoop hosts.

Remember: To monitor a Hadoop cluster that is secured with Kerberos SPNEGO-based authentication, ensure that all the hosts can be resolved from the computer where the Hadoop agent is installed.

About this task

The Hadoop agent is a single instance agent. You must configure the agent manually after it is installed. The Hadoop agent can be configured on Windows, Linux, and AIX systems.

Remember:

- For a single node Hadoop cluster, the same node performs all the roles, such as NameNode, ResourceManager, and secondary NameNode according to configuration of the Hadoop cluster. However, for a multi-node Hadoop cluster, different Hadoop nodes perform these roles.
- When you configure the agent, the agent automatically detects DataNodes and NodeManagers in the Hadoop cluster that is being monitored.
- In the Hadoop agent configuration, if SSL enabled option is selected as Yes, then for all the components that are installed and to be monitored on the Hadoop cluster such as Namenode, Secondary Namenode, Resource Manager, Standby Resource Managers, Job History Server, Journal Node, ZKFC, Datanodes , NodeManagers, Ambari Server, Cloudera Manager; SSL needs to be enabled for all these components.

When you upgrade from the socket - based agent (8.1.2 Fix Pack 2, or earlier) to the REST API - based agent (8.1.3, or later), complete the configuration steps that are specified in the subsequent topics. However, ensure that you specify the host names according to the following guidelines when you configure the agent.

- The host name of various daemon processes (NameNode, ResourceManger, and so on) that you specify must be the same (case and format) as the host names that are configured for the socket - based agent.
- The fully qualified domain name (FQDN) must be used when you specify a host name. For example, `hos1.ibm.com`. If the length of the FQDN exceeds 25 characters, specify only the short host name without the domain name. For example, if the FQDN of a host is *myhadoopclustersetupnode.ibm.com*, the short host name is `myhadoopclustersetupnode`.

After you configure the agent that is upgraded, and view data in the Cloud APM console, revert the changes that were made in the `hadoop-metrics2.properties` file for the Hadoop agent. For details, see "Upgrading your agents" on page 1111.

On Windows systems, you can run the Hadoop agent with a non-administrator user. However, such user requires a specific permission to view data in the dashboards. For information about how to grant this permission, see "Granting permission to non-admin users" on page 260.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the "Change history" on page 54.

Configuring the agent on Windows systems

You can configure the agent on Windows systems by using the **IBM Performance Management** window.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Monitoring Agent for Hadoop**.
3. Click **Configure agent**.

   ⚠ **Attention:** If **Configure agent** is disabled, click **Reconfigure**.

   The **Configure Monitoring Agent for Hadoop** window opens.
4. To monitor the Hadoop cluster with the Kerberos SPNEGO-based authentication enabled, complete these steps:
   a) Under **Is Kerberos SPNEGO-based authentication for HTTP based Hadoop services in Hadoop cluster enabled**, click **Yes**.

If you do not have Kerberos SPNEGO-based authentication to secure REST endpoints of HTTP based Hadoop services in the Hadoop cluster, click **No** and then the values for the **Realm name**, **KDC Hostname**, **SPNEGO principal name** and **SPNEGO keytab file** fields can be kept as blank.

b) In the **Realm name** field, enter the name of the Kerberos realm that is used to create service principals.
Usually, a realm name is the same as your domain name. For instance, if your computer is in the `tivoli.ibm.com` domain, the Kerberos realm name is `TIVOLI.IBM.COM` This name is case sensitive.

c) In the **KDC Hostname** field, enter the fully qualified domain name (FQDN) of the Key Distribution Center (KDC) host for the specified realm.

You can also specify the IP address of the KDC host instead of FQDN. In case of Active Directory KDC, Domain controller is the KDC host.

d) In the **SPNEGO principal name** field, enter the name of the Kerberos principal that is used to access SPNEGO authenticated REST endpoints of HTTP-based services.

The name is case sensitive, and the name format is HTTP/ *`fully_qualified_host_name@kerberos_realm`*

e) In the **SPNEGO keytab file** field, enter the name of the keytab file for the SPNEGO service with its full path, or click **Browse** and select it.

The keytab file contains the names of Kerberos service principals and keys. This file provides direct access to Hadoop services without requiring a password for each service. The file can be located at the following path: `etc/security/keytabs/`
Ensure that the SPNEGO principal name and the keytab file belong to the same host. For instance, if the principal name is *HTTP/abc.ibm.com@IBM.COM*, the keytab file that is used must belong to the *abc.ibm.com* host.
If the agent is installed on a remote computer, copy the keytab file of the principal to the remote computer at any path, and then specify this path in the **SPNEGO keytab file** field.

f) Click **Next**.

5. To monitor Hadoop cluster with HTTPS/SSL enabled, complete these steps:

a) Under **Is Hadoop Cluster SSL enabled**, click **Yes**

If you do not want the SSL enabled Hadoop cluster select **No** and then the values for the **TrustStore file path**, **TrustStore Password** fields can be kept as blank.

b) In **TrustStore file path**, select the TrustStore file stored at your local machine.

This file can be copied from the Hadoop cluster to your local machine and then used for configuration.

c) In **TrustStore Password**, enter the password you created while configuring the TrustStore file.

6. To specify values for the parameters of the Hadoop cluster, complete these steps:

a) In the **Unique Hadoop Cluster Name** field, enter the unique name for the Hadoop cluster indicating Hadoop version and flavor. The maximum character limit for this field is 12.

b) In the **NameNode Hostname** field, enter the host name of the node where the daemon process for NameNode runs.

c) In the **NameNode Port** field, enter the port number that is associated with the daemon process for NameNode. The default port number is 50070.

d) In the **ResourceManager Hostname** field, enter the host name of the node where the daemon process for ResourceManager runs.

e) In the **ResourceManager Port** field, enter the port number that is associated with the daemon process for ResourceManager. The default port number is 8088.

f) Optional: In the **JobHistoryServer Hostname** field, enter the host name of the node where the daemon process for JobHistoryServer runs.

g) Optional: In the **JobHistoryServer Port** field, enter the port number that is associated with the daemon process for JobHistoryServer. The default port number is 19888.

h) Optional: In the **Additional NameNode Hostname** field, enter the host name where the daemon process for a Standby NameNode or a Secondary NameNode runs.

i) Optional: In the **Additional NameNode Port** field, enter the port number that is associated with the daemon process for a Standby NameNode or a Secondary NameNode.

Remember: If the additional NameNode is a Standby NameNode, the default port number that is associated with the Standby NameNode daemon process is 50070. If the additional NameNode is a Secondary NameNode, the default port number that is associated with the Secondary NameNode daemon process is 50090.

j) Click **Test Connection** to verify connection to the specified host names and ports.

After you click **Test Connection**, an appropriate validation message is displayed when:

- The connection to the specified host names and ports is made or failed.
- A value for a host name is kept as blank.
- A value for a port is kept as blank.
- A non-integer value is specified for a port number.

Update the configuration values as suggested in the validation messages, and verify the connection again.

k) Optional: To add Standby ResourceManagers in the Hadoop cluster, click **Yes** under **Standby ResourceManager (s) in Hadoop Cluster**.

You are prompted to add the details of Standby ResourceManagers later.

l) Optional: To monitor Hadoop services in the Hadoop cluster that is managed by Apache Ambari, click **Yes** under **Monitoring of Hadoop services for Ambari based Hadoop installations**, and then click **Next**.

m) Optional: To monitor Cloudera Manager services in the Cloudera Hadoop cluster, click **Yes** under **Monitoring of Cloudera Manager services for Cloudera Hadoop installations**, and then click **Next**.

7. Optional: To specify the details of the Ambari server for monitoring Hadoop services, complete the following steps:

a) In the **Ambari server Hostname** field, enter the host name where the Ambari server runs.

b) In the **Ambari server Port** field, enter the port number that is associated with the Ambari server. The default port number is 8080.

c) In the **Username of Ambari user** field, enter the name of the Ambari user.

d) In the **Password of Ambari user** field, enter the password of the Ambari user.

e) Click **Next**.

8. Optional: To specify the details of the Cloudera Manager server for monitoring Cloudera Manager services, complete the following steps:

a) In the **Cloudera Manager server Hostname** field, enter the host name where the Cloudera Manager server runs.

b) In the **Cloudera Manager server Port** field, enter the port number that is associated with the Cloudera Manager server.

The default port number for HTTP based Cloudera Manager server is 7180.

c) In the **Username of Cloudera Manager server** user field, enter the name of the Cloudera Manager server's user.

d) In the **Password of Cloudera Manager server user** field, enter the password of the Cloudera Manager server's user.

e) Click **Next**.

9. To specify values for the Java parameters, complete these steps:

a) From the **Java trace level** list, select a value for the trace level that is used by Java providers.

b) Optional: In the **JVM arguments** field, specify a list of arguments for the Java virtual machine.

The list of arguments must be compatible with the version of Java that is installed along with the agent.

c) Click **Next**.

10. Optional: To add Standby ResourceManagers, complete the following steps:

    a) Click **New**.

    b) In the **Standby ResourceManager Hostname** field, enter the host name of the node where the daemon process for Standby ResourceManager runs.

    c) In the **Standby ResourceManager Port** field, enter the port number that is associated with the daemon process for Standby ResourceManager. The default port number is 8088.

    d) Click **Test Connection** to validate connection to the specified host name and the port number.

    After you click **Test Connection**, an appropriate validation message is displayed when:

    - The connection to the specified host names and ports is made or failed.
    - A value for a host name is kept as blank.
    - A value for a port is kept as blank.
    - A non-integer value is specified for a port number.

    Update the configuration values as suggested in the validation messages, and verify the connection again.

    e) Repeat steps a, b, and c to add more Standby ResourceManagers.

    If you want to remove any of the Standby ResourceManagers, click **Delete** corresponding to the Standby ResourceManager that you want to remove.

    f) Click **Next**.

11. In the **Class path for external jars** field, specify the class path for JAR files.

    This class path is added to the class path that is generated by the agent. You can keep this field blank.

12. Click **OK**.

    The specified configuration settings are saved.

13. Right-click **Monitoring Agent for Hadoop** and click **Start**.

What to do next

1. Enable the subnode events to view eventing thresholds of the Hadoop agent. For information about enabling subnode events, see “Configuring the dashboard for viewing Hadoop events” on page 260.
2. Log in to the Cloud APM console to view data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see “Starting the Cloud APM console” on page 957.

Configuring the agent on Linux and AIX systems

You run the configuration script and respond to prompts to configure the agent on Linux and AIX systems.

Procedure

1. On the command line, run the following command: `install_dir/bin/hadoop-agent.sh config`
   Where *install_dir* is the installation directory of Hadoop agent.

   The agent is installed at the following default installation directory:
   `/opt/ibm/apm/agent`

2. When the command line displays the following message, type 1 to continue with the configuration steps and press Enter.

   ```
   Edit "Monitoring Agent for Hadoop" setting? [1= yes, 2= No]
   ```

3. When the command line displays the following message, type 1 to specify values for monitoring the Hadoop cluster with the Kerberos SPNEGO-based authentication enabled, and press Enter.

Otherwise, type 2 and press Enter, and you can keep a blank value for the **Realm name**, **KDC Hostname**, **SPNEGO principal name**, and **SPNEGO keytab file** fields:

```
Is Kerberos SPNEGO-based authentication for HTTP based Hadoop services in
Hadoop cluster enabled\: [ 1=Yes, 2=No (default is: 2)
```

a) For the **Realm name** parameter, enter the name of the Kerberos realm that is used to create service principals.
Usually, a realm name is the same as your domain name. For instance, if your computer is in the `tivoli.ibm.com` domain, the Kerberos realm name is `TIVOLI.IBM.COM`. This name is case sensitive.

b) In the **KDC Hostname** field, enter the fully qualified domain name (FQDN) of the Key Distribution Center (KDC) host for the specified realm. You can also specify the IP address of the KDC host instead of FQDN. In case of Active Directory KDC, Domain controller is the KDC host

c) For the **SPNEGO principal name** parameter, enter the name of the Kerberos principal that is used to access SPNEGO authenticated REST endpoints of HTTP-based services.

The name is case sensitive, and the name format is HTTP/ *`fully_qualified_host_name@kerberos_realm`*

d) For the **SPNEGO keytab file** parameter, enter the name of the keytab file for the SPNEGO service with its full path.

The keytab file contains the names of Kerberos service principals and keys. This file provides direct access to Hadoop services without requiring a password for each service. The file can be located at the following path: `etc/security/keytabs/`
Ensure that the SPNEGO principal name and the keytab file belong to the same host. For instance, if the principal name is *HTTP/abc.ibm.com@IBM.COM*, the keytab file that is used must belong to the *abc.ibm.com* host.
If the agent is installed on a remote computer, copy the keytab file of the principal to the remote computer at any path, and then specify this path for the **SPNEGO keytab file** parameter.

4. When the command line displays the following message, type 1 to specify values for monitoring the Hadoop cluster with the SSL enabled, and press **Enter**. Otherwise, type 2 and press **Enter**, and you can keep a blank value for the **TrustStore file path** and **TrustStore Password** fields:

```
Is Hadoop Cluster SSL enabled [ 1=Yes, 2=No (default is: 2)
```

a) In **`TrustStore file path`**, specify the path of TrustStore file stored at your local machine.

This file can be copied from the Hadoop cluster to your local machine and then used for configuration.

b) In **`TrustStore Password`**, specify the password you created while configuring the TrustStore file.

5. When you are prompted to enter the details of the Hadoop cluster, specify an appropriate value for each of the following parameters, and press Enter.

a) In the **Unique Hadoop Cluster Name**, specify the unique name for the Hadoop cluster indicating Hadoop version and flavor. The maximum character limit for this field is 12.

b) For the **`NameNode Hostname`** parameter, specify the host name of the node where the daemon process for NameNode runs, and press Enter.

Attention: If you press Enter without specifying a host name, you are prompted to enter the host name.

c) For the **`NameNode Port`** parameter, specify the port number that is associated with the daemon process for NameNode, and press Enter. The default port number is 50070.

d) For the **`ResourceManager Hostname`** parameter, specify the host name of the node where the daemon process for ResourceManager runs, and press Enter.

Attention: If you press Enter without specifying a host name, you are prompted to enter the host name.

e) For the **`ResourceManager Port`** parameter, enter the port number that is associated with the daemon process for ResourceManager. The default port number is 8088.

6. Optional: When you are prompted to add the details of the following parameters of the Hadoop cluster, accept the default value or specify an appropriate value for each of the following parameters, and press Enter:

   a) For the **`JobHistoryServer Hostname`** parameter, enter the host name of the node where the daemon process for JobHistoryServer runs.

   b) For the **`JobHistoryServer Port`** parameter, enter the port number that is associated with the daemon process for JobHistoryServer. The default port number is 19888.

   c) For the **`Additional NameNode Hostname`** parameter, enter the host name of the node where the daemon process for a Secondary or a Standby NameNode runs.

   d) For the **`Additional NameNode Port`** parameter, enter the port number that is associated with the daemon process for a Secondary or a Standby NameNode. The default port number for a Secondary NameNode is 50090. For a Standby NameNode, the default port number is 50070.

7. Optional: When the command line displays the following message, enter 1 to add details of Standby ResourceMangers for high-availability cluster, and press Enter.
   ```
   Standby ResourceManager(s) in Hadoop Cluster [ 1=Yes, 2=No ] (default is:
   2):
   ```

8. When the command line displays the following message, specify 1 and press Enter to monitor Hadoop services in the Hadoop cluster that is managed by Ambari:
   ```
   Monitoring of Hadoop services for Ambari based Hadoop installations
   [ 1=Yes, 2=No ] (default is: 2):
   ```
   Otherwise, retain the default value of 2 and press Enter. If you enable the monitoring of Hadoop services, specify a value for each of the following parameters of Ambari server, and press Enter:

   a) For the **Ambari server Hostname** parameter, enter the host name where the Ambari server runs.

   b) For the **Ambari server Port** parameter, enter the port number that is associated with the Ambari server.
   The default port number is 8080.

   c) For the **Username of Ambari user** parameter, enter the name of the Ambari user.

   d) For the **Password of Ambari user** parameter, enter the password of the Ambari user.

9. When the command line displays the following message, specify 1 and press Enter to monitor Cloudera Manager services in the Hadoop cluster:
   ```
   Monitoring of Cloudera Manager services for Cloudera Hadoop installations
   [ 1=Yes, 2=No ] (default is: 2):
   ```
   Otherwise, retain the default value of 2 and press Enter. If you enable the monitoring of Cloudera Manager services, specify a value for each of the following parameters of Cloudera Manager server, and press Enter:

   a) For the **`Cloudera Manager server Hostname`** parameter, enter the host name where the Cloudera Manager server runs.

   b) For the **`Cloudera Manager server Port`** parameter, enter the port number that is associated with the Cloudera Manager server.

      The default port number for HTTP based Cloudera Manager server is 7180.

   c) For the **`Username of Cloudera Manager server`** user parameter, enter the name of the Cloudera Manager server's user.

   d) For the **`Password of Cloudera Manager server`** user parameter, enter the password of the Cloudera Manager server's user

10. When the command line displays the following message, select the appropriate Java trace level and press Enter:
    ```
    This parameter allows you to specify the trace level used by the Java
    providers Java trace level [ 1=Off, 2=Error, 3=Warning, 4=Information,
    5=Minimum Debug, 6=Medium Debug, 7=Maximum Debug, 8=All ] (default is: 2)
    ```

11. Optional: When the command line displays the following message, specify the arguments for the Java virtual machine, and press Enter. The list of arguments must be compatible with the version of Java that is installed along with the agent.

```
This parameter allows you to specify an optional list of arguments to the
java virtual machine JVM arguments (default is:)
```

12. Optional: When the command line displays the following message, enter 1 to add the following details of Standby ResourceManagers, and press Enter:
    `Edit "Hadoop High Availability(HA) Cluster with Standby ResourceManagers" settings, [1=Add, 2=Edit, 3=Del, 4=Next, 5=Exit] (default is: 5): 1`
    a) For the **`Standby ResourceManager Hostname`** parameter, enter the host name of the node where the daemon process for Standby ResourceManger runs.
    b) For **`Standby ResourceManager Port`**, enter the port number that is associated with the daemon process for Standby ResourceManager. The default port number is 8088.
    c) When you are prompted, enter 1 to add more Standby ResourceManagers, and repeat steps a and b, or enter 5 to go to the next step.
    - To edit the configuration settings of a specific Standby ResourceManager, type 4 and press Enter until you see the host name of the required Standby ResourceManager.
    - To remove a Standby ResourceManager, type 3 and press Enter after you see the host name of the Standby ResourceManger that you want to remove.
13. When you are prompted, enter the class path for the JAR files that the Java API data provider requires, and press Enter.

    The specified configuration values are saved, and a confirmation message is displayed.
14. Run the following command to start the agent: `*install_dir*/bin/hadoop-agent.sh start`

What to do next

1. Enable the subnode events to view eventing thresholds of the Hadoop agent. For information about enabling subnode events, see "Configuring the dashboard for viewing Hadoop events" on page 260.
2. Log in to the Cloud APM console to view data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. For some parameters, the default values are provided in comments. You can specify different values for these parameters, and remove the comment tags that are placed at the beginning of the parameters.

About this task

You can use the silent response file to configure the Hadoop agent on Linux, AIX, and Windows systems.

Procedure

1. Open the silent response file that is available at this path: *install_dir*`\samples\hadoop_silent_config.txt`
2. In the response file, complete the following steps:
    a) When you want to monitor the Hadoop Cluster that is enabled for Kerberos SPNEGO-based authentication, specify `yes` and enter values for the following parameters:

    ```
    HADOOP_REALM_NAME
    HADOOP_KDC_HOSTNAME
    HADOOP_PRINCIPAL_NAME
    HADOOP_SPNEGO_KEYTAB
    ```

    b) When you want to monitor the Hadoop Cluster that is SSL enabled, type `yes` and enter values for the following parameters:

    ```
    HADOOP_TRUSTSTORE_PATH
    HADOOP_TRUSTSTORE_PASSWORD
    ```

c) Enter values for the following parameters of Cluster, NameNode (NN), ResourceManager (RM), and Job History Server (JHS):

```
HADOOP_CLUSTER_NAME (optional)
HADOOP_NN_HOSTNAME
HADOOP_NN_PORT
HADOOP_RM_HOSTNAME
HADOOP_RM_PORT
HADOOP_JHS_HOSTNAME (optional)
HADOOP_JHS_PORT (optional)
```

d) Optional: For the **HADOOP_ADDITIONAL_NN_HOSTNAME** parameter, specify the host name of the Standby or Secondary NameNode.

e) Optional: For the **HADOOP_ADDITIONAL_NN_PORT** parameter, specify the port number of the Standby or Secondary NameNode.

Remember: If the additional NameNode is a Standby NameNode, the default port number that is associated with the Standby NameNode daemon process is 50070. If the additional NameNode is a Secondary NameNode, the default port number that is associated with the Secondary NameNode daemon process is 50090.

f) Optional: For the **Hadoop_SRM** parameter, specify Yes to add Standby ResourceManagers for a high-availability cluster, and go to step g.

g) Optional: To monitor Hadoop services in the Hadoop cluster that is managed by Ambari, enter values for each of the following parameters, and press Enter:

```
HADOOP_AMBARI=No
AMBARI_SERVER_HOSTNAME
AMBARI_SERVER_PORT
USERNAME_OF_AMBARI_USER
PASSWORD_OF_AMBARI_USER
```

h) Optional: To monitor services in Cloudera based Hadoop cluster that is managed by Cloudera Manager, enter values for each of the following parameters, and press Enter:

```
HADOOP_CMSERVICE=Yes
HADOOP_CM_HOSTNAME=clouderasn633
HADOOP_CM_PORT=7183
USERNAME_OF_CM=admin
PASSWORD_OF_CM_USER=admin
```

i) For the **JAVA_TRACE_LEVEL** parameter, specify the appropriate trace level.

j) Optional: For the **JAVA_JVM_ARGS** parameter, specify arguments for the Java™ virtual machine.

k) Optional: Add the host name and the port number of a Standby ResourceManager in the following format: HADOOP_SRM_PORT.*hadoop_srm_config_sec_1*=8088

Where, *hadoop_srm_config_sec_1* is the host name of the node where the daemon process for Standby ResourceManager runs, and 8088 is the default port number. To add more Standby ResourceManagers, add the host name and port number of other Standby ResourceManagers on new lines in the same format.

3. Save the response file, and run the following command:

Linux AIX *install_dir*/bin/hadoop-agent.sh config *install_dir*/samples/hadoop_silent_config.txt

Windows *install_dir*/bin/hadoop-agent.bat config *install_dir*/samples/hadoop_silent_config.txt

4. Start the agent:

Linux AIX Run the following command: *install_dir*\bin\hadoop-agent.sh start

Windows Right-click **Monitoring Agent for Hadoop** and then click **Start**.

What to do next

1. Enable the subnode events to view eventing thresholds of the Hadoop agent. For information about enabling subnode events, see "Configuring the dashboard for viewing Hadoop events" on page 260.
2. Log in to the Cloud APM console to view data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring the dashboard for viewing Hadoop events

You must configure the dashboard to enable the subnode events so that the **Events** tab can display Hadoop events.

About this task

The default value for **Enable Subnode Events** is false. Change this value to true for viewing Hadoop events.

Procedure

1. Open the Cloud APM console and go to **System Configuration**.
2. On the **Advanced Configuration** page, click **UI Integration** under **Configuration Categories**.
3. From the **Enable Subnode Events** list, select **True**.
4. Click **Save**.

Granting permission to non-admin users

On Windows systems, grant the *Debug program* permission to a non-admin user for running the Hadoop agent. This permission is required to view data in the Hadoop agent dashboards.

Procedure

Complete the following steps on the computer where the Hadoop agent is installed:

1. Click **Start > Control Panel > Administrative Tools**.
2. Double-click **Local Security Policy**.
3. In the Security Settings pane, expand **Local Policies** and click **User Rights Assignment**.
4. Right-click **Debug programs** and click **Properties**.
5. Click **Add User or Group**, and add the non-admin user name to which you want to grant this permission.
6. Click **OK**.

What to do next

Configure and run the Hadoop agent with the non-admin user.

Configuring HMC Base monitoring

The Monitoring Agent for HMC Base provides you with the capability to monitor the Hardware Management Console (HMC). The agent monitors the availability and health of the HMC resources: CPU, memory, storage, and network. The agent also reports on the HMC inventory and configuration of Power servers, CPU pools, and LPARs. The CPU utilization of the Power servers, LPARs, and pools are monitored by using HMC performance sample data.

Before you begin

Before you configure the HMC Base agent, you must complete the following tasks:

- Set up SSH connection between the system that is running the agent and the HMC. For more information, see "Setting up the SSH connection" on page 262.
- Prepare HMC SDK before you start the first agent instance. For more information, see "Preparing SDK for HMC" on page 263.

Procedure

- To configure the agent by editing the silent response file and running the script with no interaction, complete the following steps:

  1. Open the `hmc_base_silent_config.txt` file in a text editor:
     - AIX *install_dir*`/samples/hmc_base_silent_config.txt`.
  2. For **HMC Hostname**, you can specify the IP address or host name.
  3. For **HMC Username**, you must enter the logon user name for the HMC, for example, **HMC_USERNAME=** *hscroot*.

     Note: The logon user name that you assign to the HMC requires hscviewer authority at a minimum.
  4. For **HMC Password**, you must enter the password of the user.
  5. For **Maximum Number of Data Provider Log Files:**, you must specify the maximum number of data provider log files that are created. For example, **KPH_LOG_FILE_MAX_COUNT=***10*.
  6. For **Maximum Size in KB of Each Data Provider Log**, you must enter the maximum size in KB that a data provider log file might reach before a new log file is created, for example, **KPH_LOG_FILE_MAX_SIZE=** *5190*.
  7. For **Level of Detail in Data Provider Log**, you must enter the amount of detail that the data provider includes in the data provider log files, for example, **KPH_LOG_LEVEL=***Fine*. You must specify one of the following values:
     - 1= Off
     - 2=Severe
     - 3=Warning
     - 4=Info
     - 5=Fine
     - 6=Finer
     - 7=Finest
     - 8=All

     Important: The default value is 4.
  8. Save and close the `hmc_base_silent_config.txt` file, then enter: `./hmc_base-agent.sh config` *instance_name install_dir*`/samples/hmc_base_silent_config.txt` where *instance_name* is the name you want to give the instance and *install_dir* is the HMC Base agent installation directory. The default installation directory is `/opt/ibm/apm/agent`.

- AIX To configure the agent by responding to prompts, complete the following steps:

  1. Open the *install_dir*`/bin` directory, where *install_dir* is the installation directory for the HMC Base agent.
  2. To configure the HMC Base agent, run the following command: **./hmc_base-agent.sh config instance_name**.
  3. When prompted to **edit the Monitoring Agent for HMC Base settings**, press **Enter**. The default value is Yes.
  4. To enter the HMC configuration information, complete the following steps.
     a. When prompted for the **HMC Hostname**, type the host name or IP address and press **Enter**.
     b. When prompted for **HMC Username**, type the logon user name that is associated with the HMC, and press **Enter**.
  5. When prompted for **HMC Password**, type the password of the user.
  6. To enter the data provider information, complete the following steps:
     a. When prompted for the **maximum number of data provider log files**, type the amount of log files and press **Enter**.

The default maximum number of data provider log files is 10.

b. When prompted for the **`maximum size in KB of each data provider log`**, type the size and press **Enter**.

   The default maximum size in KB is 5190.

c. When prompted for the **`Level of detail in data provider log`**, type one of the following levels and press **Enter**:

   - 1= Off
   - 2=Severe
   - 3=Warning
   - 4=Info
   - 5=Fine
   - 6=Finer
   - 7=Finest
   - 8=All

What to do next

- To start the agent, enter: `./hmc_base-agent.sh start` *`InstanceName`*.
- Configure the HMC Console Server according to the instruction in "Configuring the HMC Console Server for monitoring Virtual I/O" on page 264 for monitoring Virtual I/O.
- Enable the CPU and memory utilization monitoring according to the instruction in "Enabling the CPU and memory utilization monitoring" on page 265.

Setting up the SSH connection

You must set up SSH connection between the system that is running the agent and the HMC for the agent to collect data.

About this task

The agent data provider collects data from the management console by running CLI commands over SSH. By default, the data provider waits up to 1 minute for a CLI command to finish running. After this time, the data provider closes the SSH session in which the CLI command is running, and none of the data for that command is available in agent data sets until the command runs successfully. The default path for the SSH command is `/usr/bin/ssh`. If you installed SSH in a different location, you must indicate the path by using the **KPH_SSH_PATH** environment variable.

Procedure

Use one of the following methods to set up SSH connection.

- Use the `setup_hmc_key.pl` script to set up the SSH connection.

  a) Log on to the server where the agent is installed.

  b) Open the `install_dir/aix526/ph/bin` directory, where `install_dir` is the installation directory for the HMC Base agent.

  c) Run the `perl setup_hmc_key.pl` command.

  d) Respond to prompts and provide the HMC host name or IP address; the HMC user name, which must have authority equivalent to `hscviewer` authority; and the password to create the key pair.

  e) After you create the key pair, test the connectivity by running a command such as `ssh hscroot@hmchost lshmc -V`.

If SSH is connecting to this HMC for the first time, add the HMC to the ssh known_hosts file by responding with yes to the following message:

```
The authenticity of host 'hmchost (3.3.333.333)' can't be established.
RSA key fingerprint is 4c:b4:26:27:38:f3:ec:58:01:92:26:f9:61:32:bb:4d.
Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added 'hmchost,3.3.333.333' (RSA) to the list of known hosts.
```

The agent can now use SSH to collect data from the HMC.

- Use ssh-keygen utility to generate keys and set up the SSH connection.

  a) Log on to the server where the agent is installed.

  b) Use the ssh-keygen utility to generate public and private keys with no paraphrase.
  For example, the following command generates a set of public and private keys:

  ```
  ssh-keygen -t rsa -f /.ssh/id-rsa
  ```

  Press Enter when prompted for a paraphrase. The public key that is generated is stored in the /.ssh/id-rsa.pub file. The private key is stored in the /.ssh/id-rsa file.

  c) Transfer the file that contains the public key to the HMC computer by using utilities such as scp.

  d) On the HMC computer, append the public key file to the collection of keys that are stored on the HMC.

  The stored keys are in the /.ssh/authorized_keys2 file.

  e) Add the host name and key for the HMC in the known_hosts file.

  This file is in the /.ssh directory.

  a. Run the ssh "user"@"hmc_hostname" -i "private_keyfile" date command.

  b. Enter yes when prompted to cache the keys. This command adds the entry to the known_hosts file for future connections.

  f) Run the ssh "user"@"hmc_hostname" date command.

  If the date is returned with no password prompt, the SSH keys were successfully set up.

Preparing SDK for HMC

You must prepare SDK for HMC before you start the agent instance for the first time.

About this task

Before you start your first agent instance, you must prepare the corresponding version of SDK for your HMC. After the preparation completes, you do not need to repeat this task for other HMC Base agent instance that you create for the HMC of the same version. To monitor another version of HMC, repeat this tasks to re-prepare SDK.

Procedure

- From the *agent_dir*/aix526/ph/bin directory, run the **prepareSDK.sh** script tool to automatically prepare SDK from HMC.

  - If you see the SDK is ready for HMC message, the preparation is completed.
  - If you do not see the SDK is ready for HMC message, you can manually prepare the SDK for HMC.

    For HMC V8.5.0, complete the following steps:

    1. Use a browser to download SDK from HMC directly with the following URL:

       ```
       https://HMC_IP:12443/rest/api/web/sdk
       ```

       When prompted, enter the user name and password of the hscroot account. The SDK file name is in the format of pmc_sdk_*.zip.

2. . Unzip the SDK zip file, and go to the `IBM HMC REST Web Services SDK Runtime/lib/ibm3` directory.
3. If it doesn't already exist, create a `<agent_dir>/aix526/ph/lib/`*`my_hmc_version`* sub-directory, where *my_hmc_version* is the version of your HMC environment, for example, 8502. To determine the version of your HMC environment, run the following command:

   ```
   ssh hscroot@<HMC_IP> 'lshmc -v' | grep RM |
   awk -FR '{print $3}' | tr -d '.'
   ```

4. Copy all the `.jar` files in the `IBM HMC REST Web Services SDK Runtime/lib/ibm3` folder of HMC SDK to the *`agent_dir`*`/aix526/ph/lib/`*`HMC_version`* directory.

For HMC V8.6.0 or V8.7.0, complete the following steps:

1. Use a browser to download SDK from HMC directly with the following URL:

   ```
   https://HMC_IP:12443/rest/api/web/sdk
   ```

   When prompted, enter the user name and password of the `hscroot` account. The SDK file name is in the format of pmc`-rest-sdk*.zip`.
2. Unzip the SDK zip file, and go to the `lib` sub-directory.
3. If it doesn't already exist, create a `<agent_dir>/aix526/ph/lib/`*`my_hmc_version`* sub-directory, where *my_hmc_version* is the version of your HMC environment, for example, 8602, or 87012. To determine the version of your HMC environment, run the following command:

   ```
   ssh hscroot@<HMC_IP> 'lshmc -v' | grep RM |
   awk -FR '{print $3}' | tr -d '.'
   ```

4. Copy all the `.jar` files in the `lib` folder of the HMC SDK to the *`agent_dir`*`/aix526/ph/lib/`*`my_hmc_version`* directory.

Results

You successfully prepared SDK for HMC.

What to do next

Configure the HMC Base agent according to the instructions in "Configuring HMC Base monitoring" on page 260.

Configuring the HMC Console Server for monitoring Virtual I/O

Before the HMC Base agent can monitor the status of Virtual I/O, you must configure the HMC Console Server.

Procedure

Follow the steps to configure the HMC Console Server as the prerequisites for the HMC Base agent to monitor Virtual I/O.

- Enable the PMC function of HMC console server and virtual I/O servers.
  a) Log on the HMC Console Server by using the browser in classic mode.

  ```
  https://hmc_hostname
  ```

  b) Click **HMC Management** > **Change Performance Monitoring Settings**.

  The **Change Performance Monitoring Settings** window is displayed.

  c) In the **Performance Monitoring Data Collection for Manage Servers** section, turn on the **Collection** function for the corresponding servers.

  d) Click each Virtual I/O Server to show the **Partition Properties** window for that server.

e) Under the **General** tab, ensure that the check box of the **Allow performance information collection** option is selected.

Click **OK** to save the settings.

After several minutes, you can see the network and storage traffic of corresponding servers in the **Performance Monitoring** page.

- Ensure that the HMC user for the HMC Base agent has correct privilege.

  a) When you add or edit the user, ensure that the user has the **hmcviewer** role, and that the **AllSystemResource** option for this user is enabled.

  b) In **User Properties** window, enable the **Allow remote access via the web** option.

Enabling the CPU and memory utilization monitoring

If the CPU and memory utilization data collection is disabled, the CPU and memory utilization data of each power server is not displayed in the UI.

Procedure

Use one of the following methods to enable the CPU and memory utilization monitoring.

- Enable the CPU and memory utilization monitoring by running the following HMC management command `chlparutil`.

```
chlparutil-r config -m <CECname> -s <the sample rate in seconds, always 60>
```

- Enable the CPU and memory utilization monitoring on the HMC console server.

  a) Log on the HMC console server with classic mode.

  b) Click **Servers** node on the navigation tree.

  c) Select the server and go to **Operations** > **Utilization Data** > **Change the Sample rate**.

  d) Set a sampling rate.
  The sampling rate is disabled by default. You can set the rate with appropriate values, for example, `30 minutes`.

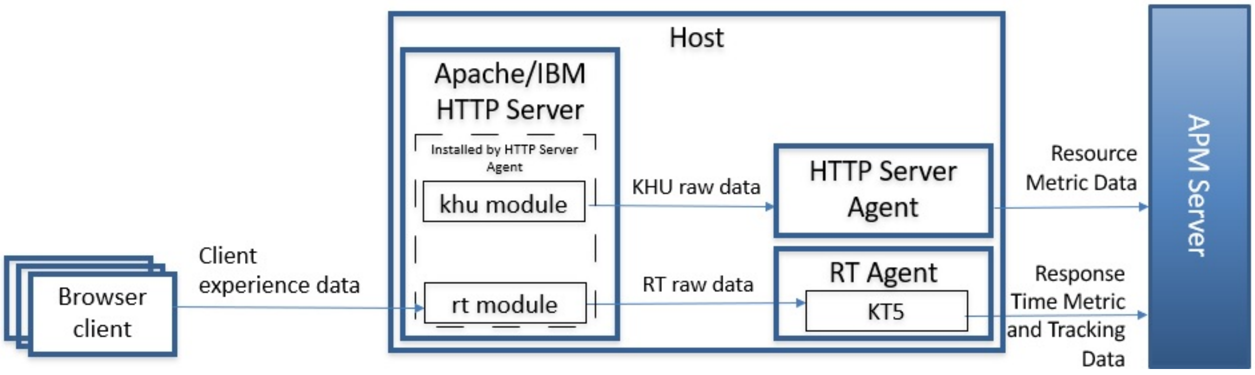
Configuring HTTP Server monitoring

Configure the HTTP Server monitoring to enable the monitoring solution of your HTTP Server.

Overview of HTTP Server monitoring

The HTTP Server monitoring solution provides monitoring capabilities that consist of two modules: **KHU** module and **RT (Response Time)** module.

Architecture



Configuring the KHU module for HTTP Server monitoring

The KHU module gets traditional resource and performance metrics. **KHU** is the agent code (unique code among all agents) for HTTP Server agent and the KHU module talks with HTTP Server agent process.

Before you begin

It is strongly suggested that install the HTTP Server agent when HTTP Server process is running because the agent installation can discover the running HTTP Server process and generate needed reference file.

There are two files involved in the configuration of the HTTP Server agent:

- HTTP Server configuration file: It comes with every HTTP server.
- HTTP Server agent data collector configuration file: It is also called reference file that is generated by HTTP Server agent when it discovers the running HTTP Server.

What you need to do is to include the reference file in the HTTP Server configuration file. And then restart HTTP Server. For details, see the Procedure section.

The HTTP Server configuration file

Each HTTP server has a configuration file by default: `http_server_install_dir/conf/httpd.conf`. In some environments, this file name might be customized. Check the exact file name with the HTTP Server administrator.

The HTTP Server agent data collector configuration file

When the HTTP Server agent discovers the HTTP server, it generates a data collector configuration file in the *install_dir*/tmp/khu directory.

If multiple HTTP Servers are discovered, multiple corresponding HTTP Server agent configuration files get generated.

The HTTP Server agent configuration file name has the format:
khu.*http_server_install_dir.conf.httpd*.conf
The first part of the agent configuration file name is khu, in which khu is the HTTP server agent code. The remaining part is the discovered HTTP server configuration file name with full path, in which / is replaced by . . For example, possible file names are as follows:

- Linux AIX `khu.usr.local.apache24.conf.httpd.conf`
- Windows `khu.C.Program Files.IBM.HTTPServer.conf.httpd.conf`

The following elements are inside the HTTP Server agent data collector configuration file :

- Details of the path of the httpd.conf file that the HTTP Server uses, for example, `KhuShmemPath "/IBM/HTTPServer/conf/httpd.conf"`.
- Location of the library to load
- Permissions that are associated with the shared memory

For more information about the configuration file, see "Reference: HTTP Server agent configuration file samples" on page 268.

Note: Normally, you do not need to modify this configuration file. Just locate and record the location.

About this task

After you install the HTTP Server agent, enable the HTTP Server agent for data collection.

Note: You may also need to do this process on the following special occasions:

- After you upgrade to HTTP Server agent version 1.0.0.4, the new alias causes the existing HTTP Server agent nodes to become offline in the Cloud APM console.
- After you upgrade from version 1.0.0.4, the existing HTTP Server agent nodes can become offline in the Cloud APM console. This can occur if you have multiple HTTP server instances with similar agent configuration file names, for example, `httpd` and `httpd01`.

Important:

- The netstat command-line network utility tool is needed for the HTTP Server agent to successfully discover the running HTTP server.
- To solve the agent node offline problem that happens after the upgrade, you must add the new HTTP server instance in the Cloud APM console after you complete this task.

Procedure

1. To activate data collection, you must reference the data collector configuration file in the HTTP server configuration file by using the `Include` statement. Append the following statement to the end of the HTTP Server configuration file:

   `Include “agent_install_dir/tmp/khu/khu.http_server_install_dir.conf.httpd.conf“`

   For example,

   Linux AIX If you have an IBM HTTP Server that is installed in the `/opt/IBM/HTTPServer` directory and the data collector configuration file name with full path is like the following: `/opt/ibm/apm/agent/tmp/khu/khu.`*`opt.IBM.HTTPServer.conf.httpd`*`.conf`. Append the following statement to the `/opt/IBM/HTTPServer/conf/httpd.conf` HTTP server configuration file:

   ```
   Include “/opt/ibm/apm/agent/tmp/khu/khu.opt.IBM.HTTPServer.conf.httpd.conf“
   ```

   Windows If you have an IBM® HTTP Server that is installed in the `C:\ProgramFiles\IBM\HTTPServer` directory and the data collector configuration file name with full path is like the following: `C:\IBM\APM\tmp\khu\khu.C.Program Files.IBM.HTTPServer.conf.httpd.conf`. Append the following statement to the `C:\Program Files\IBM\HTTPServer\conf\httpd.conf` HTTP server configuration file:

   ```
   Include “C:\IBM\APM\tmp\khu\khu.C.Program Files.IBM.HTTPServer.conf.httpd.conf“
   ```

2. Change to the following directory:

   `HTTP_server_installation_directory``/bin`

3. Restart the HTTP Server. For example:

   Linux AIX

   ```
   ./apachectl -k stop
   ./apachectl -k start
   ```

   Windows

   ```
   httpd.exe -k stop
   httpd.exe -k start
   ```

Results

You have successfully configured the agent.

What to do next

Now, you can verify the HTTP Server agent data is displayed in the Cloud APM console. For instructions on how to start the Cloud APM console, see Starting the Cloud APM console. For information about using the Applications editor, see Managing applications.

Note: If there is no traffic in the HTTP server, you will not see data in the Cloud APM console.

Configuring the RT module for HTTP Server monitoring

The RT module gets transaction response time and talks with Response Time Monitoring agent process.

The RT module is already loaded when you complete the steps of configuring KHU module. Normally, you do not need to do anything special for RT module. You only need to install the Response Time Monitoring agent to make them work together.

Note:

- The RT module together with the KHU module are installed by HTTP Server agent and loaded by HTTP Server after a successful configuration and HTTP Server restart. This gives the convenience that customer can load modules at one time, not multiple times.
- The RT module monitors all ports for HTTP and HTTPS requests on HTTP server, and no special configuration is needed.
- When there is no Response Time Monitoring agent running, although the RT module is loaded, it will not work.
- Remember to install the Response Time Monitoring after HTTP Server agent because the Response Time Monitoring agent can detect the existence of HTTP Server agent and switch itself to work with HTTP Server RT module.
- Install Response Time Monitoring agent on the same host with HTTP Server.

For more information, see "Configuring Response Time Monitoring" on page 677.

Reference: HTTP Server agent configuration file samples

There are two files involved in the configuration of the HTTP Server agent. They are the HTTP Server agent data collector configuration file and the HTTP server configuration file. A sample for the Instance alias mapping file is also provided to help explain how alias works.

HTTP Server agent data collector file samples

For IBM HTTP Server version 8 and later, 64-bit, the HTTP Server agent data collector configuration file contains this information:

```
#
# Settings for Monitoring Agent for HTTP Server module.
#

LoadModule khu_module "/tmp/ihs/lx8266/hu/lib/khuapache22dc_64.so"

<IfModule mod_khu.c>
   KhuShmemPerm 660
   KhuShmemPath "/opt/IBM/IHS/conf/httpd.conf"
   KhuCpsPath "/tmp/ihs/tmp/khu/khu_cps.properties"
</IfModule>

Alias /khu "/tmp/ihs/lx8266/hu/etc"
<Directory "/tmp/ihs/lx8266/hu/etc">
  Order deny,allow
  Allow from all
  #Require all granted
</Directory>

LoadModule wrt_module /tmp/ihs/lx8266/hu/lib/mod_wrt_ap22_64.so
WrtOriginID HU:tivvm09_httpd:HUS
```

For IBM HTTP Server version 7, 32-bit, the configuration file contains this information:

```
#
# Settings for Monitoring Agent for HTTP Server module.
#

LoadModule khu_module "/tmp/ihs/lx8266/hu/lib/khuapache22dc_32.so"

<IfModule mod_khu.c>
```

```
 KhuShmemPerm 660
 KhuShmemPath "/opt/IBM/HTTPServer/conf/httpd.conf"
 KhuCpsPath "/tmp/ihs/tmp/khu/khu_cps.properties"
</IfModule>

Alias /khu "/tmp/ihs/lx8266/hu/etc"
<Directory "/tmp/ihs/lx8266/hu/etc">
  Order deny,allow
  Allow from all
  #Require all granted
</Directory>

LoadModule wrt_module /tmp/ihs/lx8266/hu/lib/mod_wrt_ap22.so
WrtOriginID HU:linux_httpd:HUS
```

For Apache version 2.4, 64-bit, the HTTP Server agent configuration file contains this information:

```
#
# Settings for Monitoring Agent for HTTP Server module.
#

LoadModule khu_module "/tmp/ihs/lx8266/hu/lib/khuapache24dc_64.so"

<IfModule mod_khu.c>
 KhuShmemPerm 660
 KhuShmemPath "/usr/local/apache24/conf/httpd.conf"
</IfModule>

Alias /khu "/tmp/ihs/lx8266/hu/etc"
<Directory "/tmp/ihs/lx8266/hu/etc">
  Order deny,allow
  Allow from all
  Require all granted
</Directory>

LoadModule wrt_module /tmp/ihs/lx8266/hu/lib/mod_wrt_ap24_64.so
WrtOriginID HU:linux-tzsi_httpd:HUS
```

Instance alias mapping file sample

```
# Monitoring Agent for HTTP Server instance alias mapping
# INSTANCE: auto discovered by agent. Please do NOT modify.
# ALIAS: alias name for the instance. The name will be displayed in APM UI dashboard. It
must be unique
# among all instances and it must be less than 10 characters and consist of only
alphanumeric characters.
#
INSTANCE.1=/usr/local/apache24/conf/httpd.conf
ALIAS.1=httpd

INSTANCE.1=/usr/local/apache24/conf/admin.conf
ALIAS.1=admin
```

Configuring IBM Cloud monitoring

The Monitoring Agent for IBM Cloud collects virtual machine inventory and metrics from your IBM Cloud (SoftLayer) account. Use the IBM Cloud agent to track how many virtual devices you have configured and running in IBM Cloud. You can see what resources are allocated to each virtual device in the detailed dashboard page, which also shows information like the data center a device is located in, the operating system, and the projected public network bandwidth for the month.

Before you begin

- Read the entire "Configuring IBM Cloud monitoring" on page 269 topic to determine what is needed to complete the configuration.
- The directions here are for the most current release of the agent, except as indicated.

- Make sure that the system requirements for the IBM Cloud agent are met in your environment. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the IBM Cloud agent.
- Ensure that the following information is available:
  - A username for a user with at least Auditor permissions.
  - The API Key for IBM Cloud for that associated user.

About this task

The IBM Cloud agent is both a multiple instance agent and also a subnode agent. After you configure agent instances, you must start each agent instance manually.

Procedure

1. Configure the agent on Windows systems with the **IBM Performance Management** window or the silent response file.
   - "Configuring the agent on Windows systems" on page 270.
   - "Configuring the agent by using the silent response file" on page 271.
2. Configure the agent on Linux systems with the script that prompts for responses or the silent response file.
   - "Configuring the agent by responding to prompts" on page 271.
   - "Configuring the agent by using the silent response file" on page 271.

What to do next

In the Cloud APM console, go to your Application Performance Dashboard to view the data that was collected. For more information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you are unable to view the data in the agent dashboards, first check the server connection logs and then the data provider logs. The default paths to these logs are listed here:

- Linux `/opt/ibm/apm/agent/logs`
- Windows `C:\IBM\APM\TMAITM6_x64\logs`

For help with troubleshooting, see the Cloud Application Performance Management Forum.

Configuring the agent on Windows systems

You can configure the IBM Cloud agent on Windows operating systems by using the IBM Cloud Application Performance Management window. After you update the configuration values, you must start the agent to save the updated values.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Cloud Application Performance Management**.
2. In the **IBM Performance Management** window, right-click the **Monitoring Agent for IBM Cloud** template, and then click **Configure agent**.

   Remember: After you configure an agent instance for the first time, the **Configure agent** option is unavailable. To configure the agent instance again, right-click on it and then click **Reconfigure...**.
3. Enter a unique instance name then click **OK**. Use only Latin letters, Arabic numerals, and the hyphen-minus character in the instance name. Example, `icloud-inst`.
4. Click **Next** on the agent instance name window.
5. Press **New** and enter IBM Cloud SoftLayer username and API Key settings, then click **Next**.

6. Click **OK** to complete the configuration.
7. In the IBM Cloud Application Performance Management window, right-click the instance that you configured, and then click **Start**.

Configuring the agent by responding to prompts

After installation of the IBM Cloud agent, you must configure it before you start the agent. If the IBM Cloud agent is installed on a local Linux computer, you can follow these instructions to configure it interactively through command-line prompts.

About this task

Remember: If you are reconfiguring a configured agent instance, the value that is set in the last configuration is displayed for each setting. If you want to clear an existing value, press the space key when the setting is displayed.

Procedure

Follow these steps to configure the IBM Cloud agent by running a script and responding to prompts.

1. Run the following command:

   ```
   install_dir/bin/ibm_cloud-agent.sh config instance_name
   ```

   where *install_dir* is the path where the agent is installed and *instance_name* is the name that you want to give to the agent instance.

   Example

   ```
   /opt/ibm/apm/agent/bin/ibm_cloud-agent.sh config icloud-inst
   ```

2. Respond to the prompts to set configuration values for the agent.

   See "Configuration parameters for the IBM Cloud agent" on page 272 for an explanation of each of the configuration parameters.

3. Run the following command to start the agent:

   ```
   install_dir/bin/ibm_cloud-agent.sh start instance_name
   ```

   where *install_dir* is the path where the agent is installed and *instance_name* is the name of the agent instance.

   Example

   ```
   /opt/ibm/apm/agent/bin/ibm_cloud-agent.sh start icloud-inst
   ```

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

The silent response file contains the agent configuration parameters with default values that are defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

- Configure the IBM Cloud agent in the silent mode:

  a) Open the ibm_cloud_silent_config.txt file at one of the following paths in a text editor.

  – Linux *install_dir*/samples/ibm_cloud_silent_config.txt

    Example, /opt/ibm/apm/agent/samples/ibm_cloud_silent_config.txt

  – Windows *install_dir*\samples\ibm_cloud_silent_config.txt

    Example, C:\IBM\APM\samples\ibm_cloud_silent_config.txt

  where *install_dir* is the path where the agent is installed.

  b) In the ibm_cloud_silent_config.txt file, specify values for all mandatory parameters and modify the default values of other parameters as needed.

  See "Configuration parameters for the IBM Cloud agent" on page 272 for an explanation of each of the configuration parameters.

  c) Save and close the ibm_cloud_silent_config.txt file, and run the following command:

  – Linux *install_dir*/bin/ibm_cloud-agent.sh config **instance_name** *install_dir*/samples/ibm_cloud_silent_config.txt

    Example, **/opt/ibm/apm/agent/bin/ibm_cloud-agent.sh config icloud-inst /opt/ibm/apm/agent/samples/ibm_cloud_silent_config.txt**

  – Windows *install_dir*\bin\ibm_cloud-agent.bat config **instance_name** *install_dir*\samples\ibm_cloud_silent_config.txt

    Example, **C:\IBM\APM\bin\ibm_cloud-agent.bat config icloud-inst C:\IBM\APM\samples\ibm_cloud_silent_config.txt**

  where *install_dir* is the path where the agent is installed and *instance_name* is the name that you want to give to the agent instance.

  Important: Ensure that you include the absolute path to the silent response file. Otherwise, the agent data is not shown in the dashboards.

  d) Run the following command to start the agent:

  – Linux *install_dir*/bin/ibm_cloud-agent.sh start **instance_name**

    Example, **/opt/ibm/apm/agent/bin/ibm_cloud-agent.sh start icloud-inst**

  – Windows *install_dir*\bin\ibm_cloud-agent.bat start **instance_name**

    Example, **C:\IBM\APM\bin\ibm_cloud-agent.bat start icloud-inst**

  where *install_dir* is the path where the agent is installed and *instance_name* is the name of the agent instance.

Configuration parameters for the IBM Cloud agent

The configuration parameters for the IBM Cloud agent are displayed in a table.

1. IBM Cloud Configuration - Settings to monitor IBM Cloud instances remotely. Instances are automatically discovered for the API key that you want to configure.

Table 23. IBM Cloud Configuration

Parameter name	Description	Silent configuration file parameter name
Username	The username for IBM SoftLayer account that is used to retrieve metrics from the IBM Cloud API.	**KFS_USERNAME**

Table 23. IBM Cloud Configuration (continued)

Parameter name	Description	Silent configuration file parameter name
API Key	The user-specific API Key that is required to complete the authentication. API Keys are generated and can be retrieved from the IBM SoftLayer Customer Portal.	**KFS_API_KEY_PASSWORD**

Configuring IBM Integration Bus monitoring

The IBM Integration Bus agent is a multiple instance agent. You must create a first agent instance and start it manually.

Before you begin

- The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see "Change history" on page 54.
- Make sure that the system requirements for the IBM Integration Bus agent are met in your environment. For the up-to-date system requirement information, see the Detailed System Requirements Report for the IBM Integration Bus agent.

About this task

The following procedure is a roadmap for configuring IBM Integration Bus agent, which includes both required and optional steps. Complete the necessary steps according to your needs.

Procedure

1. Make sure the user ID that will be used to start and stop the IBM Integration Bus agent belongs to the **mqm** and **mqbrkrs** user groups.
2. Windows

   If IBM MQ (WebSphere MQ) is installed on the Windows system, add the IBM MQ (WebSphere MQ) library path to the **PATH** environment variable. So that the IBM Integration Bus agent can load the required IBM MQ (WebSphere MQ) libraries to start.

   a) Add the IBM MQ (WebSphere MQ) library path to the beginning of the **PATH** environment variable.

      For example, if the installation path of IBM MQ (WebSphere MQ) is `C:\IBM\WMQ75`, add `C:\IBM\WMQ75\bin` to the beginning of the **PATH** environment variable of your Windows system.

   b) Restart the Windows system for the changes to take effect.
3. Configure the IBM Integration Bus agent by specifying the following configuration parameters. There are also some optional configuration parameters that you can specify for the agent. For detailed instructions, see "Configuring the IBM Integration Bus agent" on page 274.
   - Agent ID
   - The installation directory of integration nodes (brokers) that are to be monitored
   - The 64-bit library path of IBM MQ (WebSphere MQ)
4. Configure IBM Integration Bus to enable the data that you want to monitor. See "Configuring IBM Integration Bus for data enablement" on page 278.
5. If you have enabled snapshot data collection for your integration node (broker), configure the IBM Integration Bus agent not to store any snapshot data. For instructions, see "Disabling snapshot data collection for the agent" on page 284.
6. Optional: To configure the IBM Integration Bus agent to enable transaction tracking, use the **Agent Configuration** page. For instructions, see "Configuring transaction tracking for the IBM Integration Bus agent" on page 285.

7. Optional: If you no longer need the transaction tracking function or you want to uninstall the IBM Integration Bus agent, disable transaction tracking for IBM Integration Bus and remove the agent provided user exit. For instructions, see “Disabling transaction tracking” on page 284 and “Removing the KQIUserExit user exit” on page 287.

Configuring the IBM Integration Bus agent

You must assign an instance name to the IBM Integration Bus agent and configure the agent before it can start monitoring your IBM Integration Bus environment.

Before you begin

- Make sure that the user ID that is used to start and stop the agent belongs to the **mqm** and **mqbrkrs** user groups.
- Windows If IBM MQ (WebSphere MQ) is installed on the Windows system, add the IBM MQ (WebSphere MQ) library path to the **PATH** environment variable. So that the IBM Integration Bus agent can load the required IBM MQ (WebSphere MQ) libraries to start.
  1. Add the IBM MQ (WebSphere MQ) library path to the beginning of the **PATH** environment variable.

     For example, if the installation path of IBM MQ (WebSphere MQ) is `C:\IBM\WMQ75`, add `C:\IBM\WMQ75\bin` to the beginning of the **PATH** environment variable of your Windows system.
  2. Restart the Windows system for the changes to take effect.
- You might need to provide the following information according to your environment during the agent configuration. If you do not know the appropriate configuration value to specify, gather the information from the administrator of IBM MQ (WebSphere MQ) and IBM Integration Bus.
  - If IBM MQ (WebSphere MQ) is installed on the same system with the IBM Integration Bus agent, you must provide the 64-bit library path of IBM MQ (WebSphere MQ).
  - If the IBM Integration Bus agent is configured to monitor the integration nodes of IBM Integration Bus V10 or IBM App Connect Enterprise V11, you must provide the installation directory for IBM Integration Bus V10 or IBM App Connect Enterprise V11.
  - If you want the IBM Integration Bus agent to monitor some specific integration nodes (brokers) instead of all on the same system, you must provide the name and installation path of each integration node (broker).

About this task

The IBM Integration Bus agent is a multiple instance agent; you must create the first instance and start the agent manually.

You can choose to configure the agent with or without interactions on UNIX or Linux systems. On Windows systems, you can configure the agent without interactions only.

- To configure the agent with interaction, run the configuration script and respond to prompts. See “Interactive configuration” on page 274.
- To configure the agent without interaction, edit the silent response file and then run the configuration script. See “Silent configuration” on page 276.

Important: If you also installed ITCAM Agent for WebSphere Message Broker, which is delivered as one of the ITCAM for Applications product, on the same system as the IBM Integration Bus agent, which is delivered in Cloud APM, do not use them to monitor the same integration node (broker) on the system.

Interactive configuration

Procedure

To configure the agent by running the script and responding to prompts, complete the following steps:

1. Enter the following command:

```
install_dir/bin/iib-agent.sh config instance_name
```

where *instance_name* is the name that you want to give to the agent instance.

Important: Interactive configuration is not supported on Windows systems.

2. After you confirm that you want to configure IBM Integration Bus agent, specify the configuration values for general agent settings.

   a) When prompted for the **`Agent Id`** parameter, specify a unique alphanumeric string with a maximum length of 8 characters.

      Remember: The maximum agent ID length is changed to 8 characters from IBM Integration Bus agent version 7.3.0.1. For previous versions, the maximum agent ID length is 4 characters.

      The managed system name includes the agent ID that you specify, for example, `monitoredbrokername:agentID:KQIB`, where *monitoredbrokername* is the name of the monitored integration node (broker).

   b) When prompted for the **`IIB version 10 or ACE version 11`** Install Directory parameter, if you want to monitor integration nodes of IBM Integration Bus V10, or IBM App Connect Enterprise V11, specify the installation directory of IBM Integration Bus V10 or IBM App Connect Enterprise V11. For example, `/opt/ibm/mqsi/ace-11.0.0.3`. If you do not want to monitor IBM Integration Bus V10 and IBM App Connect Enterprise V11, press Enter to accept the default.

      Remember: You can specify only one installation directory for the **`IIB version 10 or ACE version 11`** Install Directory parameter. If you installed IBM Integration Bus V10 or IBM App Connect Enterprise V11 in different directories and you want to monitor them all, create multiple agent instances and specify one installation directory of IBM Integration Bus V10 or IBM App Connect Enterprise V11 for each agent instance.

3. Optional: Use the **Monitored Broker Settings** section to specify whether you want to use this agent to monitor only some specific integration nodes (brokers).

   By default, all integration nodes (brokers) that are running on the same host system as the IBM Integration Bus agent are monitored, as determined by self-discovery. If you want the agent to monitor some specific integration nodes (brokers), specify the name of the integration node (broker) that you want to monitor and set the **Collect Node Data** setting to No, which is the default value, in the **Monitored Broker Settings** section. There can be multiple **Monitored Broker Settings** sections. Each section controls the monitoring settings for one integration node (broker).

   Tip: You can specify more than one **Monitored Broker Settings** section. When you edit the **Monitored Broker Settings** section, the following options are available:

   - Add: Create a **Monitored Broker Settings** section to configure for another integration node (broker).
   - Edit: Modify the settings of current **Monitored Broker Settings** section.
   - Del: Delete the current **Monitored Broker Settings** section.
   - Next: Move to the next **Monitored Broker Settings** section.
   - Exit: Exit the **Monitored Broker Settings** configuration.

4. If you confirm that IBM MQ (WebSphere MQ) is installed on the same system, you are prompted for the **`WebSphere MQ 64-bit library path`** parameter. Press Enter to accept the default value, which is the 64-bit library path of IBM MQ (WebSphere MQ) automatically discovered by the agent. If no default value is displayed, you must provide the 64-bit library path of IBM MQ (WebSphere MQ) before you proceed to the next step. For example, `/opt/mqm8/lib64`.

   Remember: If your integration nodes (brokers) use different versions of queue managers, specify the latest version of the IBM MQ (WebSphere MQ) 64-bit library path for this parameter.

5. After the configuration completes, enter the following command to start the agent:

```
install_dir/bin/iib-agent.sh start instance_name
```

Silent configuration

Procedure

To configure the agent by editing the silent response file and running the script with no interaction, complete the following steps:

1. Open the following agent silent response file in a text editor.
   - Linux AIX `install_dir/samples/iib_silent_config.txt`
   - Windows `install_dir\tmaitm6_x64\samples\qi_silent_config.txt`

   where *install_dir* is the installation directory of the IBM Integration Bus agent. The default installation directory is as follows:

   - Linux AIX `/opt/ibm/apm/agent`
   - Windows `C:\IBM\APM`
2. For the **agentId** parameter, specify a unique alphanumeric string with a maximum length of 8 characters as a short identifier for the agent.

   Remember: The maximum agent ID length is changed to 8 characters from IBM Integration Bus agent version 7.3.0.1. For previous versions, the maximum agent ID length is 4 characters.

   The managed system name includes the agent ID that you specify, for example, *monitoredbrokername*:*agentID*:KQIB, where *monitoredbrokername* is the name of the monitored integration node (broker).
3. If you want to monitor the integration nodes of IBM Integration Bus V10 or IBM App Connect Enterprise V11, specify the installation directory for IBM Integration Bus V10 or IBM App Connect Enterprise V11 for the **defaultWMBInstallDirectory** parameter. For example, `C:\Program Files\IBM\ACE\11.0.0.3\` for a Windows system, or `/opt/ibm/mqsi/ace-11.0.0.3` for a Linux system. If you do not want to monitor IBM Integration Bus V10 and IBM App Connect Enterprise V11, this parameter is not required because the IBM Integration Bus agent can discover the integration nodes (brokers) from earlier versions automatically.

   Remember: You can specify only one installation directory for the **defaultWMBInstallDirectory** parameter. If you installed IBM Integration Bus V10 or IBM App Connect Enterprise V11 in different directories and you want to monitor them all, create multiple agent instances and specify one installation directory of IBM Integration Bus V10 or IBM App Connect Enterprise V11 for each agent instance.
4. Optional: Specify whether you want to use this agent to monitor only some specific integration nodes (brokers).

   By default, all integration nodes (brokers) that are running on the same host system as the IBM Integration Bus agent are monitored, as determined by self-discovery. To monitor specific integration nodes (brokers), set the **collectNodeData** and **WMBInstallDirectory** parameters for each integration node (broker) that you want to monitor.

   collectNodeData

   Specifies whether node definition data is collected for the monitored integration node (broker). The syntax is `collectNodeData.`*brkr_name*=NO|YES, where *brkr_name* is the name of the integration node (broker).

   The default value is NO. It is recommended to use the default value because node definition data is not supported in the Cloud APM console.

   WMBInstallDirectory

   The installation directory of the integration node (broker) to be monitored. The syntax is `WMBInstallDirectory.`*brkr_name*=*broker_install_dir*, where *broker_install_dir* is the installation directory of the integration node (broker) to be monitored.

   Remember: For a version 10 integration node, the **WMBInstallDirectory** parameter can override the **defaultWMBInstallDirectory** parameter that you set in the previous step.

   For example, to monitor only two integration nodes (brokers) that are named BK1 and BK2, set the parameters as follows:

   ```
   collectNodeData.BK1=NO
   collectNodeData.BK2=NO
   WMBInstallDirectory.BK1=BK1_install_dir
   WMBInstallDirectory.BK2=BK2_install_dir
   ```

5. To monitor brokers that are earlier than IBM Integration Bus V10, specify the 64-bit library path of IBM MQ (WebSphere MQ) for the **WMQLIBPATH** parameter. For example, `C:\Program Files\IBM\WebSphere MQ\bin64` for a Windows system, or `/opt/mqm8/lib64` for a Linux system.

   Remember: If your integration nodes (brokers) use different versions of queue managers, specify the latest version of the IBM MQ (WebSphere MQ) 64-bit library path for this parameter.

6. Save and close the agent silent response file, and then enter the following command:

   - Linux AIX `install_dir/bin/iib-agent.sh config instance_name path_to_responsefile`
   - Windows `install_dir\BIN\iib-agent.bat config "instance_name path_to_responsefile"`

   where *instance_name* is the name of the instance that you configure, and *path_to_responsefile* is the full path of the silent response file.

   Warning: On Windows systems, do not include double quotation marks ("") that enclose the full path to the silent response file, as this will cause a configuration error.

7. After the configuration completes, enter the following command to start the agent:

   - Linux AIX

     ```
     install_dir/bin/iib-agent.sh start instance_name
     ```

   - Windows

     ```
     install_dir\bin\iib-agent.bat start instance_name
     ```

Results

Now, you can log in to the Cloud APM console and use the Applications editor to add the IBM Integration Bus agent instance to the Application Performance Dashboard. For instructions on how to start the Cloud APM console, see "Starting the Cloud APM console" on page 957. For information about using the Applications editor, see "Managing applications" on page 1073.

Remember: Whenever you update or migrate a monitored integration node (broker), you must restart the IBM Integration Bus agent after the integration node (broker) upgrade or migration.

What to do next

The next step is to configure IBM Integration Bus for data enablement. The following data is available on the Application Performance Dashboard only after you enable them in IBM Integration Bus:

- Archive accounting and statistics
- JVM resource statistics
- Transaction tracking

For instructions, see "Configuring IBM Integration Bus for data enablement" on page 278.

Configuring IBM Integration Bus for data enablement

For some data to be available in the Cloud APM console, you must configure IBM Integration Bus to enable the required data collection.

Before you begin

Make sure that the IBM Integration Bus agent is configured.

Remember: Transaction tracking enablement requires you to restart the integration node (broker).

About this task

Archive statistics and resource statistics can be monitored by the IBM Integration Bus agent only after the data collection is enabled for the integration node (broker). Similarly, if you want to see the transaction tracking in the middleware and topology dashboards, you must enable transaction tracking within the integration node (broker) before you enable transaction tracking for the IBM Integration Bus agent.

Decide what type of data that you want to monitor with the IBM Integration Bus agent and complete the following steps according to your needs.

Integration servers that are owned by the integration node have a default `server.conf.yaml` configuration file for each integration server that is stored in a subdirectory of the integration node directory. Any properties that you set for the integration node in the `node.conf.yaml` file are inherited by the integration servers that it owns. Nonetheless, you can change an integration server's properties by modifying them in its `server.conf.yaml` file. (For more information, see Configuring an integration node by modifying the node.conf.yaml file in the IBM App Connect Enterprise documentation.)

Procedure

- To enable archive statistics data collection for the integration node (broker), see "Enabling archive accounting and statistics data collection" on page 278.
- To enable resource statistics data for an integration node (broker), see "Enabling JVM resource statistics" on page 281.
- To enable transaction tracking for message flows within an integration node (broker), see "Enabling transaction tracking" on page 282.
- If you no longer want transaction tracking data, remember to disable transaction tracking for the integration node (broker) on which it was enabled. See "Disabling transaction tracking" on page 284.

Enabling archive accounting and statistics data collection

About this task

To enable archive accounting and statistics collection for message flows that belong to the integration node (broker), issue the **`mqsichangeflowstats`** command from the `bin` directory of the integration node (broker) installation directory.

Remember: Issue the **`mqsichangeflowstats`** command to the integration node (broker) according to your requirements for monitoring data. It is recommended that you enable only the statistics that you require, because there can be a lot of data and processing when you have many message flows. For more detailed information about the **`mqsichangeflowstats`** command, refer to IBM Integration Bus documentation.

Important: IBM Cloud Application Performance Management does not support snapshot accounting and statistics data due to the amount of data and processing required for the set 20 second snapshot interval. Archive data provides the same exact attributes as snapshot data, and is more suitable for the regular production monitoring provided by IBM Cloud Application Performance Management. If you have enabled snapshot data collection for the integration node (broker), remember to configure the IBM Integration Bus agent not to store the snapshot data. For instructions, see "Disabling snapshot data collection for the agent" on page 284.

Procedure

- To get most data for message flows, issue the following command. This command is recommended because it does not enable the most detailed terminal statistics that provide invocation counts per terminal per node. The terminal level consumes a lot of storage.

  ```
  mqsichangeflowstats BrokerName -a -g -j -c active -t none -n basic -o xml
  ```

- In ACE version 11, to get most data for message flows, modify the `node.conf.yaml` or `server.conf.yaml` file as follows. These properties are recommended because they do not enable the most detailed terminal statistics that provide invocation counts per terminal per node. The terminal level consumes a lot of storage.

  ```
  Statistics:
    # Application message flows will by default inherit Snapshot and Archive values
    # set here
    Snapshot:
      #publicationOn: 'inactive' # choose 1 of : active|inactive, default inactive
                                 # Ensure Events.OperationalEvents.MQ|MQTT
                                 # is set for outputFormat json,xml
      #accountingOrigin: 'none'  # choose 1 of : none|basic
      #nodeDataLevel: 'none'     # choose 1 of : none|basic|advanced
      #outputFormat: 'usertrace' # comma separated list of :
                                 #csv,bluemix,json,xml,usertrace
      #threadDataLevel: 'none'   # choose 1 of : none|basic
    Archive:
      archivalOn: 'active'     # choose 1 of : active|inactive,
                               # default inactive
                                 # Ensure Events.OperationalEvents.MQ|MQTT
                                 # is set for outputFormat xml
      #accountingOrigin: 'none'  # choose 1 of : none|basic
      #majorInterval: 60         # Sets the interval in minutes at which
                                #archive statistics are published
      nodeDataLevel: 'basic'         # choose 1 of : none|basic|advanced
      outputFormat: 'xml'  # comma separated list of : csv,xml,usertrace
      #threadDataLevel: 'none'   # choose 1 of : none|basic
  ```

  Note: If you want to disable this setting, comment out the lines of **archivalOn: 'active'**, **nodeDataLevel: 'basic'**, and **outputFormat: 'xml'**.

- To get all the data supported by the IBM Integration Bus agent, issue the following command:

  ```
  mqsichangeflowstats BrokerName -a -g -j -c active -t none -n advanced -o xml
  ```

- In ACE version 11, to get all the data supported by the IBM Integration Bus agent, modify the `node.conf.yaml` or `server.conf.yaml` file as follows:

  ```
  Statistics:
    # Application message flows will by default inherit Snapshot and Archive values
    # set here
    Snapshot:
      #publicationOn: 'inactive' # choose 1 of : active|inactive, default inactive
                                 # Ensure Events.OperationalEvents.MQ|MQTT
                                 # is set for outputFormat json,xml
      #accountingOrigin: 'none'  # choose 1 of : none|basic
      #nodeDataLevel: 'none'     # choose 1 of : none|basic|advanced
      #outputFormat: 'usertrace' # comma separated list of :
                                 # csv,bluemix,json,xml,usertrace
      #threadDataLevel: 'none'   # choose 1 of : none|basic
    Archive:
      archivalOn: 'active'    # choose 1 of : active|inactive, default inactive
                                 # Ensure Events.OperationalEvents.MQ|MQTT
                                 # is set for outputFormat xml
      #accountingOrigin: 'none'  # choose 1 of : none|basic
      #majorInterval: 60       # Sets the interval in minutes at which
                               # archive statistics are published
      nodeDataLevel: 'advanced'     # choose 1 of : none|basic|advanced
      outputFormat: 'xml' # comma separated list of : csv,xml,usertrace
      #threadDataLevel: 'none' # choose 1 of : none|basic
  ```

  Note: If you want to disable this setting, comment out the lines of **archivalOn: 'active'**, **nodeDataLevel: 'advanced'**, and **outputFormat: 'xml'**.

- To reduce the amount of data but still reasonably monitor all message flows without further details, issue the following command:

```
mqsichangeflowstats BrokerName -a -g -j -c active -t none -n none -o xml
```

- In ACE version 11, to reduce the amount of data but still reasonably monitor all message flows without further details, modify the `node.conf.yaml` or `server.conf.yaml` file as follows:

```
Statistics:
  # Application message flows will by default inherit Snapshot and Archive values
  #set here
  Snapshot:
    #publicationOn: 'inactive' # choose 1 of : active|inactive, default inactive
                               # Ensure Events.OperationalEvents.MQ|MQTT
                               # is set for outputFormat json,xml
    #accountingOrigin: 'none'  # choose 1 of : none|basic
    #nodeDataLevel: 'none'     # choose 1 of : none|basic|advanced
    #outputFormat: 'usertrace' # comma separated list of :
                               # csv,bluemix,json,xml,usertrace
    #threadDataLevel: 'none'   # choose 1 of : none|basic
  Archive:
    archivalOn: 'active'    # choose 1 of : active|inactive, default inactive
                               # Ensure Events.OperationalEvents.MQ|MQTT
                               # is set for outputFormat xml
    #accountingOrigin: 'none'  # choose 1 of : none|basic
    #majorInterval: 60       # Sets the interval in minutes at which
                             # archive statistics are published
    nodeDataLevel: 'none'         # choose 1 of : none|basic|advanced
    outputFormat: 'xml' # comma separated list of : csv,xml,usertrace
    #threadDataLevel: 'none' # choose 1 of : none|basic
```

Note: If you want to disable this setting, comment out the lines of **archivalOn: 'active'**, **nodeDataLevel: 'none'**, and **outputFormat: 'xml'**.

- If you have a large number of message flows and want to reduce the amount of data, you can specify which message flows to monitor by replacing the `-g` or `-j` option in the previously mentioned commands.
  - To specify a particular integration server (execution group) for enablement, replace `-g` with `-e` *`IntegrationServerName`* .
  - To identify a particular message flow for enablement, replace `-j` with `-f` *`MessageFlowName`*.
  - If you have grouped your message flows into applications, to specify a particular application for enablement, add `-k` *`ApplicationName`* to the `-j` option.
- The IBM Integration Bus agent collects archive accounting and statistics data at the interval of 5 minutes. To set the interval at which the integration node (broker) produces the archive accounting and statistics data to the same interval, issue the following command with the integration node (broker) stopped, and then restart the integration node (broker):

```
mqsichangebroker BrokerName -v 5
```

- In ACE version 11, the IBM Integration Bus agent collects archive accounting and statistics data at the interval of 5 minutes. To set the interval at which the integration node (broker) produces the archive accounting and statistics data to the same interval, modify the `node.conf.yaml` or `server.conf.yaml` file as follows:

```
Statistics:
  # Application message flows will by default inherit Snapshot and Archive values
  # set here
  Snapshot:
    #publicationOn: 'inactive' # choose 1 of : active|inactive, default inactive
                               # Ensure Events.OperationalEvents.MQ|MQTT
                               # is set for outputFormat json,xml
    #accountingOrigin: 'none'  # choose 1 of : none|basic
    #nodeDataLevel: 'none'     # choose 1 of : none|basic|advanced
    #outputFormat: 'usertrace' # comma separated list of :
                               # csv,bluemix,json,xml,usertrace
    #threadDataLevel: 'none'   # choose 1 of : none|basic
  Archive:
    archivalOn: 'active'     # choose 1 of : active|inactive, default inactive
                               # Ensure Events.OperationalEvents.MQ|MQTT
                               # is set for outputFormat xml
    #accountingOrigin: 'none'  # choose 1 of : none|basic
    majorInterval: 5         # Sets the interval in minutes at which
                             # archive statistics are published
    nodeDataLevel: 'none'         # choose 1 of : none|basic|advanced
    outputFormat: 'xml' # comma separated list of : csv,xml,usertrace
    #threadDataLevel: 'none'       # choose 1 of : none|basic
```

Results

After the IBM Integration Bus agent is configured and started, the message flow accounting and statistics data is displayed in the following group widgets:

- Message Flow Dashboard
  - Commits & Backouts
  - CPU Microseconds
  - Elapsed Microseconds
  - Input Byte Rate
  - Input Message Rate
  - Input Message Size
  - Input Message Wait CPU Microseconds
  - Input Message Wait Elapsed Microseconds
  - Message Flow Errors
  - Message Processing Node Statistics
- Processing Node Dashboard
  - CPU Microseconds
  - Elapsed Microseconds
  - Invocations
  - Processing Node Status
  - Terminal Statistics

Enabling JVM resource statistics

About this task

To enable JVM resource statistics for integration servers that belong to the integration node (broker), issue the **mqsichangeresourcestats** command from the bin directory of the integration node (broker) installation directory.

Remember: The JVM resource statistics are considered optional because only a few attributes of data are displayed for the high cost of the agent processing this data every 20 seconds. Be sure to consider whether you really need the JVM resource statistics data.

Procedure

- To enable the statistics across all integration servers in the integration node (broker), issue the following command:

  ```
  mqsichangeresourcestats BrokerName -c active
  ```

- In ACE version 11, to enable the statistics across all integration servers in the integration node (broker), modify the node.conf.yaml file as follows:

  ```
  Statistics:
    # Application message flows will by default inherit Snapshot and Archive values
    # set here
    Snapshot:
      #publicationOn: 'inactive' # choose 1 of : active|inactive, default inactive
                                 # Ensure Events.OperationalEvents.MQ|MQTT
  ```

```
                               # is set for outputFormat json,xml
    #accountingOrigin: 'none'  # choose 1 of : none|basic
    #nodeDataLevel: 'none'     # choose 1 of : none|basic|advanced
    #outputFormat: 'usertrace' # comma separated list of :
                               # csv,bluemix,json,xml,usertrace
    #threadDataLevel: 'none'   # choose 1 of : none|basic
  Archive:
    archivalOn: 'active'    # choose 1 of : active|inactive, default inactive
                               # Ensure Events.OperationalEvents.MQ|MQTT
                               # is set for outputFormat xml
    #accountingOrigin: 'none'  # choose 1 of : none|basic
    majorInterval: 5         # Sets the interval in minutes at which
                             # archive statistics are published
    nodeDataLevel: 'advanced'     # choose 1 of : none|basic|advanced
    outputFormat: 'xml' # comma separated list of : csv,xml,usertrace
    threadDataLevel: 'basic'      # choose 1 of : none|basic
   Resource:
     reportingOn: true       # choose 1 of : true|false, default false
......
```

Note: If you want to disable this setting, comment out **reportingOn: true**.

- To enable the statistics for a given integration server in the integration node (broker), issue the following command:

```
mqsichangeresourcestats BrokerName -e IntegrationServerName -c active
```

- In ACE version 11, to enable the statistics for a given integration server in the integration node (broker), modify the `server.conf.yaml` file as follows:

```
Statistics:
  # Application message flows will by default inherit Snapshot and Archive values
  # set here
  Snapshot:
    #publicationOn: 'inactive' # choose 1 of : active|inactive, default inactive
                               # Ensure Events.OperationalEvents.MQ|MQTT
                               # is set for outputFormat json,xml
    #accountingOrigin: 'none'  # choose 1 of : none|basic
    #nodeDataLevel: 'none'     # choose 1 of : none|basic|advanced
    #outputFormat: 'usertrace' # comma separated list of :
                               # csv,bluemix,json,xml,usertrace
    #threadDataLevel: 'none'   # choose 1 of : none|basic
  Archive:
    archivalOn: 'active'    # choose 1 of : active|inactive, default inactive
                               # Ensure Events.OperationalEvents.MQ|MQTT
                               # is set for outputFormat xml
    #accountingOrigin: 'none'  # choose 1 of : none|basic
    majorInterval: 5         # Sets the interval in minutes at which
                             # archive statistics are published
    nodeDataLevel: 'advanced'     # choose 1 of : none|basic|advanced
    outputFormat: 'xml' # comma separated list of : csv,xml,usertrace
    threadDataLevel: 'basic'   # choose 1 of : none|basic
  Resource:
    reportingOn: true        # choose 1 of : true|false, default false
```

Note: If you want to disable this setting, comment out **reportingOn: true**.

Results

The JVM resource statistics data is displayed in the following group widgets:

- Garbage Collection Count
- Garbage Collection Duration
- JVM Non-Heap Memory
- JVM Heap Memory

Enabling transaction tracking

Before you begin

1. Make sure that the IBM Integration Bus agent is installed. A user exit named `KQIUserExit` is provided to enable IBM Integration Bus for transaction tracking.

2. Make sure that the user who will start the integration node (broker) has access to the KQI User Exit module directory. That is, ensure that you add the user ID that is used to start the integration node (broker) to the group under which you installed the IBM Integration Bus agent.

About this task

You must deploy the `KQIUserExit` user exit to the integration node (broker). Otherwise, no data is available in the middleware and topology dashboards even after you have enabled the IBM Integration Bus agent for transaction tracking.

Tip: The following IBM Integration Bus nodes are included in the middleware and topology dashboards by the `KQIUserExit` user exit as uninstrumented services:

- Database and compute nodes where an ODBC data source is specified
- TCP/IP nodes
- File nodes for remote FTP or FTPS servers
- MQ nodes, unless already instrumented

Procedure

To enable transaction tracking for IBM Integration Bus, complete the following steps:

1. Linux AIX
   Close any broker shells that have loaded the MQSI environment.
2. Open an IBM Integration Bus command console with one of the following methods. If you have multiple versions of integration nodes (brokers) installed, ensure that you start the command console for the correct version.
   - Windows Click **Start** > **IBM Integration Bus** > **IBM Integration Console**
   - Linux AIX From the `bin` directory of integration node (broker) installation directory, issue the **`mqsiprofile`** command.
3. Stop the integration node (broker) that you want to configure with the **`mqsistop`** command.
4. Enable transaction tracking for the message flow within the integration node (broker) by adding the KQIUserExit user exit with the **`mqsichangebroker`** command.
   - To enable transaction tracking for all message flows within the integration node (broker), run the following command:

     ```
     mqsichangebroker broker_name -e "KQIUserExit"
     ```

   - To enable transaction tracking for a specific message flow within the integration node (broker), run the following command:

     ```
     mqsichangeflowuserexits broker_name -e execution_group_name -k application_name -f
     message_flow_name -a "KQIUserExit"
     ```

5. Alternatively, in Ace version 11, enable transaction tracking for the message flow within the integration node (broker) by adding the `KQIUserExit` to the `node.conf.yaml` or the `server.conf.yaml` file.

   ```
   UserExits:
   activeUserExitList: 'KQIUserExit' # Specify the name
   #of an installed user exit to activate.
   ```

   Note: If you want to disable the transaction tracking, comment out **activeUserExitList: 'KQIUserExit'**.
6. Restart the integration node (broker) with the **`mqsistart`** command.

Disabling transaction tracking

Procedure

To disable transaction tracking for IBM Integration Bus, complete the following steps:

1. Open an IBM Integration Bus command console with one of the following methods. If you have multiple versions of integration nodes (brokers) installed, ensure that you start the command console for the correct version.
   - Windows Click **Start** > **IBM Integration Bus** > **IBM Integration Console**
   - Linux AIX From the `bin` directory of integration node (broker) installation directory, issue the **`mqsiprofile`** command.
2. Disable transaction tracking for the message flow within an integration node (broker) with one of the following methods:
   - To disable transaction tracking for a specific message flow, use the **`mqsichangeflowuserexits`** command:

     ```
     mqsichangeflowuserexits broker_name -e execution_group_name
     -f message_flow_name -a ""
     ```
   - To disable transaction tracking for all message flows within the integration node (broker), first stop the integration node (broker) with the **`mqsistop`** command and then issue the **`mqsichangebroker`** command:

     ```
     mqsichangebroker broker_name -e ""
     ```

What to do next

- For transaction tracking, after you enable transaction tracking for IBM Integration Bus, you must also enable transaction tracking for the agent. For instructions, see "Configuring transaction tracking for the IBM Integration Bus agent" on page 285.
- If you have enabled snapshot data collection for your integration node (broker), configure the IBM Integration Bus agent not to store any snapshot data. Cloud APM does not support snapshot accounting and statistics data due to the amount of data and processing required for the set 20 second snapshot interval. For instructions, see "Disabling snapshot data collection for the agent" on page 284.

Disabling snapshot data collection for the agent

Cloud APM does not support snapshot accounting and statistics data due to the amount of data and processing required for the set 20 second snapshot interval. If you have enabled snapshot data collection for the broker, remember to configure the IBM Integration Bus agent not to store the snapshot data.

Procedure

1. Open the agent configuration file in a text editor. The agent configuration file is in one of the following directories depending on the operating system:
   - Linux AIX *`install_dir`*`/config/<`*`hostname`*`>_qi_<`*`instance_name`*`>.cfg`
   - Windows *`install_dir`*`\TMAITM6_x64\<`*`hostname`*`>_qi_<`*`instance_name`*`>.cfg`

   where *install_dir* is the agent installation directory; *hostname* is the host name of the operating system; *instance_name* is the agent instance name.
2. Edit the file by adding the following parameter to the KqiAgent section:

   ```
   defaultRetainRecentSnapshotSamples=0
   ```

   Example:

```
INSTANCE=inst1 [
SECTION=KqiAgent [ { agentId=inst1 } { instName=inst1 }
{defaultRetainRecentSnapshotSamples=0}]
SECTION=MonitorBroker:BRK1 [ { collectNodeData=NO } ]
SECTION=MonitorBroker:BRK2 [ { collectNodeData=NO } ]
]
```

3. Save and close the file.
4. Restart the IBM Integration Bus agent for the changes to take effect.

Configuring transaction tracking for the IBM Integration Bus agent

Transaction tracking data for IBM Integration Bus can be displayed in the middleware and topology dashboards after you enable the data collection on the **Agent Configuration** page for the IBM Integration Bus agent.

Before you begin

- Make sure that transaction tracking is enabled for IBM Integration Bus with the agent provided user exit named `KQIUserExit`. If you have not done it, follow the instructions in "Enabling transaction tracking" on page 282.
- Make sure that the IBM Integration Bus agent is configured appropriately. If you have not done it, follow the instructions in "Configuring the IBM Integration Bus agent" on page 274.

Procedure

To configure transaction tracking for the IBM Integration Bus agent, complete the following steps:

1. From the navigation bar, click **System Configuration** > **Agent Configuration**.

   The **Agent Configuration** page is displayed.
2. Click the **IBM Integration Bus** tab.
3. Select the check boxes for the agent instances and take one of the following actions from the **Actions** list:

   - To enable transaction tracking, click **Set Transaction Tracking** > **Enabled**. The status in the **Transaction Tracking** column is updated to Enabled.
   - To disable the transaction tracking data, click **Set Transaction Tracking** > **Disabled**. The status in the **Transaction Tracking** column is updated to Disabled.

Results

You have configured transaction tracking for the selected agent instances. Transaction tracking data can be displayed in the middleware and topology dashboards after you enable the data collection. For further information, see "Adding middleware applications to the Application Performance Dashboard" on page 100.

Specifying unique managed system name for IBM Integration Bus agent

The IBM Integration Bus agent instance name displayed on the Cloud APM console is also known as the managed system name(MSN). You can use the agent configuration parameter to specify unique MSN for each agent instance.

About this task

When the IBM Integration Bus agent is started, it registers the MSN in the format of *monitoredbrokername*:*agentID*:KQIB for each agent instance, where *monitoredbrokername* is the name of monitored broker and *agentID* is the agent ID that is set by agent configuration parameter. The maximum length of the MSN is 32 characters. If the MSN length exceeds 32 characters, it will truncated.

Unique MSN might be required in the following circumstances:

- You are running more than one IBM Integration Bus agent on the same system.

- You are running more than one monitored broker with the same name on different systems.

To specify an agent ID to get a unique MSN, use the **Agent Id** option during interactive configuration or use the **agentId** parameter in the silent response file.

Remember: If you have not configured the IBM Integration Bus agent for the first time after installation, follow the steps as documented in "Configuring the IBM Integration Bus agent" on page 274.

Procedure

- To use the **Agent Id** option during interactive configuration, complete the following steps:
  a) Enter the following command:

  ```
  install_dir/bin/iib-agent.sh config instance_name
  ```

  where *instance_name* is the agent instance name that you want to specify an agent ID for.

  b) Follow the options to configure the agent instance.

  If no change is needed for an option, use the default value.

  c) When the Agent Id option appears, specify the middle qualifier for the managed system name.

  The valid format is an alphanumeric string with a maximum length of 8 characters.

- To use the **agentId** parameter in the silent response file, complete the following steps:
  a) Open the following agent silent response file in a text editor.
  - Linux AIX *install_dir*/samples/iib_silent_config.txt
  - Windows *install_dir*\tmaitm6_x64\samples\qi_silent_config.txt

  b) Specify an agent ID for the **agentId** parameter.

  The valid format is an alphanumeric string with a maximum length of 8 characters.

  c) Save and close the silent response file, and then run the following command from the command line:
  - Linux AIX *install_dir*/bin/iib-agent.sh config *instance_name path_to_responsefile*
  - Windows *install_dir*\BIN\iib-agent.bat config "*instance_name path_to_responsefile*"

  where *instance_name* is the name of the instance that you configure, and *path_to_responsefile* is the full path of the silent response file.

  ⚠ **Warning:** On Windows systems, do not include double quotation marks ("") that enclose the full path to the silent response file, as this will cause a configuration error.

  d) After the configuration completes, enter the following command to start the agent:
  - Linux AIX

    ```
    install_dir/bin/iib-agent.sh start instance_name
    ```

  - Windows

    ```
    install_dir\bin\iib-agent.bat start instance_name
    ```

What to do next

Log in to the Cloud APM console. If the agent instance with previous MSN is still displayed as offline, edit your application to remove it and then add the new agent instance with the assigned agent ID.

Removing the KQIUserExit user exit

Before you uninstall the IBM Integration Bus agent, you must first remove the `KQIUserExit` user exit.

Procedure

Complete the following steps to remove the `KQIUserExit` user exit that you deployed to IBM Integration Bus for transaction tracking:

1. Navigate to the `bin` directory of the IBM Integration Bus agent.
   - Windows *agent_install_dir*`\`*arch*`\qi\bin`
   - Linux AIX *agent_install_dir*`/`*arch*`/qi/bin`

   where:

   - *agent_install_dir* is the agent installation directory. The default is `C:\IBM\APM` on Windows systems and `/opt/ibm/apm/agent` on Linux and AIX systems.
   - *arch* is the architecture code of the platform. For example, lx8266 represents Linux Intel v2.6 (64-bit). For a complete list of the architecture codes, see the *agent_install_dir*`/archdsc.tbl` file.
2. Run the **`configDC`** script to remove the user exit library interactively:
   - Windows

     ```
     configDC.bat -disable iib_install_dir
     ```

   - Linux AIX

     ```
     ./configDC.sh -disable iib_install_dir
     ```

   where *iib_install_dir* is the installation directory of the IBM Integration Bus.

Example

The following example removes the agent provided user exit for version 9.0 brokers that are installed on an AIX system:

```
cd /opt/IBM/ITM/aix513/qi/bin
./configDC.sh -disable /opt/IBM/mqsi/9.0
```

Configuring IBM MQ Appliances monitoring

The MQ Appliance agent is a multi-instance agent. After installation, you must configure the agent by creating an agent instance before you can start monitoring with the agent.

Before you begin

- The directions here are for the most current release of the agent, except as indicated.

Procedure

- On Linux and UNIX systems, you can configure the agent with the configuration script that prompts for responses or the silent response file.
  - "Configuring the agent by responding to prompts" on page 288
  - "Configuring the agent by using the silent response file" on page 288
- On Windows systems, you can configure the agent only with the silent response file.
  - "Configuring the agent by using the silent response file" on page 288

What to do next

In the Cloud APM console, go to your Application Performance Dashboard to view the data that was collected. For more information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you are unable to view the data in the agent dashboards, first check the server connection logs and then the data provider logs. The default paths to these logs are listed here:

- Linux AIX `/opt/ibm/apm/agent/logs`
- Windows `C:\IBM\APM\TMAITM6_x64\logs`

Configuring the agent by responding to prompts

You must assign an instance name to the MQ Appliance agent and configure the agent before it can start monitoring your IBM® MQ Appliances.

Procedure

To configure the agent by running the script and responding to prompts, complete the following steps:

1. Run the following command:

   ```
   install_dir/bin/mq_appliance-agent.sh config instance_name
   ```

   where *install_dir* is the path where the agent is installed and *instance_name* is the name that you want to give to the agent instance.

   Example:

   ```
   /opt/ibm/apm/agent/bin/mq_appliance-agent.sh config AQM904
   ```

2. Respond to the prompts to set configuration values for the agent.

   See "Configuration parameters for the MQ Appliance agent" on page 289 for an explanation of each of the configuration parameters.

3. Run the following command to start the agent:

   ```
   install_dir/bin/mq_appliance-agent.sh start instance_name
   ```

   Example:

   ```
   /opt/ibm/apm/agent/bin/mq_appliance-agent.sh start AQM904
   ```

Results

Now, you can log in to the Cloud APM console and use the Applications editor to add the MQ Appliance agent instance to the Application Performance Dashboard. For instructions on how to start the Cloud APM console, see "Starting the Cloud APM console" on page 957. For information about using the Applications editor, see "Managing applications" on page 1073.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

The silent response file contains the agent configuration parameters with default values that are defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

To configure the agent by editing the silent response file and running the script without interaction, complete the following steps:

1. Open the `mq_appliance_silent_config.txt` file in one of the following directories in a text editor.

   - Linux AIX *install_dir*/samples/mq_appliance_silent_config.txt
   - Windows *install_dir*\samples\mq_appliance_silent_config.txt

   where, *install_dir* is the agent installation directory. For example, `/opt/ibm/apm/agent`.
2. In the `mq_appliance_silent_config.txt` file, specify values for all mandatory parameters and modify the default values of other parameters as needed.

   See "Configuration parameters for the MQ Appliance agent" on page 289 for an explanation of each of the configuration parameters.
3. Save and close the `mq_appliance_silent_config.txt` file, and run the following command:

   - Linux AIX

     ```
     install_dir/bin/mq_appliance-agent.sh config instance_name path_to_silent_file
     ```

   - Windows

     ```
     install_dir\bin\mq_appliance-agent.bat config instance_name path_to_silent_file
     ```

   where:

   - *instance_name* is the name that you want to give to the agent instance. For example, `AQM904`.
   - *path_to_silent_file* is the path to the `mq_appliance_silent_config.txt` file. For example, `/opt/ibm/apm/agent/samples/mq_appliance_silent_config.txt`.
4. After configuration completes, run the following command to start the agent:

   - Linux AIX

     ```
     install_dir/bin/mq_appliance-agent.sh start instance_name
     ```

   - Windows

     ```
     install_dir\bin\mq_appliance-agent.bat start instance_name
     ```

Results

Now, you can log in to the Cloud APM console and use the Applications editor to add the MQ Appliance agent instance to the Application Performance Dashboard. For instructions on how to start the Cloud APM console, see "Starting the Cloud APM console" on page 957. For information about using the Applications editor, see "Managing applications" on page 1073.

Configuration parameters for the MQ Appliance agent

The configuration parameters for the MQ Appliance agent are displayed in tables which group them according to sections.

- Table 24 on page 290: Properties for receiving SNMP events and decoding V3 events.
- Table 25 on page 291: Properties for Java settings.
- Table 26 on page 291: Properties for the proxy server used by HTTP providers.
- Table 27 on page 291: Properties for HTTP server.

- Table 28 on page 292: Properties for connecting to the MQ appliance.

Table 24. SNMP event configuration parameters

Parameter name	Description	Parameter name in silent configuration file
Port Number	The port number that is used to listen for SNMP events. The default is 162.	KQZ_SNMPEVENT_PORT
Security Level	The security level that is used to connect to the SNMP event. It can be one of the following values: • 1=noAuthNoPriv • 2=authNoPriv • 3=authPriv The default is 2.	KQZ_SNMPEVENT_ SECURITY_LEVEL
User Name	The user name that is used for connecting to the SNMP agent. The default is `snmpuser`.	KQZ_SNMPEVENT _USER_NAME
Auth Protocol	The authorization protocol that is used to connect to the SNMP agent. It can be one of the following values: • 1=MD5 • 2=SHA The default is 2.	KQZ_SNMPEVENT_AUTH _PROTOCOL
Auth Password	The authorization pass phrase that is used for connecting to the SNMP agent.	KQZ_SNMPEVENT_AUTH _PASSWORD
Priv Password	The privacy pass phrase that is used for connecting to the SNMP agent.	KQZ_SNMPEVENT_PRIV _PASSWORD
Trap configuration file	The location of the trap configuration file.	KQZ_SNMPEVENT_ TRAPCNFG_FILE

Table 25. Java configuration parameters

Parameter name	Description	Parameter name in silent configuration file
Java trace level	The trace level that is used by the Java providers. It can be one of the following values: • 1=Off • 2=Error • 3=Warning • 4=Information • 5=Minimum Debug • 6=Medium Debug • 7=Maximum Debug • 8=All The default is 2.	JAVA_TRACE _LEVEL

Table 26. Proxy server configuration parameters

Parameter name	Description	Parameter name in silent configuration file
Proxy Hostname	The hostname of the proxy server.	KQZ_HTTP _PROXY_HOSTNAME
Proxy Port	The port number of the proxy server. The default is 80.	KQZ_HTTP_PROXY_PORT
Proxy User Name	The user name for the proxy server.	KQZ_HTTP _PROXY_USER
Proxy Password	The password for the proxy server.	KQZ_HTTP _PROXY_PASSWORD

Table 27. HTTP server configuration parameters

Parameter name	Description	Parameter name in silent configuration file
HTTP user name	The user name for accessing the MQ Appliance REST Management interface.	KQZ_HTTP _USER
HTTP password	The password for accessing the MQ Appliance REST Management interface.	KQZ_HTTP _PASSWORD
Certificate Validation Enabled	Whether to enable certificate validation. It can be one of the following values: • 1=true • 2=false The default is 2.	KQZ_HTTP_CERTIFICATE _VALIDATION

Table 28. MQ appliance connection configuration parameters

Parameter name	Description	Parameter name in silent configuration file
Appliance Host or IP Address	The hostname or IP address of the MQ appliance. The default is `https://hostnameoripaddress: https://9.123.123.123`.	KMK_APPLIANCE_HOST_OR_IP_ADDRESS.arm1
Appliance Port Number	The port number for HTTPS connection to the MQ appliance. The default is 5554.	KMK_APPLIANCE_PORT_NUMBER.arm1
Appliance User Name	The user name that is used for connecting to the MQ appliance.	KMK_APPLIANCE_USER_NAME.arm1
Appliance User Password	The password for the MQ appliance user.	KMK_APPLIANCE_USER_PASSWORD.arm1
Agent Host Identification	The hostname of the system where the MQ Appliance agent is running. The default is `9.123.123.111`.	KMK_APM_AGENT_IDENTIFICATION.arm1
Certificate Validation Enabled	Whether to enable certificate validation for HTTP connection. • 1=true • 2=false The default is 2.	KMK_CERTIFICATE_VALIDATION_ENABLED.arm1

Configuring InfoSphere DataStage monitoring

You must configure the DataStage agent so that the agent can collect data to monitor the health and performance of the DataStage server resources.

Before you begin

Review the hardware and software prerequisites, see Software Product Compatibility Reports for DataStage agent

About this task

The DataStage agent is a multiple instance agent. You must create the first instance and start the agent manually.

The product version and the agent version often differ. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the "Change history" on page 54.

Configuring the agent on Windows systems

You can use the IBM Cloud Application Performance Management window to configure the agent on Windows systems.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Template** in the **Task/SubSystem** column, and click **Configure Using Defaults**.

The **Monitoring Agent for DataStage** window opens.

3. In the **Enter a unique instance name** field, type an agent instance name and click **OK**.
4. In the **Monitoring Agent for DataStage** window, specify values for the configuration parameters and click **OK**.

   For information about the configuration parameters, see "Configuration parameters of the agent" on page 294.
5. In the **IBM Performance Management** window, right-click the agent instance that you created and click **Start** to start the agent.

Configuring the agent on Linux systems

To configure the agent on Linux operating systems, you must run the script and respond to prompts.

Procedure

1. On the command line, change the path to the agent installation directory.
   Example: `/opt/ibm/apm/agent/bin`
2. Run the following command where instance_name is the name that you want to give to the instance:

   ```
   ./datastage-agent.sh config instance_name
   ```
3. When the command line displays the following message, type 1 and enter:

   ```
   Edit 'Monitoring Agent for DataStage' setting? [1=Yes, 2=No]
   ```
4. Specify values for the configuration parameters when you are prompted.

   For information about the configuration parameters, see "Configuration parameters of the agent" on page 294.
5. Run the following command to start the agent:

   ```
   ./datastage-agent.sh start instance_name
   ```

Configuring environment variables

You can configure environment variables to change the behavior of the DataStage agent.

Procedure

1. Open the following file in a text editor:

   a) Windows *install_dir*\TMAITM6_x64\KDTENV_*instance_name*

   b) Linux *install_dir*/config/.dt.environment
2. Edit the following environment variables:

   - **KDT_FIRST_COLLECTION_INTERVAL**: The time interval in seconds for first data collection. Set this time interval to a duration by which the agent would collect previous Job runs data in the specified time until the agent starts. The default value is 300 seconds (5 minutes). So, if the agent starts at 2:00 PM, it collects the Job runs data from 1:55 PM to 2:00 PM. This is to avoid data storm of historical job runs when the agent starts collecting data. All subsequent agent data collection for job runs fetch only the newly added job runs that took place since the last collection.
   - **KDT_SSL_CONTEXT**: The SSL protocol that is enabled on the Service Tier (WebSphere Application Server). The default value is TLS.
   - **KDT_META_SCHEMA_NAME**: The name of database schema that is created for the metadata repository. The default value is DSODB for Db2 and xmeta for MSSQL and Oracle databases.
   - **KDT_DATABASE_SERVICE_NAME**: The database or service name that is used by the agent to connect to the metadata repository. The default value is XMETA for Db2, xmeta for MSSQL, and ORCL for Oracle databases.
   - **KDT_DISABLED_ATTRIBUTEGROUP**: A comma-separated list of attribute groups whose data collection needs to be unavailable. Following values can be set as single or multiple for respective

attribute group: JobRuns, JobProperties, JobRunLog, JobStages, JobParameters, EngineSystemConfiguration, EngineSystemResources, EngineServiceStatus, EngineStatusSummary, JobActivity, AgentConfiguration, and JobConfiguration.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

You can use the silent response file to configure the DataStage agent on Linux and Windows system. After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

1. In a text editor, open the silent config file that is available at the following location and specify values for all the parameters:

   Windows `install_dir\samples\datastage_silent_config.txt`
   Linux `install_dir\samples\datastage_silent_config_UNIX.txt`

   Windows `C:\IBM\APM\samples`

   Linux `/opt/ibm/apm/agent/samples`

   For information about the configuration parameters, see "Configuration parameters of the agent" on page 294.
2. On the command line, change the path to *install_dir*\bin.
3. Run the following command:

   Windows `datastage-agent.bat config instance_name install_dir\samples \datastage_silent_config.txt`

   Linux `datastage-agent.sh config instance_name install_dir\samples \datastage_silent_config_UNIX.txt`
4. Start the agent.

   Windows In the **IBM Performance Management** window, right-click the agent instance that you created, and click **Start**.

   Linux Run the following command: `./datastage-agent.sh start instance_name`

What to do next

Log in to the Cloud APM console to view data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

For help with troubleshooting, see the Troubleshooting section.

Configuration parameters of the agent

While configuring the DataStage agent, you can change the service tier, metadata repository, and advanced configuration parameters.

Service tier configuration parameters

The configuration parameters that are required for the agent to connect to the service tier.

The following table contains detailed descriptions of the service tier configuration parameters of the DataStage agent.

Table 29. Names and descriptions of the service tier configuration parameters		
Parameter name	**Description**	**Mandatory field**
Hostname	Hostname of the computer where service tier is installed. If the computer is part of a domain then provide fully qualified domain name (FQDN).Default Value is localhost.	Yes
HTTPS Port	HTTPS port for REST interface on the computer where service tier is installed. Default Value is 9443.	Yes
WAS Username	The user name for connecting to WebSphere Application Server. Default Value is wasadmin.	Yes
WAS Password	The password for connecting to WebSphere Application Server.	Yes
Confirm WAS Password	The password that is specified in the **WAS Password** field.	Yes

Metadata repository configuration parameters

The configuration parameters that are required for the agent to connect to the metadata repository.

The following table contains detailed descriptions of the metadata repository configuration parameters of the DataStage agent.

Table 30. Names and descriptions of the metadata repository configuration parameters		
Parameter name	**Description**	**Mandatory field**
Database Type	Database type of the metadata repository. Db2Default Value is 1.	Yes
Hostname	Hostname of the computer where metadata repository is installed. If the computer is part of a domain then provide fully qualified domain name (FQDN).Default Value is localhost.	Yes
Database Port	Database port on metadata repository for JDBC Connection. Default Value is 50000.	Yes
Database Username	The username for connecting to operations database. Default Value is dsodb.	Yes
Database Password	The password for connecting to operations database.	Yes
Confirm Database Password	The password that is specified in the **Database Password** field.	Yes
JDBC Driver Path	Path to the JDBC driver including jar file. For example,/home/jars/db.jar on Linux.	Yes

Advanced configuration parameters

Table 31. Names and descriptions of the advanced configuration parameters		
Parameter name	**Description**	**Mandatory field**
Java trace level	The tracel levels that are used by the Java Custom providers. Default Value is 2.	Yes

Java API Client Configuration parameters

Table 32. Names and descriptions of the Java API Client Configuration parameters

Parameter name	Description	Mandatory field
Class path for external jars		No

Configuring Internet Service Monitor

The Internet Service Monitoring Agent offers the capability to determine whether a particular service performs adequately, identify problem areas and report service performance measured against Service Level agreements. Internet Service Monitoring agent works by emulating the actions of a real user. It regularly polls or tests Internet services to check their status and performance.

Overview

When monitoring Internet services, you define what is to be monitored, for whom, and when. You can configure the monitors through Internet Service Monitoring agent configuration user interface.

The monitor test specific Internet services and forward the results of the tests to the Databridge. The monitors emulate the actions of a real user of the service.

For example, the HTTP monitor periodically attempts to access a web page by emulating requests that a web browser would usually send when a user goes to the page. The monitor records the result of the test, which is sent to the Databridge.

Internet Service Monitoring

Each monitor is designed to test one type of protocol or service. For example, the HTTP monitor tests the availability of resources such as web pages over the Hypertext Transfer Protocol, and the FTP monitor tests the transfer of files between hosts running the File Transfer Protocol.

A monitor can test many different instances of the same service, such as a series of web pages served by a range of hosts.

Web Service Monitoring

Using the Internet Service Monitoring range of monitors, you can tailor the type of web service monitoring you provide from basic Internet service monitoring testing the availability of a web page, to combining sequences of tests.

Internet service monitoring uses high volume, low complexity polling to test the availability of web services. For example, if you want to monitor the general availability of a website, you might use the HTTP monitor to poll many URLs at regular intervals.

Using a combination of monitors, you can build a level of service monitoring appropriate to your requirements:

- HTTP and HTTPS monitors

  Monitor the availability of resources over HTTP or HTTPS by running basic, single-request tests at high volume.

- Transaction monitor (TRANSX)

  Combine sequences of tests carried out by a group of monitors, simulating the actions of a real user. For example, dialing up a service, accessing a number of pages on several websites and then accessing email services.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the "Change history" on page 54.

Configuring Internet Service Monitoring through user interface

To monitor Internet services, create user profiles, profile elements, and monitoring schedules. Configure user profiles, profile elements, and monitoring schedules by using the Internet Service Monitoring user interface.

About this task

A user profile is a customer, or a department, or a group of services for which you monitor internet or web services. For each user profile, user needs to define one or more profile elements. For example, user might define a profile element to monitor a web page delivered over an HTTP service, or a profile element to monitor the availability of an FTP service. User profiles typically contain multiple profile elements, each profile element tests one of the services provided to that user. Each user profile also has an associated monitoring schedule that determines on which day and at what time the tests that are defined in the profile are to run.

To access the Internet Service Monitoring Agent's configuration window through IBM Application Performance Management dashboard use the following method:

Procedure

1. On the Application Performance Management dashboard click Icon. Click **Agent Configuration**, Agent configuration window opens.
2. Click **ISM** tab to configure Internet Service Monitoring Agent.

   You can Create, Edit, Delete, Refresh, Schedule, and filter the user profiles. Deploy the created user profiles on selected managed system. The version displayed is the version of the managed system. Profile name indicates the user profile deployed against the selected managed systems. Follow these steps to configure any of the profiles and deploy on the managed systems.
3. To add a profile click ⊕ icon. Enter the **Profile Name** and **Description** in the dialog box.
4. Click **Next**.
5. Select a monitor from the monitors dropdown list and click **Next**.
6. Provide the values of the fields and click **Add**.

   Multiple monitors can be selected for a profile. See “Available Internet Service Monitoring monitors” on page 300 for available monitors.
7. Click **Done**.
8. Click ↻ icon.
9. In the **Filter** field, search the user profiles by its name.
10. To deploy the profile that is created on a managed system, select the check box of the created profiles, which are to be configured and select a managed system. Click **Deploy** to deploy the profile on the selected managed system name.

Editing profile

All the user profiles that are created are editable.

About this task

Use the following procedure to edit the profiles.

Procedure

1. Select **profile name** and click ✎.
2. Select a service to edit and click **Edit**.

   a. Add a monitor by using ⊕ icon and delete monitor by using ⊖

   b. To rename a profile, double-click **Profile Name** text field, modify the profile name and click **Rename Profile**.
3. Edit the values for the selected service.

   Note:

   - To enable password field for editing, double-click **username** text field. User can edit or add username, and change the password.
   - To enable **sslkeypassword** field for editing, double-click its text to change the secret key.
4. To Activate or Deactivate a monitor element for a particular profile, refer the following steps:
   - Open a profile in edit mode.
   - Select the check boxes for monitor elements that you want to activate or deactivate and click on **Activate** or **Deactivate** buttons respectively.
   - Click **Ok** from **Click on refresh icon to reflect the change** dialog box.
   - Click ↻.
   - Verify the changes in the **Active** field. If monitor element is activated the value of **Active** field will be `True`. For Deactivated monitor elements value will be `False`.
5. To delete multiple monitor elements, select the check-boxes next to monitor elements in the profile edit grid and click on ⊖
6. Click **Save** and click ↻ icon to refresh.

Scheduling a profile

The profiles that are created can be scheduled to deploy on a particular date and time.

About this task

Use the following procedure to schedule profiles.

Procedure

1. Select **profile name**.
2. Click **Schedule** button.
3. Schedule the profile by selecting day against time. User can drag the grid to select any wanted time.
4. Click **Save**.
5. Click ↻ icon to refresh.

Deleting a profile

The profiles that are created can be deleted permanently.

About this task

Use the following procedure to delete the profile.

Procedure

1. Select **profile name**.
2. Click ⊖ icon to delete the profile.

OID Groups

Object Identifier (OID) groups are optional monitor specific parameters. They define sets of one or more OIDs of a device's Management Information Base (MIB) objects. The SNMP monitor uses the OID groups to retrieve data from those MIB objects whose OIDs appear in a specified OID group.

The details of the MIB objects from which the monitor extract data are as follows:

- OID Value

  The numerical identifier of the MIB object instance expressed using either ASN.1 notation, for example `.1.3.6.1.2.1.1.2.0`, or the object's name, for example `sysObjectID.0`

  Note: When using ASN.1 notation, you must include the leading `.` character in the OID.

  Note: You may only use an object's instance name to specify the OID value if the MIB document that defines the name is accessible by the monitor. The default directory for MIB documents is `$ISHOME/mibs`.
- OID Name

  The name of the MIB object, for example `sysObjectID`. This name is used in service level classifications and in `$oidName`*n* monitor elements.
- OID Unit

  The units of the data contained in the MIB object. For example, seconds, bytes, or bits per second (BPS). Set to BPS to enable bits per second calculation for the OID. Bits per second values are calculated as:

  ```
  current_poll_value - prev_poll_value) / poll_interval * 8
  ```
- Selector

  The index value of the MIB object. The following table shows an example that results in the selector searching all the `ifDescr` rows for the value `FastEthernet0/1`, giving a row index of 2. Then the row `ifPhysAddress.2` is queried and the value `0:6:53:34:d2:a1` is returned. In this way the index 2 is not directly specified, so if the index for `FastEthernet0/1` changes, the OID groups do not need to be re-configured.

Table 33. Use of the index value

MIB object	MIB object value
OID value	`ifPhysAddress`
OID name	`FastEthernet0/1PhysicalAddress`
OID unit	`string`
Selector	`ifDescr=FastEthernet0/1`

Creating OID group and MIB object

OID groups are created globally and can be used by all user profiles that monitor SNMP-enabled devices

Procedure

Complete the following steps to create an OID group and MIB object.

1. Click the **OIDs** button to create an OID group on the Internet Service Monitoring Agent dashboard.
2. Click ⊕ icon and enter the OID group name in the **OID Group Name** field.
3. Click ⊕ icon to add the MIB object.
   a. Enter the **value**, **Name**, **Unit** and **selector** for the MIB object.
   b. Click **Add**.

   MIB object is created successfully.
4. Click ↻ icon to refresh.

   OID group is created successfully
5. Select an **OID Group Name** and click **View** to see the list of all the MIB objects created under the selected **OID Group**.

6. Click **Close**.

Editing OID group and MIB object

You can edit the OID groups. The MIB objects can also be edited while creating the OID group or after creating the OID group.

Procedure

Complete the following steps to edit an OID group.

1. Click the **OIDs** button, to edit an OID group on the Internet Service Monitoring Agent dashboard..
2. Select the OID group name from the **OID Group Name** list and click ✎ icon.
3. Select the value from the **Edit OID Group** and click ✎ icon.
4. Modify the MIB object fields according to your requirement and click **Save** button.
5. Click **Save** on **OID Groups** pop-up page.
6. Click **Close**.

Deleting OID group

MIB objects are contained in OID groups and used by the SNMP monitor to obtain data. You can delete individual MIB objects from an OID group or delete all MIB objects by deleting the entire OID group.

Procedure

Complete the following steps to delete an OID group.

1. To delete the ODI group, click the **OIDs** button on the Internet Service Monitoring Agent dashboard.
2. Select the OID Group Name from the **OID Group Name** list and click ⊖ icon.

   The OID group along with the MIB object is deleted.
3. Click **Close**.

Deleting MIB object

Procedure

Complete the following steps to delete the MIB group.

1. To delete the MIB object, click the **OIDs** button on the Internet Service Monitoring Agent dashboard.
2. Select the OID Group Name from the **OID Group Name** list and click ✎ icon.
3. Select the MIB object value and click ⊖ icon.

   MIB object is deleted.
4. Click **Save** on the **Edit OID Group** pop-up page.
5. Click **Close**.

Available Internet Service Monitoring monitors

The Internet Service Monitoring agent a suite of monitors that cover a broad range of Internet services.

The following table lists the monitors available in Internet Service Monitoring agent and the types of service that it monitors.

Table 34. Available Internet service monitors

Monitor Name	Type of service monitored
DHCP	Dynamic Host Configuration Protocol. To configure DHCP, see "DHCP Monitor" on page 313.

Table 34. Available Internet service monitors (continued)

Monitor Name	Type of service monitored
DNS	Domain Name Service. To configure DNS, see“DNS Monitor” on page 315.
FTP	File Transport Protocol. To configure FTP, see “FTP Monitor” on page 320.
HTTP	HyperText Transport Protocol. To configure HTTP, see“HTTP Monitor” on page 324 .
HTTPS	HyperText Transport Protocol (Secure). To configure HTTPS, see “HTTPS Monitor” on page 333.
ICMP	Internet Control Message Protocol. To configure ICMP, see“ICMP Monitor” on page 338.
LDAP	Lightweight Directory Access Protocol. To configure LDAP, see “LDAP Monitor” on page 343.
IMAP4	Internet Message Access Protocol. To configure IMAP4, see “IMAP4 monitor” on page 349.
NTP	Network Time Protocol. To configure NTP, see “NTP Monitor” on page 354.
NNTP	Network News Transport Protocol. To configure NNTP, see “NNTP Monitor” on page 356.
POP3	Post Office Protocol. To configure POP3, see “POP3 Monitor” on page 361.
RADIUS	Remote Authentication Dial-In User Service. To Configure RADIUS, see “RADIUS Monitor” on page 366.
RPING	Remote Ping (Cisco, Juniper, and RFC2925). To configure RPING, see “RPING Monitor” on page 371.
RTSP	Real-Time Streaming Protocol. To configure RTSP, see “RTSP Monitor” on page 376.
SAA	Cisco Service Assurance Agent. To configure SAA, see “SAA monitor” on page 382.
SIP	Session Initiation Protocol. To configure SIP, see “SIP Monitor” on page 397
SMTP	Simple Mail Transport Protocol. To configure SMTP, see “SMTP Monitor” on page 402.
SNMP	Simple Network Management Protocol. To configure SNMP, see “SNMP Monitor” on page 406.

Table 34. Available Internet service monitors (continued)

Monitor Name	Type of service monitored
SOAP	XML-based messaging protocol. To configure SOAP, see "SOAP Monitor" on page 411.
TCPPort	Transmission Control Protocol. To configure TCPPort, see "TCPPort Monitor" on page 415
TFTP	Trivial File Transfer Protocol. To configure TFTP, see "TFTP Monitor" on page 419.
TRANSX	Transactions. To configure TRANSX, see "TRANSX Monitor" on page 424.

Files

Executable File

Each Internet service monitor consists of an executable file, properties file, rules file, and a log file.

Monitor executable files are located in the `$ISHOME/platform/arch/bin` directory. The value for `arch` is the architecture code for the Windows - `win 32` operating system.

Properties File

The properties file is a text file and includes default settings that are preceded by the hash symbol.

To change a setting, either change the default setting and remove the hash symbol or copy and paste the line that contains the default settings, make the change and remove the hash symbol. This enables you to restore the defaults later. Monitor properties files are located in the `$ISHOME/etc/props` directory.

Rules File

Rules files are similar to IBM Application Performance Management Netcool/OMNIbus probe rules files. For information about their syntax, see the *IBM Application Performance Management Netcool/OMNIbus Probe and Gateway Guide*.

Monitor rules files are located in the `$ISHOME/etc/rules` directory.

Log file

Log files store messages about the monitor's operation.

Monitor log files are located in the `$ISHOME/log` directory. The `MessageLog` property determines the location and name of the log file. The `MessageLevel` property selects the level of information that is written to the log file, for example, detailed debugging messages or unrecoverable error messages. The `MaxLogFileSize` property determines the size of the log file before rolling over.

The default name of the log file is `name.log` where *name* is the name of the monitor.

Common features

There are a number of features that are common to all Internet service monitors. These features consist of properties, results produced by the monitors, and status messages.

This section describes the properties of all monitors. Monitor specific properties are described in the individual monitor sections.

In the following table, the default property parameters are underlined where applicable.

Table 35. Common properties

Property name	Property parameter	Description
AddRoute	0|1	Creates a route from the IP address of the network interface used by the monitor to the IP address of the monitored host. 0 - disabled 1 - enabled (monitor uses the route specified in the profile element, and not over another network interface). **Note:** This property isn't supported on AIX and HP-UX platforms.
BridgeIPAddress	not applicable	Specifies the IP address of the Databridge. This property isn't configurable; the Databridge is always on the local host.
BridgePort	integer	Port number used by the Databridge. Set this property to the same value as the Databridge SocketPort property. Default: 9510
BridgeSSLAuthenticatePeer	0|1	If you want to configure SSL authentication between the monitor and bridge, or between the bridge and the agent, set BridgeSSLAuthenticatePeer to 1 and restart the bridge. This action authenticates the certificates from the server. Certificates are stored in the BridgeSSLTrustStore. 0 - disabled 1 - enabled
BridgeSSLCertificateFile	string	Specifies the path and filename of the digital Bridge SSL certificate. Default: $ISHOME/certificates/monitorCert.pem
BridgeSSLCipherSet	string	Specifies a CipherSet. If you update this value, use the Cipher syntax defined in the OpenSSL documentation. **Note:** Set the same value on the Internet service monitoring agent, all monitors, and the Databridge. Default: RC4:3DES:DES:+EXP

Table 35. Common properties (continued)

Property name	Property parameter	Description
`BridgeSSLDisableSSLv2`	<u>0</u>|1	Determines which types of sockets are accepted. • If set to 0, both SSLv2 and SSLv3 are accepted. • If set to 1, sockets are opened in SSLv3 only. **Restriction:** Set the same value on the Internet service monitoring agent, all monitors, and the Databridge.
`BridgeSSLEncryption`	0|<u>1</u>	Enables Bridge SSL encryption. Set this property to the same value as the corresponding Databridge property. 0 - disabled 1 - enabled **Note:** Set the same value for all monitors.
`BridgeSSLKeyFile`	string	The path and the filename of the Bridge SSL private key file. Default: `$ISHOME/certificates/monitorKey.pem`
`BridgeSSLKeyPassword`	string	The password used to encrypt the Bridge SSL private key. Default: `tivoli`

Table 35. Common properties (continued)

Property name	Property parameter	Description
`BridgeSSLTruststore`	string	The path and file name of the Trusted certificate file for authentication. This is only required when using the `BridgeSSLAuthenticatePeer` setting. If you want to configure SSL authentication between the monitor and bridge, or between the bridge and the agent, set `BridgeSSLAuthenticatePeer` to `1` and restart the bridge. This action authenticates the certificates from the server. You can store certificates in both the `SSLTrustStoreFile` and the `SSLTrustStorePath`. Defaults: • `SSLTrustStoreFile`, `$ISHOME/certificates/trust.pem` • `SSLTrustStorePath`, `$ISHOME/certificates/` To add new certificates, complete one of the following steps: • Add a certificate to the end of the list in the `SSLTrustStoreFile` text file. • Add a certificate to the `SSLTrustStorePath` directory, and run the OpenSSL `c_rehash` *`certificate_dir`* command to hash the certificates.
`BridgeTimeout`	integer	Time, in seconds, that the monitor waits for a response from the Databridge.
`ConfigFile`	string	Use to point to a monitor configuration file. Default: blank (empty string).
`ConfigurationCheckInterval`	integer	The interval (in seconds) at which the monitor checks for changes to the profile. Default: `1`
`Datalog`	`0`|`1`	Forces the monitor to log performance data in a datalog file. The performance data is logged in: `$ISHOME/datalogs/`*`userprofile`* `0` - disabled `1` - enabled

Table 35. Common properties (continued)

Property name	Property parameter	Description
`DatalogFormat`	string	Defines the format of the datalog file. The parameter is a space-separated list of elements, the values of which should be stored in the datalog file. For each poll result written to the datalog file, the current time (`$time`) and the time taken (`$totalTime`) are logged, followed by all the elements defined in this property.
`DatalogNameFormat`	string	Format of the datalog filename.
`Domain`	string	Specifies the domain name of the host running the monitor. If this property is not set, the monitor attempts to guess the domain name using the NIS and DNS configurations.
`DumpProps`	not applicable	Displays a list of all properties for a monitor.
`FullHostInfo`	0|1	Specifies whether to map the `$host` element to an IP address element `$hostIP` (if `$host` is a DNS name) or to a DNS name element (if `$host` is an IP address). `0` - disabled `1` - enabled **Note:** This isn't available in the TRANSX monitor.
`GroupID`	string	The group ID that the monitor should be run as.
`Help`	0|1	Displays the help for command-line options without running the monitor. `0` - disabled `1` - enabled
`IdentifierChecksumFields`	string	Deprecated.
`IgnoreUnmatchedDVC`	0|1	If a particular service level classification isn't matched, element not created by the monitor, then ignore that element in the service level calculation. **Tip:** In earlier releases of Internet Service Monitoring agent, service level classifications were called Discrete Value Classifications (DVCs). `0` - disabled `1` - enabled

Table 35. Common properties (continued)

Property name	**Property parameter**	**Description**
IpAddress	string	Specifies the IP address of the network interface that the monitor uses during tests. If this property isn't set, the monitor attempts to determine the host machine's IP address using a host name lookup. This attempt may fail if the host machine has more than one network interface.
Manager	string	Specifies the name of the management application, which is used in ObjectServer event de-duplication.
MaxCCA	integer	Sets the maximum number of concurrent connections that the monitor can have at any one time. Note that if you set this value too high, you might severely affect the performance of the monitor. This property isn't available for the ICMP monitor. Default: 10
MaxLogFileSize	integer	The maximum size (in bytes) of the log file. Default: 1048576
MessageLevel	string	The lowest level of messages to be sent to the message log. Values, in ascending order of severity are: debug, info, warn, error, and fatal. Default: warn
MessageLog	string	Location of the log file. Default: $ISHOME/log/*monitor*.log
MinPoll	integer	Defines the minimum polling interval allowed. If any of the monitor configuration files have a poll interval set less than this value, the value in the configuration file is overridden. Default: 60
MsgDailyLog	integer	Enables generation of a daily log file. Default: 0 - Daily log disabled
MsgTimeLog	string	Specifies the time (in 24-hour format HHMM) after which the monitor generates a daily log, if MsgDailyLog is enabled. Default: 0000 - 12 midnight

Table 35. Common properties (continued)

Property name	**Property parameter**	**Description**
Name	string	The name of the monitor. Setting this property resets the PropsFile, RulesFile and MessageLog properties to their defaults.
NewProfileCheckMultiple	integer	Multiple that indicates how often the monitor checks for new configuration files when checking for profile changes. Default: 10
NoRecover	integer	Instructs the monitor not to recover the store and forward file. Default: 0 - recovery isn't suppressed
Pause	integer	Sets the interval (in seconds) at which a monitor spawns threads. Setting this property to higher values, such as 100 or more, forces the monitor to spawn threads at a slower rate. Increasing the value is generally only necessary on slow systems. This property isn't supported on the ICMP monitor. Default: 50
PreviousFields	string	Elements specified by this property (using the form "<element>, <element>, …") are stored for one poll and prefixed with the string previous.
Profile	string	The name of the customer profile, or profiles to use. The string can be a single profile name, a space separated list of profile names, or *, which forces the monitor to use all available profiles. Default: *
ProfileUpdateTimeout	integer	The number of milliseconds that a profile file must remain static before it can be read by a monitor and updated. The allowable range is 1-20000 milliseconds. Default: 100
PropsFile	string	The name of the properties file. Default: $ISHOME/etc/props/*monitor*.props
QFile	string	Sets the name of the store and forward file. Default: $ISHOME/var/*monitor*.saf.

Table 35. Common properties (continued)

Property name	Property parameter	Description
QSize	integer	Sets the reserved size (in bytes) of the store and forward file. Default: 10240000
UserID	string	The user ID that the monitor should be run as. **Note:** Don’t use this property with the DHCP monitor.
Version	not applicable	Prints the monitor version without running the monitor.

Common monitor elements

This section describes the elements produced by all monitors. Monitor specific elements are described in the individual monitor sections. Produced elements can be viewed in the Internet Service Monitoring Agent dashboard.

If you use IBM Application Performance Management, the elements that can be viewed on the agent's dashboard as attributes are determined by a mapping file generated by the Internet service monitoring agent. This mapping file isn’t configurable.

Table 36 on page 309 lists the elements produced by all monitors. Elements indicated by an asterisk (*) are available as workspace attributes. The names of the attributes are shown within brackets. Absence of an asterisk indicates there’s no equivalent workspace attribute. Attributes shown in brackets but without an element indicates that they are only available as workspace attributes, there is no equivalent element.

Table 36. Common monitor elements

Element name	Element description
$consecutiveFailures	If $failureRetests is nonzero and the test fails according to the service level classification, this element is created starting with the value of 1. The value increases until either the test no longer fails, at which point $consecutiveFailures is set to 0, or until the following poll. If, at this poll, the service level is passed or starts increasing again, the element is no longer created. If the value of this element exceeds the value of $failureRetests, the value of $consecutiveFailures is reset to 1. **Note:** The TRANSX monitor doesn’t generate this element.
$datalogPath* (guid)	The path to the datalog file used by the monitor. The workspace attribute uses the last 100 characters of the path.
$description* (Description)	Contains the text description provided in the **Description** field of the monitor profile element.

Table 36. Common monitor elements (continued)

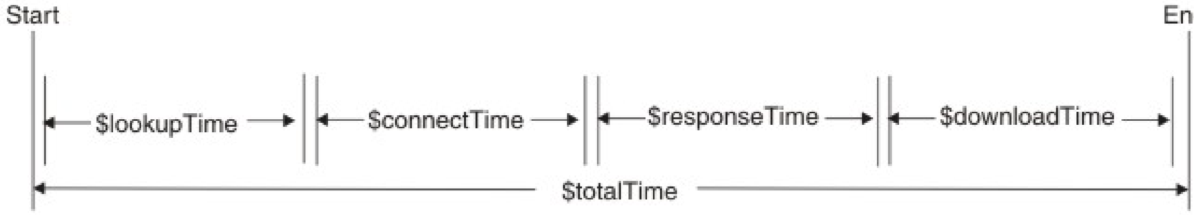
Element name	Element description
`$failureRetestInterval`	The poll interval used during failure retesting. This is only valid if `$failureRetests` is greater than `0`. If the retest interval is greater than the normal poll interval, it’s set equal to the normal poll interval. **Note:** The TRANSX monitor doesn’t generate this element.
`$failureRetests`	The number of service level failures that has to be exceeded before a failed event is recorded and sent to the ObjectServer. **Note:** The TRANSX monitor doesn’t generate this element.
`$host*` (`Host`)	The name of the host or server. Stored in the configuration file.
`$hostName`	Contains the host name of the `$host` element (if `$host` is an IP address).
`$hostIP`	Contains the host IP of the `$host` (if `$host` is a DNS name).
`$identchecksum*` (`Identchecksum`)	The identifier of the profile element.
`$lastServiceLevel*` (`LastServiceLevel`)	The service level number of the previous poll. This is cleared if the profile changes.
`$lastServiceLevelCounter`	The `serviceLevelCounter` in the previous poll. This is reset if the profile changes.
`$monitorDNSdomain`	The domain name of the machine running the monitor, as used by DNS.
`$monitorHost*` (`MonitorLocation`)	The name of the host running the monitor.
`$monitorNISdomain`	The domain name of the host running the monitor, as used by NIS (Network Information Service).
`$monitorDomain`	Overrides the `$monitorDNSdomain` and `$monitorNISdomain` settings.
`$message*` (`ResultMessage`)	A text string describing the result of the poll. For example, `Connection failed`, `OK`, or `Success`.
(Node)	The name of the system on which Internet Service Monitoring is running. This attribute is added by the Internet service monitoring agent.
`$pollInterval`	The poll interval specified in each monitor.

Table 36. Common monitor elements (continued)

Element name	Element description
$resultString* (ResultString)	A text string indicating the service level classification applied to the results of the poll. For example, TotalTime > 20.
$service* (Service)	The name of the service being monitored. For example, FTP or HTTP.
$serviceLevel* (ServiceLevel)	The service level number of the poll, as defined in the service level classification: 0 - Unknown 1 - Good 2 - Marginal 3 - Failed
$serviceLevelCounter	The number of times that the service level number has remained unchanged.
(ServiceLevelString)	The string associated with the returned service level (Unknown, Good, Marginal, or Failed).
$startTimePoll	The time when the poll started.
$time	The UNIX time, in seconds, when the poll occurred.
$timeStamp* (Timestamp)	The date and time when the test was performed. The timestamp format uses local settings.
$transxName	The name of the transaction. This is produced by a monitor if the monitor is used in a transaction.
Profile Details	
$profile* (IsmProfile)	The name of the user profile.
Timings - for information about how timings are measured, see "Time Calculations" on page 311.	
$timeout	The number of seconds in which the server must respond. Taken from the configuration file.
$totalTime* (TotalTime)	The total time taken to run an operation in seconds. This includes all lookup, connect, and download times where applicable, and interim processing time.

Time Calculations

Monitors attempt to divide the time taken to complete a poll into different timed stages. For example, this could include the time taken to obtain a host's IP address, or the time taken to successfully connect to a host.



The `$totalTime` is always slightly longer than the sum of the other times, because it includes the overhead incurred by the monitor's activities such as processing received data and performing system calls. `$totalTime` is measured in seconds.

Status messages
Monitors return the generated status messages after each service test. Status messages indicate the outcome of tests.

Messages generally originate from the monitored service or the network environment outside the monitor. Table 37 on page 312 describes the common status messages returned by monitors in the `ResultMessage` attribute when using IBM Application Performance Management. Monitor specific status messages are described in the individual monitor sections.

In addition to the messages provided by the individual monitors, some monitors such as the HTTP monitor, report messages for the underlying operating system. For example, if the TCP connection fails, Internet Service Monitoring agent uses the string defined by the operating system, such as **`connection refused`**, **`timeout`**, **`network unreachable`**, and other strings.

Table 37. Common status messages

Message	**Description**
`OK`	The request from the monitor succeeded. Monitors may have other status messages that indicate that a process was successful. See the *Status Messages* section for each monitor.
`Received response to request not originating from this monitor - ignored`	Received a reply from the server to a message that didn’t originate from the designated monitor.

`Connection failed`
`Connect to server failed`
	The monitor failed to connect to the server. See the log file for more information.

`Connection closed unexpectedly`	The connection to the server was broken.
`Connection timed out`	The connection succeeded, but then the server stopped responding.
`Connection closed by foreign host`	The remote host closed the connection before the monitor expected.
`Timed out waiting to read/ write`	A data connection to the monitored server was established, but it has stopped responding.
`No Response from server`	Request timed out.

Table 37. Common status messages (continued)

Message	Description
Format Error Server Failure No such host or domain Not Implemented Request refused Unknown Error	Error returned by the monitored server.
Network is down Network is unreachable Network dropped connection on reset Software caused connection abort Connection reset by peer Connection timed out Connection refused Host is down No route to host Remote peer released connection	There's a problem with the network.

DHCP Monitor

The DHCP monitor checks the availability and response time of DHCP servers.

You assign service level classifications according to the time it takes for the DHCP server to respond to a request from the DHCP monitor, by using either the total, lookup, or response time.

Table 38. DHCP monitor files

Monitor files	Name or Location
Monitor executable	`nco_m_dhcp`
Properties file	`$ISMHOME/etc/props/dhcp.props`
Rules file	`$ISMHOME/etc/rules/dhcp.rules`
Log file	`$ISMHOME/log/dhcp.log`

Guidelines for Configuring the DHCP monitor

The DHCP monitor tests DHCP services by acting as a limited DHCP client. It sends a DHCP INFORM request to the target DHCP server on the same network by using UDP as the transport protocol over an established connection, and awaits a corresponding DHCP ACK from the server. The monitor doesn't request an IP address, or affect the expiry on existing IP addresses.

Note: Monitored DHCP servers must support DHCP INFORM requests and be RFC2131 compliant.

The DHCP monitor must run as `root` because it binds to a port less than 1024.

Limitation

The DHCP monitor can’t use any network interface that is configured by using a DHCP client. Instead, configure the monitor to use a network interface whose IP address isn’t assigned dynamically.

Configuring the DHCP Monitor Service Test

Table 39. DHCP monitor configuration

Field	Description
server	The host name of the DHCP server. Example is dhcp1.mycompany.com
localip	The IP address network interface that the monitor uses to perform the test. Example is 192.168.n.n
description	A text field for providing descriptive information on the element. Example is DHCP monitor
port	The port number of the DHCP server and the default value is 67.
localport	The port number that the monitor uses to perform the test and the default value is 68.
timeout	The time, in seconds, to wait for the server to respond and the default value is 30.
retries	The number of times the monitor must retry the connection to the DHCP server before quitting and the default value is 0. Example is 2.
poll	The time, in seconds, between each poll of the server by using the current profile element and default value is 300.
failureretests	The number of times to retest before indicating a failure and the default value is 0.
retestinterval	The time, in seconds, to wait between each retest on failure and the default value is 30.

In addition to the test results common to all elements, the DHCP monitor generates a set of test results that contain data specific to DHCP service tests.

Table 40. DHCP monitor elements

Element	Description
`$clientIP*ClientIp`	The IP address of the host where the monitor is running.
`$lookupTime*(LookupTime)`	The time taken to obtain the IP address of the host server.
`$responseTime*ResponseTime`	The time between when the connection is established and the first byte of data is received.
`$retries`	The maximum number of retries, as specified during the monitor configuration.
`$router`	The IP address of the router as returned by the DHCP server.

Status messages

The DHCP monitor provides status messages in the `ResultMessage` attribute when using IBM Application Performance Management. These messages indicate the result of the test.

Table 41. DHCP monitor status messages

Message	Description
`Received DHCPACK Received DHCPNAK`	A DHCP server responds to the DHCP inform request sent by the monitor.
`This monitor requires root privileges to run`	Log in as root.
`Did not receive valid DHCP MESSAGE`	Unrecognized response from the DHCP server.
`Did not receive valid DHCP MESSAGE TYPE`	Unrecognized response from the DHCP server (DHCPACK or DHCPNAK expected).

`Invalid transaction ID`
`Received response to request not originating from this monitor - ignored`
	Received a reply from a DHCP server to a message that didn't originate from this monitor.

`Unexpected op-code returned`	Received an unexpected message on this port.
`Connection failed`	The server name that is specified is invalid.
`Failed to send request to DHCP server`	The operating system can't identify specifically why the request might not be sent to the server so returns this status message that indicates a problem with the network.
`No Response from server`	The DHCP server isn't responding.

DNS Monitor

The DNS monitor uses the DNS (Domain Name System) service to find information about one or more hosts.

The DNS monitor uses either the IP address of the host to search for the host name, or the host name to search for the IP address. The monitor measures the performance of the service by recording the result of the search and the response times. The monitor also records details about each query sent to the server.

Table 42. DNS monitor file summary

Monitor files	Name or Location
Monitor executable	`nco_m_dns`
Properties file	`$ISMHOME/etc/props/dns.props`
Rules file	`$ISMHOME/etc/rules/dns.rules`
Log file	`$ISMHOME/log/dns.log`

Guidelines for configuring DNS Monitor

The DNS monitor can be configured to look up the IP address or host name of the target host. Depending on the type of lookup, the monitor communicates with the DNS server in a different way.

IP address lookup

When performing an IP address lookup test, the monitor is given a host name, which it uses to locate an IP address.

The monitor tests the DNS as follows:

1. The monitor queries the DNS server by using the fully qualified host name of HostA (`hosta.dev.net`) to request its IP address.

   If the DNS server can locate the IP address of the host, it returns it to the monitor. If the DNS server can't locate the IP address of the host, it returns a message containing details of the failed search to the monitor.

   If the request is timed out, the monitor will retry (if retries are configured). If there are no retries, the monitor will create a failed event.

   If the host name specified in the configuration is a domain name, such as `mycompany.com`, rather than a fully qualified host name, such as `hostx.mycompany.com`, the monitor retrieves information about the whole domain. This information is stored in two extra elements: `$domainNameServer` and `$domainNameAddr`.
2. If the message returned to the monitor contains a canonical name, the monitor concludes that the name given in the configuration file must have been an alias. The monitor sends the canonical name to the DNS server to request the host's IP address.

   If the DNS server locates the IP address of the host using its canonical name, it returns it to the monitor. If the DNS server can't locate the IP address of the host, it returns a message containing details of the failed search to the monitor.
3. If the first two attempts to query the DNS server fail, the monitor sends the IP address of the DNS Server (`192.168.n.n`) to the DNS server and requests its fully qualified host name.

   If the DNS server can locate its own fully qualified host name, it returns it to the monitor. If the DNS server can't locate its own fully qualified host name, it returns a message containing details of the failed search. The request for the server's fully qualified host name (a reverse DNS lookup request), isn't supported on all types of DNS servers. If the target DNS server doesn't support reverse lookups, you can prevent the DNS monitor from sending this request by setting the `LookupServerName`property to `0`.

Recursive lookup

Non-recursive lookups present a more accurate picture of how the DNS server is performing, whereas recursive lookups give a better indication of the DNS performance that Internet applications (and therefore users) are getting. The DNS monitor supports both recursive and non-recursive lookups.

This is typically how Internet applications making DNS queries work. For example, a web browser always specifies recursive lookups when it's attempting to resolve the host portion of a URL.

If a DNS server can't answer a query because it doesn't contain an entry for the host in its database, it can recursively query DNS servers higher up in the hierarchy.

DNS query types

The DNS monitor supports a range of DNS query types. Use the query code when specifying the type of DNS query.

Table 43. DNS query types

Query Code	Query Type
`A`	Host Address
`NS`	Authoritative name server
`MD`	Mail destination
`MF`	Mail forwarder
`CNAME`	Canonical name for an alias
`SOA`	Start of a zone of authority

Table 43. DNS query types (continued)

Query Code	**Query Type**
MB	Mailbox domain name
MG	Mail group member
MR	Mail rename domain name.
NULL	Null RR
WKS	Well known service description
PTR	Domain name pointer
HINFO	Host information
MINFO	Mailbox or mail list information
MX	Mail exchange
TXT	Text strings
AXFR	Transfer of an entire zone
MAILB	Mailbox-related records
MAILA	Mail agent RR
ANY	All records

Configuring DNS Monitor Service tests

Use the DNS monitor configuration parameters to define DNS service tests.

Table 44. Table 3. DNS monitor configuration

Field	**Description**
server	The IP address of the primary DNS server. Example is 192.168.n.n
host	The host name of the target host. Example is www.myconpany.com
description	A text field for providing descriptive information on the element. Example is DNS monitor.
recursivelookups	Enables or disables recursive lookups. • `recurse` (use `true` in ismbatch). • `norecurse` (use `false` in ismbatch). Default: `recurse`.
port	Port on the DNS server to which the monitor listens and the default value is 53.
localip	Specifies the IP address of the network interface on the host machine to which the monitor binds when it performs the test. If the monitor's `IpAddress` property is set, it overrides the value of this field..
querytype	The DNS query type used in the test. For a list of supported query types see Table 43 on page 316.
timeout	The time, in seconds, to wait for the server to respond. Default: `10`.

Table 44. Table 3. DNS monitor configuration (continued)

Field	Description
retries	The number of times the monitor should retry to contact the DNS server before quitting.
poll	The time, in seconds, between each poll. Default: `300`.
failureretests	The number of times to retest before indicating a failure. Default: `0`.
retestinterval	The time, in seconds, to wait between each failure retest. Default: `10`.

Monitoring elemets

In addition to the test results common to all elements, the DNS monitor generates a set of test results containing data specific to DNS service tests.

Table 45. Table 4.DNS monitor elements

Element	Description
`$authoritative`	This element is only created if the information retrieved came from an authoritative DNS server. If the DNS server wasn't authoritative, this element isn't created.
`$domainEmailAddr`	The contact address of the target domain.
`$domainNameServer`	The name of the DNS server for the target domain.
`$fromAliasTime`	The time between issuing a request for a conical name, received from a previous query, and receiving an IP address.
`$localIP`	The local IP address the monitor is configured to use. This may be blank on a machine with only one interface.
`$lookup*(HostLookup)`	The host name or IP address of the target host that the monitor is trying to locate.
`$lookupCName`	The official host name of the target host. This element is only created if the official host name is different from the host name in `$lookupName`.
`$lookupIP*(HostIp)`	The IP address of the target host.
`$lookupName*(Host)`	The full host name of the target host.
`$mxRecords`	The number of MX records found.
`$port`	The port on which the service is monitored.
`$queryType`	The type of DNS query used in the test. For a list of supported query types. See Table 43 on page 316.
`$responseTime*(ResponseTime)`	The time between the monitor issuing a request to the DNS server and receiving a reply from it.
`$retries`	The maximum number of retries, as specified in the profile element.
`$serverIP`	The IP address of the DNS server.

Table 45. Table 4.DNS monitor elements (continued)

Element	Description
`$serverName`	The host name of the DNS server.
`$serverTime`	Time for the server to resolve its own name.

MX record handling

Two elements are created for each MX record found by the DNS monitor: `$mxHostn` and `$mxPreferencen`.

`$mxHostn` stores the host name of an MX record. `$mxPreferencen` contains the preference weighting of the host. The n increments for each record pair to differentiate them. The monitor stores the total number of MX records for a particular host in the element `$mxRecords`. The record pairs are sorted in descending order of MX preference.

Status message

The DNS monitor provides status messages in the `ResultMessage` attribute when using IBM Application Performance Management . These messages indicate the result of the test.

Table 46. Table 5.DNS monitor status messages

Message	Description
`Domain information received`	Request for a domain name succeeded.
`Success`	The request succeeded.
`Invalid Response`	Unrecognized response from the DNS server.
`Connection failed`	The server name specified is invalid.
`No Response from server`	Request timed out.
`Failed to send DNS request`	There's a problem with the network.
`No such domain (no recursion)`	The domain name is incorrect.

Properties

The properties specific to the DNS monitor are described in the following table.

Table 47. DNS monitor properties

Property name	Property parameter	Description
`AcceptCNAME`	`0|1`	If enabled, the DNS monitor accepts the canonical name in the DNS response and doesn't perform any further lookups.
`DNSQueryType`	string	The DNS query type used in tests. See Table 43 on page 316 for a list of supported query types. Default: ANY.

Table 47. DNS monitor properties (continued)

Property name	Property parameter	Description
`LookupServerName`	`0`|`1`	Enables reverse DNS lookup on the DNS server IP address. `0` - disabled `1` - enabled

FTP Monitor

The FTP monitor tests FTP services by either uploading files to, or downloading files from FTP servers. It monitors the performance of the service by recording the response time and data transfer rate, and monitors disk space and file integrity.

Table 48. FTP monitor summary

Monitor files	Name or location
Monitor executable	`nco_m_ftp`
Properties file	`$ISHOME/etc/props/ftp.props`
Rules file	`$ISHOME/etc/rules/ftp.rules`
Log file	`$ISHOME/log/ftp.log`

Guidelines for configuring the FTP monitor

The FTP monitor tests the availability of an FTP server by uploading a file to the server using an FTP STOR command, or downloading a file from the server using an FTP RETR command.

Configuring FTP Monitor Service tests

The configuration parameters for the FTP monitor are described in the following table.

Table 49. FTP monitor configuration

Field	Description
server	The IP address of the target FTP server, or the machine you want to FTP from. Example is `ftp.mycompany.com`
localfile	For FTP GET operations, this field specifies the name and path to which the file is downloaded. For FTP PUT operations, this field specifies the name and path of the file that is uploaded to the FTP server. The default value is FULL PATHNAME. Example is `$ISMHOME/etc/ism/downloads/ftp-test.tar.Z`
remotefile	For FTP GET operations, this field specifies the name and path of the file that is downloaded from the server. For FTP PUT operations, this field specifies the name and path to which the file is uploaded on the FTP server. The default value is FULL PATHNAME. Example is `/sales/prodlist.tar.Z`
description	A text field for providing descriptive information about the element.

Table 49. FTP monitor configuration (continued)

Field	Description
port	The default port that the FTP server uses. Default: 21
username	The username used to log on to the target FTP server.
password	The password used to log on to the target FTP server. Leave this blank if the FTP account doesn't require a password.
command	The FTP command for the monitor to use: • GET or RECV - Download a file from the target FTP server • SEND or PUT - Upload a file to the target FTP server Default: GET.
conntype	Specifies the type of connection for the monitor to establish with the server when it tries to transfer the file: • Active • Passive Default: Active.
timeout	The time, in seconds, to wait for the server to respond. Default: 30.
poll	The time, in seconds, between each poll. Default: 300
failureretests	The number of times to retest before indicating a failure. Default: 0.
retestinterval	The time period in seconds to wait between each failure retest. Default: 10.

Regular expression matching

You can perform a regular expression search on the information being downloaded by entering up to 50 different regular expressions. The FTP monitor attempts to match the contents retrieved to each of the regular expressions.

If a match for a specified regular expression is found, the matched lines (or as much as can fit in the monitor's internal buffer) are returned in the corresponding `$regexpMatchn` element. If the regular expression matches more than once in the information downloaded, only the first match is returned. The status of each regular expression test is indicated by the `$regexpStatusn` elements. You can use the regular expression matches and their status information as criteria for service level classifications.

Regular expressions perform string matching on content downloaded during service tests. These expressions may contain one or more regular expression operators, which determine what content is matched by the expression.

Note: Regular expression syntax can be used to match strings on single lines only. Internet Service Monitoring can't match strings that include new lines or carriage returns. Use multiple regular expressions to match strings that cover multiple lines. You can also use SLC rules to raise alarms based on the result of multiple regular expressions.

Table 50. Regular expression operators

Character	Description
`.`	Matches any single character. For example, the regular expression `r.t` matches the strings `rat`, `rut`, `r t`, but not `root`.
`$`	Matches the end of a line. For example, the regular expression dog$ matches the end of the string `it's a dog` but not the string `There are a lot of dogs`.
`^`	Matches the beginning of a line. For example, the regular expression `^When in` matches the beginning of the string `When in the course of human events` but wouldn’t match `What and When in the`.
`*`	Matches zero or more occurrences of the character immediately preceding. For example, the regular expression `.*` matches any number of any characters.
`\`	Treats the subsequent character as an ordinary character. For example, `\$` matches the dollar sign character ($) rather than the end of a line. Similarly, the expression `\.` matches the period character rather than any single character.
`[]`	Matches any one of the characters between the brackets. For example, the regular expression `r[aou]t` matches `rat`, `rot`, and `rut`, but not `rit`. Specify ranges of characters by using a hyphen. For example, the regular expression `[0-9]` matches any digit. You can also specify multiple ranges. For example, the regular expression `[A-Za-z]` matches any letter case.
`|`	Matches phrases containing either of the conditions specified. For example, `him|her` matches the line `it belongs to him` and the line `it belongs to her`, but doesn’t match the line `it belongs to them`.

Note: If you prefer the output data strings with `curly braces{}` or `double quotes ""` then you need to add an escape character `backslash \` before every `curly brace` and `double quote` in the regular expression.

For example, if the data string is `{"templates":true,"mongodb":true,"ldap":true,"ucd":true,"github":true}`then the regular expression should appear as `\{\"templates\":true,\"mongodb\":true,\"ldap\":true,\"ucd\":true,\"github\":true\}`

Monitor elements

In addition to the test results common to all elements, the FTP monitor generates a set of test results containing data specific to FTP service tests.

Table 51. FTP monitor elements

Element	Description
`$bytesPerSec*(BytesPerSec)`	The average number of bytes transferred each second.
`$bytesTransfered*` `(BytesTransferred)`	The number of bytes uploaded or downloaded.
`$checksum`	The Checksum element doesn't normally provide meaningful values for service level classifications because checksum values aren't known when the profile element is created (the monitor calculates checksum values while tests are in progress). The $checksum and $previousChecksum monitor elements are intended for alert enrichment using the monitor's rules file.
`$command*(FtpCommand)`	The FTP command issued by the monitor.
`$connectionType*(FtpConnection)`	The type of data connection used. This can be `ACTIVE` or `PASSIVE`.
`$connectTime*(ConnectTime)`	The time taken to connect to the FTP server.
`$downloadTime`	The time taken to download the file.
`$localFile*(FtpLocalFile)`	Full pathname of the file stored on the local host. This element is taken from the configuration file.
`$lookupTime*(LookupTime)`	The time taken to look up the FTP server IP address.
`$previousChecksum`	The PreviousChecksum element doesn't normally provide meaningful values for service level classifications because checksum values are not known when the profile element is created (the monitor calculates checksum values while tests are in progress). The $previousChecksum and $checksum monitor elements are intended for alert enrichment using the monitor's rules file.
`$regexp`*n*	The regular expression.
`$regexpMatch`*n*	The contents of the line matching the regular expression.
`$regexpStatus`*n*	The status of the regular expression match: `NONE` - No regular expression checking is configured `MATCHED` - A match was found for the regular expression `FAILED` - A match wasn't found for the regular expression
`$remoteFile*(FtpRemoteFile)`	Full pathname of the file stored on the remote host (the FTP server). This element is taken from the configuration file.
`$responseTime*(ResponseTime)`	The time taken after a connection is created until the first byte of the target file is received.
`$status`	The status code returned by the FTP server.
`$transferTime*(TransferTime)`	Sets the value to `$uploadTime` or `$downloadTime`.
`$uploadTime`	The time taken to upload the file.

Table 51. FTP monitor elements (continued)

Element	Description
`$username`	User name (account name) used by the monitor to log in to the target host. This element is taken from the configuration file if `$message` contains `OK`.

Status messages

The FTP monitor provides status messages in the `ResultMessage` attribute when using IBM Application Performance Management. These messages indicate the result of the test.

Table 52. FTP monitor status messages

Message	Description
`OK`	The FTP request succeeded.
`Unable to open local file for reading/writing`	See the FTP monitor log file for more information.
`Unable to read from/write to local file`	
`Unable to read from data connection`	A data connection to the FTP server was established, but a problem occurred.
`Unable to upload to ftp server`	
`Timed out waiting to read/write`	
`Connection closed by foreign host`	The connection to the FTP server was broken.
`Connection closed unexpectedly`	
`Connection failed`	The monitor failed to connect to the FTP server. See the FTP monitor log file for more information.

HTTP Monitor

The HTTP monitor checks the availability and response time of web servers.

It can monitor individual web pages, including that uses CGI, which would normally require the user to enter data into fields. It can also monitor the download time for elements such as images on a web page.

Table 53. HTTP monitor file summary

Monitor files	Name or location
Monitor executable	`nco_m_http`
Properties file	`$ISHOME/etc/props/http.props`
Rules file	`$ISHOME/etc/rules/http.rules`
Log file	`$ISHOME/log/http.log`

Guidelines for configuring HTTP monitor

The HTTP and HTTPS monitors check the availability and response time of web servers. Use the HTTP monitor in the following situations:

- The target website is static.

  For dynamic websites, use the TRANSX monitor.
- The target website is served over the HTTP protocol.

  For websites that deliver content over the HTTPS protocol, select the HTTPS monitor.

- To perform monitoring across multiple platforms.
- Where speed is a determining factor (the HTTP monitor provides high performance).

HTTP request types

The HTTP monitor emulates a web browser that supports the HTTP/1.0 protocol. To test the web server, the monitor sends it a request for a web page using any of the following HTTP request types:

- `HEAD`

  The `HEAD` command attempts to access a web page and return the HTTP header. Issuing the `HEAD` command is a fast way to check that a web page is accessible.

- `GET`

  The `GET` command attempts to access the web page and return the whole page, including the HTTP header. It doesn't attempt to return files associated with the page, such as images.

- `GETALL`

  The `GETALL` command attempts to access the web page and return the whole page including the HTTP header, background, images, applets, frames, cascading stylesheet (CSS) files, and scripts. Like the `HEAD` and `GET` commands, this command also checks that a web page is accessible, but because the `GETALL` command returns the whole page and all its associated files, it may give a more realistic indication of the time taken to access the page. The monitor also uses multiple threads during a `GETALL` command to more accurately match the behavior of web browsers.

- `POST`

  The `POST` command attempts to access a web page that contains an HTTP form and complete the fields of that form. Add body text for the POST request to the **Body** tab in the Internet Service Monitoring agent configuration, or use the @Body group in the Internet Service Monitoring agent configuration or ismbatch. Alternatively, you can use the FORM parameters. You can't use both Body text and FORM parameters in the POST request.

Using a proxy server

You can test the availability of web pages through a proxy server. When you configure the monitor to use a proxy, it sends HTTP requests through the proxy. If required, you can bypass the proxy cache. You configure the parameters for the proxy server on the **Proxy Details** tab. The HTTP monitor supports authenticated access to proxy servers. This authentication is independent of any authentication required by the target web page.

Proxy server elements

In earlier versions, when you configured a profile element to use a proxy server, by default the HTTP monitor inserted the name of the proxy server and port into the `$server` and `$port` elements, instead of the name and port of the intended destination server. To preserve the value of the intended destination server name and port in earlier versions, set the `generateProxyTokens` property to `1`, or start the monitor with the `-generateproxytokens` command line parameter.

In addition to preserving the values of the `$server` and `$port` elements when this property or command line parameter is set, the monitor generates the `$proxyServer`, `$proxyPort`, `$proxyAuthType`, `$proxyUsername`, and `$proxyCache` elements.

Authentication

If the web page that you want to monitor, or the proxy server you want to test, requires authentication specify credentials for accessing the page in the `authenticationtype`, `username`, and `password`parameter fields on the Advanced or Proxy Details tab.

To disable authentication, set `authenticationtype` to `NONE`.

To select basic authentication:

1. Set the `authenticationtype` to `BASIC`.

2. Set username and password to those required by the web page or proxy server.

To select NTLM:

1. Set authenticationtype to NTLMv1 or NTLMv2.
2. Set username and password to those required by the web page or proxy server.

Note:

The monitor limits the length of HTTP requests to 4096 characters. If the length of the additional form data results in a request length that exceeds this limit, the monitor doesn't include the additional form data in the request.

Configuring HTTP Monitor Service Test

Use the HTTP monitor configuration parameters to define HTTP service tests.

Table 54. HTTP monitor configuration

Field	Description
server	The host name of the server to be monitored. Example is `www.mycompany.com`
page	The URL of the page to be monitored. Example is `index.html`
description	A text field for providing descriptive information on the element. Example is`monitoring via a proxy server`
port	The port on the HTTP server to use. Default: `80`
localip	Specifies the IP address of the network interface that the monitor uses for the test. If this field is empty, the monitor uses the interface specified by the `IpAddress` property.
version	The HTTP protocol version to be used: • `1.0` • `1.1` Default: `1.0`
command	The HTTP request type: • `HEAD` • `GET` • `GETALL` • `POST` Default: `GET`
formname	When used in a transaction, the HTTP monitor scans the specified form for default values. Any values found are automatically completed the next HTTP step in the transaction.

Table 54. HTTP monitor configuration (continued)

Field	Description
authenticationtype	Specifies the challenge -response authentication mechanism for authenticating network users: • NONE - No authentication • BASIC • NTLMv1 - Windows NTLM version 1 challenge/response authentication • NTLMv2 - Windows NTLM version 2 Default: NONE
username	The username (account name) for the monitor to use to log in to the server.
password	The password corresponding to the username for the monitor to use to log into the server.
timeout	The time, in seconds, to wait for the server to respond. Default: 30
poll	The time, in seconds, between each poll. Default: 300 .
failureretests	The number of times to retest before indicating a failure. Default: 0
retestinterval	The time, in seconds, to wait between each failure retest. Default: 10
verifycertificate	It's by default disabled.
Proxy details	
server	The host name of the proxy server.
port	The port on the proxy server to use. Default: 8080
authenticationtype	The server authentication type for the proxy server. For more information, see previous `authenticationtype`. Default: NONE
username	Used by the monitor together with the password to log into the proxy server.
password	Used by the monitor together with the username to log into the proxy server and the label is password.
useproxy	Configures the monitor to perform the request using a proxy server. • `proxy` (use `true` in ismbatch) • `noproxy` (use `false` in ismbatch) Default: `noproxy`

Table 54. HTTP monitor configuration (continued)

Field	Description
hostnamelookuppreference	Determines which IP version, IPv6 or IPv4, is applied to the supplied host name. The options are: • `default` sets the monitor to use monitor-wide properties settings. This is the default. • `4Then6` selects IPv4 and then IPv6. Uses IPv4 addresses if they are available. If no IPv4 addresses are found, IPv6 addresses are used. • `6Then4` selects IPv6 and then IPv4. Uses IPv6 addresses if they are available. If no IPv6 addresses are found, IPv4 addresses are used. • `4Only` selects IPv4 only. Uses IPv4 addresses only. If there are no IPv4 addresses, the poll returns an error. • `6Only` selects IPv6 only. Uses IPv6 addresses only. If there are no IPv6 addresses, the poll returns an error. • `6Or4` selects either IPv4 or IPv6. Uses the first address returned from the host name.
nocache	It's by default set to cache.

Regular Expression

You can perform a regular expression search on the information being downloaded by entering up to 50 different regular expressions. The HTTP monitor attempts to match the contents retrieved to each of the regular expressions. If a match for a specified regular expression is found, the matched lines (or as much as can fit in the monitor's internal buffer) are returned in the corresponding `$regexpMatchn` element. If the regular expression matches more than once in the information downloaded, only the first match is returned. The status of each regular expression test is indicated by the `$regexpStatusn` elements. You can use the regular expression matches and their status information as criteria for service level classifications. SeeTable 50 on page 322 for information about regular expression syntax,

Head and Form parameter

The HTTP monitor can send extra data in the header fields and message body of HTTP requests.

You configure the parameters for this extra data on the Parameters tab. The parameters are Name, Value and Type and they operate in the following way:

- Name-value pairs of type HEAD specify additional header fields, such as `User-Agent` and `Referer`, included in all HTTP requests sent by the monitor. Header fields may be specified for any type of HTTP method (`GET`, `GETALL`, `HEAD` or `POST`).

  For ITCAM for Transactions V7.4.0.1 and later, the default user agent head parameter, `Mozilla/5.0 (ISM-MONITOR)` is added for every new HTTP or HTTPS element. The default user-agent header is so that the HTTP and HTTPS monitors can be used for websites that switch content based on the browser client.
- Name-value pairs of type FORM specify extra data included in the message body of HTTP `POST`requests sent by the monitor. If the target page contains a form matching the name specified in the formname field, the monitor treats any name-value pairs in the form as if they were configured in the profile element.

Note:

The monitor limits the length of HTTP requests to 4096 characters. If the length of the additional form data results in a request length that exceeds this limit, the monitor doesn't include the additional form data in the request.

Monitor elements

In addition to the test results common to all elements, the HTTP monitor generates a set of test results containing data specific to HTTP service tests. Elements indicated by an asterisk (*) are available as attribute. The names of the attributes are shown within brackets. Absence of an asterisk indicates there's no equivalent attribute. Attributes shown in bracket but without an element indicates that they are only available as attribute, there's no equivalent element.

Table 55. HTTP monitor elements

Element	Description
`$bytesPerSec*(BytesPerSec)`	The average number of bytes transferred each second.
`$bytesTransfered*(BytesTransferred)`	The number of bytes uploaded or downloaded.
`$checksum`	The Checksum element doesn't normally provide meaningful values for service level classifications because checksum values aren't known when the profile element is created (the monitor calculates checksum values while tests are in progress). The $checksum and $previousChecksum monitor elements are intended for alert enrichment using the monitor's rules file.
`$command`	The HTTP command issued by the monitor. For example, `HEAD`, `GET`, `GETALL`, or `POST`.
`$connectTime*(ConnectTime)`	The time taken to connect to the server.
`$downloadTime*(DownloadTime)`	The time taken to download the file.
`(Elements)`	The number of page elements received.
`$formname`	The name of the form used in a `POST` action.
`$lastStatus*(PageStatus)`	If a profile element retrieves multiple pages, this element contains the result string of the last page retrieved. This value is the same as that of `$urlResult`*n* where *n* is equal to the value of `$pageCount`.
`$lastModified`	The value of the `Last-Modified` HTTP header field of the first page retrieved.
`$page*(Page)`	The page accessed on the HTTP server.
`$pageCount`	The total number of resources downloaded during a `GETALL` test, excluding the test page itself. If the tested page doesn't refer to any other resources, this element isn't generated.
`$port*(Port)`	The port used to access the HTTP server. If the test used a proxy server, this is the value of the port on the proxy server to which the request was submitted. To preserve the port of the intended destination server, set the `generateProxyTokens` property to `1`, or start the monitor with the `-generateproxytokens` command line parameter

Table 55. HTTP monitor elements (continued)

Element	Description
$previousChecksum	The PreviousChecksum element doesn’t normally provide meaningful values for service level classifications because checksaren’tes aren’t known when the profile element is created (the monitor calculates checksum values while tests are in progress). The $previousChecksum and $checksum monitor elements are intended for alert enrichment using the monitor's rules file.
$proxyAuthType	The server authentication type for the proxy server.
$proxyCache	The value true indicates that the proxy server retrieved the web page from the server, rather than from its own cache.
$proxyPort	The port number of the proxy server to which the request was submitted.
$proxyServer	The host name of the proxy server.
$proxyUsername	Used by the monitor together with the password to log in to the proxy server.
$regexpMatch*n*	The contents of the line matching the regular expression.
$regexp*n*	The regular expression.
$regexpMatch*n*	The contents of the line matching the regular expression.
$regexpStatus*n*	The status of the regular expression match: NONE - No regular expression checking is configured MATCHED - A match was found for the regular expression FAILED - A match wasn’t found for the regular expression
$responsetime* (ResponseTime)	The time taken after a connection is created, until the first byte of the page is received.
$timeSinceModification	The time that has elapsed since the page was last modified. This is the difference between the time of the test and the value of the Last-Modified HTTP header field of the first page retrieved.
$urlDownloadTimes*n** (UrlDownloadTime)	URL download time of each element in a GETALL request. Each element is numbered, starting with 000 ($urlDownloadTime000, $urlDownloadTime001, $urlDownloadTime002, and so on).
$url*n**(Url)	URL of each page in a GETALL test. Each page is numbered, starting with 000 ($url000, $url001, $url002, and so on).
$urlResult*n** (UrlResultString)	Result string for each page downloaded in a GETALL request. Each result is numbered, starting with 000 ($urlResult000, $urlResult001, $urlResult002, and so on).
$username	The name used to access pages that require the user to be authenticated.

Status message

The HTTP monitor provides status messages in the ResultMessage attribute when using IBM Application Performance Management. These messages indicate the result of the test.

Table 56. Table 4. HTTP monitor status messages

Message	Description
`OK`	The request from the monitor succeeded.
`Form not found`	The requested page cannot be found.
`Initialise Fetch Page Failed`	Not enough memory to allocate space for the HTTP page fetching mechanism. The previous line message should contain further information.
`Connection Failed`	The monitor failed to connect for reasons other than the link is down, connection is reset, link is unreachable, connection timed out, connection terminated, or host is down. See the HTTP monitor log file for more information.

Form parameters and regular expression matching

Monitor the operation of the form `http://support.mycompany.com/cgi-bin/search.cgi` by sending HTTP POST requests with the form parameter `search=ism`, and use a regular expression to match the string `Your search was successful` in the response. If this string is returned in the response, classify the service level as `Good`, and `Failed` otherwise.

Create a new HTTP profile element and set the fields as shown in the following table.

Table 57. HTTP form profile element example

Profile element configuration field	Value
server	`support.mycompany.com`
page	`/cgi-bin/search.cgi`
description	`Example - form parameters and regular expressions`
Regular expression details	
match 1	`Your search was successful`
Service level classification details	
statement	`Regexp Status 1 = MATCHED then status GOOD`
Head and Form details	
name	`search`
value	`ism`
type	`FORM`

Properties

The properties and command-line options specific to the HTTP monitor are described in the following table.

Table 58. HTTP monitor properties

Property name	Property parameter	Description
AllowDuplicateDownload	0|1	Forces pages to be downloaded each time they are found. 0 - disabled (downloaded only once) 1 - enabled
ForceHTMLParse	0|1	Forces pages that don't have content-type text/html to be parsed as HTML. 0 - disabled 1 - enabled
GenerateProxyTokens	0|1	Specifies whether the monitor generates additional elements containing information about the proxy server if a proxy server is used in a test. 0 - disabled 1 - enabled (additional elements $server and $port contain values for the proxy server)
GETALLThreadNum	1|2|3|4|5	Specifies the number of separate threads to use during a GETALL request.
GetLinkTags	0|1	Activates download of linked stylesheets for GETALL requests: 0 - disabled 1 - enabled (if the target page contains a link tag with attribute value rel=stylesheet, the monitor attempts to download the resource referred to by the link tag's href attribute)
HostnameLookupPreference	string	Determines which IP version, IPv6 or IPv4, is applied to the supplied host name. The possible values are: • 4Then6 selects IPv4 and then IPv6. Uses IPv4 addresses if they are available. If no IPv4 addresses are found, IPv6 addresses are used. • 6Then4 selects IPv6 and then IPv4. Uses IPv6 addresses if they are available. If no IPv6 addresses are found, IPv4 addresses are used. • 4Only selects IPv4 only. Uses IPv4 addresses only. If there are no IPv4 addresses, the poll returns an error. • 6Only selects IPv6 only. Uses IPv6 addresses only. If there are no IPv6 addresses, the poll returns an error. • 6Or4 selects either IPv4 or IPv6. Uses the first address returned from the host name. Default: 4Then6

Table 58. HTTP monitor properties (continued)

Property name	Property parameter	Description
Ipv6Address	integer	The local address to bind to as an origin for HTTP requests when using HTTP IPv6. Default: no address
NoParseExtensions	string	A comma-separated list of file extensions indicating file types that the monitor won't parse and instead only download.
OutputDirectory	string	Specifies the output directory to use if OutputResult is true (set to 1). Default: $ISHOME/var
OutputResult	0|1	Specifies whether the monitor saves the data it receives from the service. 0 - disabled 1 - enabled
RelativeRedirects	0|1	Allows Location fields in HTTP 301 and HTTP 302 status codes to contain relative URLs instead of absolute URLs. 0 - absolute URLs 1 - relative URLs
RFCPOST	0|1	Specifies that the monitor should follow RFC1945 and RFC2616 and send a second POST after a redirect. Many web servers don't expect a POST after a redirect and most browsers don't follow the RFCs. 0 - disabled 1 - enabled

HTTPS Monitor

The HTTPS monitor checks the availability and response time of web servers. It can monitor individual web pages, including that uses HTML forms, which normally require the user to enter data into fields.

Note: The HTTPS monitor works in the same way as the HTTP monitor, but it communicates with the HTTP server by using version 2 or version 3 of the SSL (Secure Sockets Layer) protocol, which encrypts all communications between the server and the monitor.

Table 59. HTTPS monitor file summary

Monitor files.	Name or location
Monitor executable	nco_m_https
Properties file	$ISHOME/etc/props/https.props
Rules file	$ISHOME/etc/rules/https.rules
Log file	$ISHOME/log/https.log

Guidelines for configuring HTTPS monitor

The HTTPS monitor checks the availability and response time of web servers. Use the HTTPS monitor in the following situations:

- The target website is static.

  For dynamic websites, use the TRANSX monitor.
- The target website is served over the HTTPS protocol.

  For websites that deliver content over the HTTP protocol, select the HTTP monitor.
- To perform monitoring across multiple platforms.
- Where speed is a determining factor (the HTTPS monitor provides high performance).

Client-side certificate

The monitor enables you to monitor servers that require client-side certificates for mutual authentication.

Specify the SSL certificate file, key file, and key password when creating a profile element.

Certificates must be in Privacy Enhanced Mail (PEM) format. If your certificate is in another format, you must convert it to PEM format. Certificates can be converted by using software such as openSSL, which is available from http://www.openssl.org.

Note: If you always use the same certificate, key, and password in all profile elements, specify them using monitor properties instead of defining them in every profile element you create.

Configuring HTTPS monitor service tests

Use the HTTPS monitor configuration parameters to define HTTPS service tests.

Table 60. HTTPS monitor configuration

Field	Description
server	The host name of the server to be monitored. Example is `www.myconpany.com`
page	The URL of the page to be monitored. Example is `/secure/`
description	A text field for providing descriptive information on the element.
port	The port on the server to use. Default: 443
localip	Specifies the IP address of the network interface that the monitor uses for the test. If this field is empty, the monitor uses the interface that is specified by the `IpAddress` property.
version	The HTTPS protocol version to be used: • `1.0` • `1.1` Default: `1.0`

Table 60. HTTPS monitor configuration (continued)

Field	Description
command	The request type: • `HEAD` • `GET` • `GETALL` • `POST` Default: `GET`
formname	When used in a transaction, the HTTPS monitor scans the specified form for default values. Any values that are found are automatically completed the next HTTPS step in the transaction.
authenticationtype	Specifies the challenge-response authentication mechanism for authenticating network users: • `NONE` - No authentication. • `BASIC` • `NTLMv1` - Windows NTLM version 1 challenge/response authentication. • `NTLMv2` - Windows NTLM version 2. Default: `NONE`
username	The username (account name) for the monitor to use to log in to the HTTPS server.
password	TThe password corresponding to the username for the monitor to use to log in to the HTTPS server.
sslcertificatefile	The path and filename of the digital certificate file that is used in the monitor element. If the path isn’t absolute, the monitor interprets it relative to the working directory (`$ISMHOME/platform/arch/bin`). If you don’t specify a certificate file, the monitor uses the certificate that is specified by the monitor property `SSLCertificateFile`.
sslkeyfile	The path and filename of the file containing the SSL private key, which is used to identify the server and sign the SSL messages.
sslkeypassword	The password used to encrypt the SSL private key.
timeout	The time, in seconds, to wait for the server to respond. Default: `30`
poll	The time, in seconds, between each poll. Default: `300`
failureretests	The number of times to retest before indicating a failure. Default: `0`

Table 60. HTTPS monitor configuration (continued)

Field	Description
retestinterval	The time, in seconds, to wait between each failure retest. Default: `10`
Proxy details	
server	The host name of the proxy server.
port	The port on the proxy server to use.
authenticationtype	The server authentication type for the proxy HTTPS server. See authenticationtype for further information.
username	The username for the monitor to use to log in to the proxy HTTPS server.
password	The password for the monitor to use to log in to the proxy HTTPS server.
useproxy	Configures the monitor to perform the request by using a proxy server. • `proxy` (use `true` in ismbatch) • `noproxy` (use `false` in ismbatch) Default is `noproxy`
hostnamelookuppreference	Determines which IP version, IPv6, or IPv4, is applied to the supplied host name. The options are: • `default` sets the monitor to use monitor-wide properties settings. This is the default. • `4Then6` selects IPv4 and then IPv6. Uses IPv4 addresses if they are available. If no IPv4 addresses are found, IPv6 addresses are used. • `6Then4` selects IPv6 and then IPv4. Uses IPv6 addresses if they are available. If no IPv6 addresses are found, IPv4 addresses are used. • `4Only` selects IPv4 only. Uses IPv4 addresses only. If there are no IPv4 addresses, the poll returns an error. • `6Only` selects IPv6 only. Uses IPv6 addresses only. If there are no IPv6 addresses, the poll returns an error. • `6Or4` selects either IPv4 or IPv6. Uses the first address that is returned from the host name.

Regular expression matching

You can perform a regular expression search on the information being downloaded by entering up to 50 different regular expressions. The HTTPS monitor attempts to match the contents that are retrieved to each of the regular expressions.

If a match for a specified regular expression is found, the matched lines (or as much as can fit in the monitor’s internal buffer) are returned in the corresponding `$regexpMatchn` element. If the regular expression matches more than once in the information downloaded, only the first match is returned. The status of each regular expression test is indicated by the `$regexpStatusn` elements. You can use the regular expression matches and their status information as criteria for service level classifications.

For more information, see Table 50 on page 322.

Head and Form parameter

Similar to the HTTP monitor, the HTTPS monitor can send extra data in the header fields and message body of HTTP requests.

For details on head and form parameters, see HTTP Head and Form parameter.

Monitor elements

Table 61. HTTPS SSL monitor elements

Element	Description
$SSLcertificateSerialNumber	The serial number of the X509 certificate presented by the server.
$SSLcipherSuiteCount	The number of cipher suites available on the connection.
$SSLcipherSuiteList	The list of cipher suites available on the connection.
$SSLcipherSuiteName	The cipher suite selected for the connection.
$SSLeffectiveSessionKeyBits	The number of bits in the session key. This is typically 128 or 168, or 40 for export versions.
$SSLHandshakeTime* (SslHandshakeTime)	The time taken to establish the SSL connection.
$SSLissuerName	The issuer name for the server's X509 format certification.
$SSLprotocolVersion	The version of SSL being used, either v2 or v3.
$SSLpublicKeyLengthBits	The size of the server's public key. This is typically 1024 bits, except where an export specification cipher suite is used.
$SSLserverCertificateValidFrom	The date that the server certificate is valid from.
$SSLserverCertificateValidTo	The date that the server certificate is valid to.
$SSLserverName	SSL server name.
$SSLsubjectName	The subject name for the X509 format certification. This is typically the name of the organization controlling the server.

Elements indicated by an asterisk (*) are available as attributes. The names of the attributes are shown within brackets. Absence of an asterisk indicates there’s no equivalent attribute. Attributes that are shown in bracket but without element indicates that they are only available as attributes, there's no equivalent element.

The HTTPS monitor produces the same extra elements as the HTTP monitor, as described in Table 55 on page 329. In addition, it produces the elements that are related to SSL if a client-side certificate is used in the test, as described in Table 61 on page 337.

In addition to the test results common to all elements, the HTTPS monitor generates a set of test results containing data specific to HTTPS service tests.

Status Message

The HTTPS monitor provides status messages in the `ResultMessage` attribute when using IBM Application Performance Management. These messages indicate the result of the test.

In addition to the HTTP status messages, the HTTPS monitor also generates the messages that are listed in Table 62 on page 338.

Table 62. HTTPS monitor status messages

Message	Description
`OK`	The monitor that is successfully connected to the server.
`SSL handshake failed`	The monitor failed to initialize SSL connectivity after establishing a connection to the server.
`Connection failed`	The monitor failed to connect for reasons other than the link is down, connection is reset, link is unreachable, connection timed out, connection terminated, or host is down. See the HTTP monitor log file for more information.

Properties

The HTTPS monitor has the same properties as the HTTP monitor.

For details about the properties options that are the same as the HTTP monitor, see Table 58 on page 332. Table 5 lists some more properties that are specific to HTTPS.

Table 63. HTTPS monitor specific properties

Property name	Property parameter	Description
SSLCertificateFile	string	The path and filename of the digital certificate file that is used if no certificate is explicitly specified for an HTTPS element during its creation. If the path isn't absolute, the monitor interprets it relative to the working directory (`$ISHOME/platform/arch/bin`).
SSLCipherSuite	string	The cipher suite to use for SSL operations. Default: `RC4:3DES:DES:+EXP`
SSLDisableTLS	integer	Disables TLSv1 for heritage support. Default: `0` - TLSv1 is enabled. 1 - TLSv1 is disabled.
SSLKeyFile	string	The file containing the SSL private key.
SSLKeyPassword	string	The password used to encrypt the SSL private key.

Cipher Suits

The `SSLCipherSuite` property specifies the cipher suite that is used by the HTTPS monitor. For more information about SSL settings, see "SSL setting in Internet Service Monitoring" on page 428.

ICMP Monitor

The ICMP monitor tests the performance of the Internet Control Message Protocol service running on a network. To do this, the monitor uses the ICMP echo command.

The following table lists the ICMP monitor files.

Table 64. ICMP monitor files

Monitor files	Name or location
Monitor executable	`nco_m_icmp`
Properties file	`$ISHOME/etc/props/icmp.props`

Table 64. ICMP monitor files (continued)

Monitor files	Name or location
Rules file	`$ISHOME/etc/rules/icmp.rules`
Log file	`$ISHOME/log/icmp.log`

Guidelines for configuring ICMP monitor

The ICMP monitor issues ICMP echo requests (commonly called `pings`) to target hosts and waits for an echo reply response. It records lookup times, round-trip times, and success rate metrics that provide an indication of how well the network is performing. When the monitor issues an echo request, the request may pass through one or more routers before it reaches the target host. These routers can respond to the monitor before the target host has received the echo request. If an echo request issued by the monitor passes through a router, the router may issue a reply to the monitor. This reply might indicate that the router cannot locate the target host, or that the router is too busy to process the request. It's possible that the monitor might receive replies from multiple routers before it receives an echo-reply from the target host. If the monitor successfully receives an echo-reply from the target host, it records the time taken. If the monitor doesn't receive a reply from the target server within the specified timeout period, the request is recorded as failed. You can configure the monitor to send multiple ICMP echo requests to the same target in each test. The monitor records statistics for each of the requests sent.

Note: Run the ICMP monitor as `root` because it opens a raw socket to send ICMP packets.

Configuring ICMP monitor service tests

Use the ICMP monitor configuration parameters to define service tests. When you configure the monitor, default values are shown for the timeout and poll interval parameters. These defaults are 30 and 300 seconds respectively. Other defaults listed in the table aren't shown during configuration but are applied when the configuration details are saved if no value has been specified.

The followinf table lists the ICMP monitor configurations.

Table 65. ICMP monitor configuration

Field	Description
server	The host name or IP address of the server to which the echo requests are sent. Example is `test.myconpany.com`
description	A text field for providing descriptive information on the element.
timeout	The time, in seconds, to wait for the server to respond to each echo request. Default: `10`
numberofpings	The number of echo requests to send. Default: 5
packetinterval	The time, in seconds, to wait between sending echo requests. Default: `1`
packetsize	The size, in bytes, of each echo request sent. Default: 64
typeofservice	Sets the Type of Service field in the IP layer. Both IPv4-style Type Of Service (TOS) values and DSCP Differentiated Service Field values may be entered. Valid values are `0` -255.

Table 65. ICMP monitor configuration (continued)

Field	Description
retries	The number of times the monitor should retry each echo request before quitting. Default: 0
poll	The time, in seconds, between each poll. Default: 300
failureretests	The number of times to retest before indicating a failure. Default: 0
retestinterval	The time, in seconds, to wait between each failure retest. Default: 10
hostnamelookuppreference	Determines which IP version, IPv6 or IPv4, is applied to the supplied host name. Options are: • default sets the monitor to use monitor-wide properties settings. This is the default. • 4Then6 selects IPv4 and then IPv6. Uses IPv4 addresses if they are available. If no IPv4 addresses are found, IPv6 addresses are used. • 6Then4 selects IPv6 and then IPv4. Uses IPv6 addresses if they are available. If no IPv6 addresses are found, IPv4 addresses are used. • 4Only selects IPv4 only. Uses IPv4 addresses only. If there are no IPv4 addresses, the poll returns an error. • 6Only selects IPv6 only. Uses IPv6 addresses only. If there are no IPv6 addresses, the poll returns an error. • 6Or4 selects either IPv4 or IPv6. Uses the first address returned from the host name.

Note: Monitor the availability of the host test.mycompany.com by checking the response times at 10-minute intervals. Attempt to connect to the server within 30 seconds and, if it times out, retry twice more. If it still fails, repeat the test three times with 5 seconds between each retry.

Monitor elements

In addition to the test results common to all elements, the ICMP monitor generates a set of test results containing data specific to ICMP service tests.

The following table describes the additional elements for the ICMP monitor.

Elements indicated by an asterisk (*) are available as attributes. The names of the attributes are shown within brackets below the element. Absence of an asterisk indicates there's no equivalent attribute. Attributes shown in bracket but without an element indicate that they are only available as attributes, there's no equivalent element.

Table 66. ICMP monitor elements

Element	Description
$averageRTT*(Average RTT)	The average round-trip time in seconds.
$endTime	The UNIX time the response was received.

Table 66. ICMP monitor elements (continued)

Element	Description
`$jitter`	The absolute value of the difference between the arrival times of two adjacent ICMP echo requests, minus their departure times. This value is calculated according to the formula specified in RFC2598. The element is created only if the number of echo requests is greater than one. If more than two echo requests are used, the value is the average jitter between all pairs of echo requests.
`$lookupTime*(LookupTime)`	The time taken to obtain the IP address of the host server.
`$maxRTT*(MaxRTT)`	The maximum round-trip time in seconds.

`$minRTT*`
`(MinRTT)`
	The minimum round-trip time in seconds.

`$numberPackets`	The number of ICMP echo requests sent, as specified in the profile element.
`$packetInterval`	The time between sending each ICMP echo request, as specified in the profile element.
`$packetRetries`	The number of times the monitor tried to resend ICMP echo requests before exiting.
`$packetSize`	The size (in bytes) of each ICMP echo request, as specified in the profile element.
`$pingAttempts Failed`	The number of attempts made for the first unsuccessful ICMP echo request.
`$pingAttempts Responded`	The number of attempts made for the first successful ICMP echo request.
`$pingMessageFailed`	The message returned for the first unsuccessful ICMP echo request.
`$pingMessage Responded`	The message returned for the first successful ICMP echo request.
`$pingReceivedTime Failed`	The UNIX time the first unsuccessful echo response was received.
`$pingReceivedTime Responded`	The UNIX time the first successful echo response was received.
`$pingRespondIP Failed`	The IP address that responded to the first unsuccessful ICMP echo request.
`$pingRespondIP Responded`	The IP address that responded to the first successful ICMP echo request.
`$pingRTTFailed`	The round-trip time for the first unsuccessful ICMP echo request in seconds.
`$pingRTTResponded`	The round-trip time for the first successful ICMP echo request in seconds.
`$pingSentTime Failed`	The UNIX time that the first unsuccessful ICMP echo request was sent.
`$pingSentTime Responded`	The UNIX time that the first successful ICMP echo request was sent.

Table 66. ICMP monitor elements (continued)

Element	Description
`$pingsFailed`	The number of ICMP echo requests sent to which there was no echo response.
`$pingsResponded`	The number of valid echo responses received.
`$pingTime`	The time taken to receive the echo response after sending the ICMP echo request.
`$respondPercent*` (`RespondPercent`)	The percentage of ICMP echo requests sent for which there was a response.
`$responseTime`	The time taken for the target host to respond to an ICMP echo request.
`$sentTime`	The UNIX time that the ICMP echo requests were sent.
`$spreadRTT`	The difference between `$maxRTT` and `$minRTT`.
`$startTime`	The UNIX time the test began.
`$totalHostTime`	The time taken to receive the echo response after beginning the test.
`$typeOfService`	The Type of Service field in the IP layer, as specified when adding a new ICMP element. For details, see "ICMP Monitor" on page 338.

The ICMP monitor creates a separate set of `$pingname` elements to record the results for each ICMP echo request sent during the test. The number of requests sent is indicated by `$numberPackets`. For example, for the `$pingRTT` element, if `$numberPackets` is 3, the monitor creates three elements (`$pingRTT1`, `$pingRTT2`, and `$pingRTT3`), containing the round-trip time measurement for the three ICMP echo requests sent.

Status message

The ICMP monitor provides status messages in the `ResultMessage` attribute when using IBM Application Performance Management. These messages indicate the result of the test.

The following table describes the ICMP status messages.

Table 67. ICMP monitor status messages

Message	Description
`Pings Complete`	The ICMP echo request succeeded.
`ICMP echo failed`	The monitor cannot issue the ICMP echo request because there's a problem with the monitor host or its connection to the network.
`Timed out`	The ICMP echo request timed out.
`Unreachable`	This message is returned from a router and isn't necessarily accurate.
`Source quench`	A router is too busy to process the ICMP echo request.
`Time exceeded`	This message is returned from a router. It indicates that the ICMP echo request has been forwarded around the network too many times.
`Parameter problem`	This message is returned from a router. It indicates that the router can't process the ICMP echo request. This might be because the message has been corrupted.

Properties

The properties specific to the ICMP monitor are described in the following.

Table 68. ICMP properties

Property name	Property parameter	Description
EventsPerSec	not applicable	This property isn't supported.
IntraPingWait	integer	The minimum time interval in milliseconds between all pings sent by the ICMP monitor. Use to tune your system to spread the network traffic over a longer period. For example, in an environment with thousands of targeted ICMP hosts, set IntraPingWait to 3. Default: 0
Ipv6Address	integer	The local address to bind to as an origin for ICMP echo requests when using ICMP IPv6. Default: no address
MaxDNSResolvingThreads	integer	The maximum number of threads to be used by the DNS Resolver. Default: 20
MaxPacketSize	integer	The maximum ICMP packet size in bytes.
PingsPerSec	integer	The number of echo requests the monitor attempts to send per second. The number of actual requests sent dependents on CPU and network load. Default: 100
SocketBufferSize	integer	The size of the receiving socket buffer (in kilobytes). Default: 32

LDAP Monitor

The LDAP monitor tests the operation of Lightweight Directory Access Protocol (LDAP) servers.

The following table lists the LDAP monitor files.

Table 69. LDAP monitor files

Monitor files	Name or location
Monitor executable	nco_m_ldap
Properties file	$ISHOME/etc/props/ldap.props
Rules file	$ISHOME/etc/rules/ldap.rules
Log file	$ISHOME/log/ldap.log

Guidelines for configuring LDAP monitor

The LDAP monitor tests LDAP services by connecting to an LDAP server and attempting to locate a specific entry. If the server succeeds in locating the entry, it returns the contents of the entry to the monitor. The LDAP monitor can use SSL to authenticate and connect to the LDAP server.

To configure the LDAP monitor, it is necessary to understand how the LDAP protocol and the monitored directory service work. LDAP is an Internet Protocol for accessing and managing Directory Services. A directory service is a distributed database application. A directory consists of entries. For example, a directory might contain entries that relate to an organization's employees or resources. Each entry contains a set of attributes, for example the entries in a directory of employees might contain an employee's name, telephone number and address.

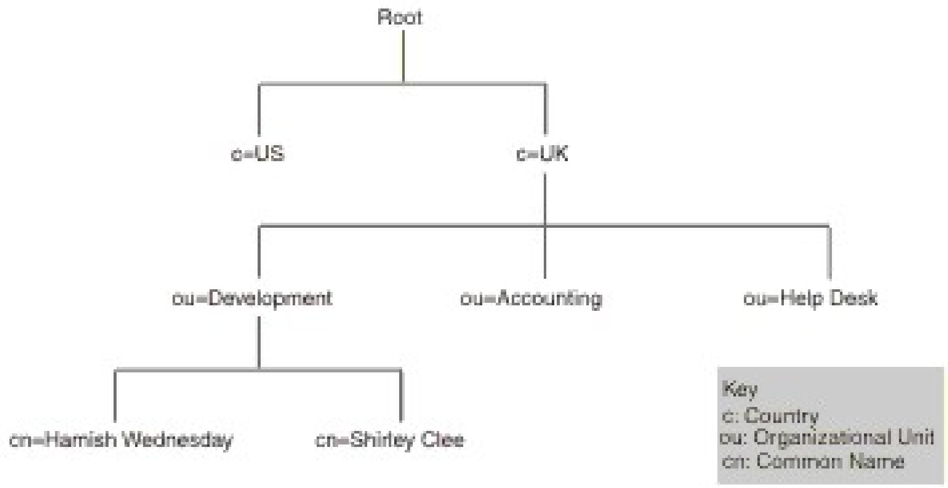
Individual Directory Services can be constructed differently, so the monitoring procedure can also differ.

LDAP versions

The LDAP monitor supports both version 2 and 3 of LDAP. By default, the monitor attempts to connect to the target LDAP server that uses version 3, then automatically falls back to version 2 if the attempt fails. You can force the monitor to always use version 2 by setting the **NOLDAPV3** property.

Example directory service

This example directory service stores the personal details of all employees. The directory is divided into countries and then into departments. Employees and their attributes are stored under each department.



The Directory hierarchy example image shows an extract from an example directory. This figure shows a directory structure. At the apex level is root. The two subdirectories represent countries and are labeled UK and US. The UK subdirectory is further divided into three other subdirectories representing the organization units. They are labeled as Development, Accounting, and Help Desk. Within the Development organization unit there are two subdirectories, for common names which are Shirley Clee and Hamish Wednesday.

Entities are referenced by their distinguished names. A distinguished name is the route to the entity. For example, the distinguished names of the accounting department and Hamish Wednesday would be:

```
dn="ou=accounting, c=UK"
dn="cn=Hamish Wednesday, ou=Development, c=UK"
```

The entry for each employee has multiple attributes. For example, the entry for Hamish Wednesday contains the following details.

```
cn: Hamish Wednesday
uid: ham
mail: HWednesday@development.mycompany.com
telephoneNumber: 88 88 55 44
```

Each entity in the directory hierarchy can be protected by a username (in LDAP it is a distinguished name) and password. The monitor uses this username and password to access the LDAP server.

When the monitor accesses the server, it indicates where in the directory hierarchy the search for the target entity begins. This is specified in the searchBase field as a distinguished name. For example, the search could begin at the department level:

```
ou=Accounting, c=UK
```

Note: The entities that make up a distinguished name are in reverse order. That is, they start at the lowest point in the hierarchy, then list each preceding entity.

The target entity is passed to the server in the filter field. This field contains an attribute of the target entity. For example, to search for Hamish Wednesday's entity, the filter field might contain:

```
(uid=ham)
```

The LDAP server uses the fields that are supplied by the monitor to search for the target entity. The result of the search is returned to the monitor.

If the search is successful, the server also returns the attributes of the target entity. The monitor converts it into elements whose names are created dynamically. For example, the monitor would convert the entry for Hamish Wednesday into:

```
$dnMatched = "cn=Hamish Wednesday, ou=Development, c=UK"
$cn = "Hamish Wednesday"
$uid = "ham"
$mail = "HWednesday@development.mycompany.com"
$telephoneNumber = "88 88 55 44"
```

LDAP Authentication

SSL LDAP Server Authentication relies on public-private key certificates, signed by certificate authorities, such as Verisign and Thawte. For SSL authentication, the LDAP monitor uses the Netscape cert7db database of public certificates to verify LDAP server certificate signatures issued by certificate authorities.

If you are using certificates that are signed by a certificate authority that is recognized by Netscape, such as Verisign or Thawte, the LDAP monitor recognizes them automatically. If you are using certificates that are signed by your organization or by an organization not in the Netscape database, you must add them to the cert7db database.

Use the certutil utility, available from Netscape to add your certificates to the database. The cert7db database for the LDAP monitor is in the file `$ISHOME/certificates/cert7.db`.

To monitor LDAP servers that are secured by SSL or TLS encryption, set the environment variables as described in the following table:

Table 70. Environment variables required to monitor secure LDAP servers

Variable	Description	Setting
`LDAPTLS_CACERT`	Specifies the file that contains the CA certificates	File containing server certificate. For example, `cacert.pem`.
`LDAPTLS_REQCERT`	Specifies the checks to perform on a server certificate	Select from `never`|`allow`|`try`|`demand`.

For more information, see http://www.openldap.org.

Properties

Properties that are specific to the LDAP monitor are described in the following table:

Table 71. LDAP monitor properties options

Property name	Property parameter	Description
NOLDAPV3	0|1	Force the monitor to use LDAP v2 instead of LDAP v3. 0 - use LDAP v3 1 - use LDAP v2

Cipher suites

The `SSLCipherSuite` property specifies the cipher suite that is used by the LDAP monitor. For more information about SSL settings, see "SSL setting in Internet Service Monitoring" on page 428.

Configuring LDAP monitor service tests

Use the LDAP monitor configuration parameters to define service tests.

When you configure the monitor, default values are shown for the timeout parameter is 30 seconds and poll interval parameter is 300 seconds. Other defaults that are listed in the table are not shown during configuration but are applied when the configuration details are saved if value is not specified.

Following table describes the LDAP monitor configurations:

Table 72. LDAP monitor configuration

Field	Description
server	The name or IP address of the LDAP server to be monitored. For example, `ldap.mycompany.in`.
searchbase	The distinguished name of the location from which to start the search. For example, `ou=Accounting, c=UK`.
filter	An attribute of the target entity to search for. For example, `(uid=ham)`.
description	A text field for providing descriptive information on the element. For example, `LDAP monitor`.
Active	Selects whether the profile element is to be activated after it is created. For example, `Selected`.
port	The port on the LDAP server to connect to. You must specify the SSL port if you are using SSL authentication. Default: 389
username	The username used to log in to the directory service. The format of the username depends on the setting of Authentication Type. You can specify a Windows domain, that is, `DOMAIN\username`. For example, `jbloggs`.
password	The password used to log in to the directory service, if necessary. For example, `secret9`.

Table 72. LDAP monitor configuration (continued)

Field	Description
authenticationtype	The LDAP authentication method to use: • SIMPLE (anonymous or plain text password) • SSL-SIMPLE • SASL-DIGEST-MD5 **Note:** SASL-DIGEST-MD5 authentication is not available on Linux operating system. If you set the authenticationtype to SIMPLE or SSL-SIMPLE, enter the username in distinguished name format. If you set the authenticationtype to SASL-DIGEST-MD5, enter the username as SASL bind-ids. To log in to the LDAP server as an anonymous user, set the authenticationtype to SIMPLE and leave the username and password fields blank. Default: SIMPLE
saslrealm	The authentication realm for the LDAP server; usually the fully qualified domain name of the server. If you want to share passwords between multiple systems, you might use a domain name. For example, `mycompany.com`.
timeout	The time, in seconds, to wait for the server to respond. Default: 30
poll	The time, in seconds, between each poll. Default: 300
failureretests	The number of times to retest before failure is indicated. Default: 0
retestinterval	The time, in seconds, to wait between each failure retest. Default: 10

Service level classifications

Available service level classification options for the LDAP monitor are:

```
totalTime
connectTime
searchTime
initTime
dnMatched
message
```

In service level classifications:

- Specify extra service level classifications by manually entering the name of the monitor element. The name must match the name that is shown for the element in the Monitor elements section.
- `message` can be any message that is forwarded in the **`$message`** element to IBM Application Performance Management server if used in any widget. For a list of possible values, see Status messages.
- The operand is a string or a positive number.

Monitor elements

In addition to the test results common to all elements, the LDAP monitor generates a set of test results that contains data specific to LDAP service tests.

The following table lists the additional elements for the LDAP monitor.

Table 73. LDAP monitor elements

Element	Description
$authentication	The type of user authentication method required by the LDAP server (Standard or CRAM-MD5).
$connectTime* (ConnectTime)	The time taken to connect to the LDAP server.
$distinguishedName* (UserName)	The distinguished name used to log in to the directory service.
$dnMatched	The entity matched in the search.
$filter* (SrchFilter)	The attribute used to locate the target entity.
$initTime* (InitTime)	The time taken to initialize the LDAP client.
$port* (Port)	The port on the LDAP server to which the monitor is connected.
$saslRealm	The SASL realm that you specified after new LDAP element is added.
$searchBase* (SearchBase)	The distinguished name of the entity from which the search was started.
$searchTime* (SearchTime)	The time taken to complete the search.

Status messages

The LDAP monitor provides status messages in the **`ResultMessage`** attribute when using IBM Application Performance Management. These messages indicate the result of the test.

The following table describes the LDAP status messages.

Table 74. LDAP monitor status messages

Message	Description
`Search successful`	The request succeeded.
`Search failed`	The request failed
`No match`	The server might not find a matching entry in the search criteria.

Table 74. LDAP monitor status messages (continued)

Message	Description
`Connection timed out`	The connection succeeded, but then the server stopped responding.
`Initialisation failed - an unrecognised authentication type was specified`	Occurs if an authentication type is used that the LDAP monitor does not support.
`Client initialisation failed`	Initialization of the LDAP structures failed because of inadequate memory.
`Bind (authentication) failed`	The server that is waiting for the bind to complete is timed out.
`SASL bind is not possible because server does not support LDAPv3`	The server must support LDAPv3 to create an SASL bind.
`SASL bind is not possible because one of 'bind_id' (username), password or sasl_realm is blank`	In order for a bind to occur, all authentication fields must have a value. Therefore, an SASL bind is not possible if the user is logged in anonymously (in plain text) by using the SIMPLE authentication type.
`SASL bind error`	The reason for SASL bind failure cannot be identified.
`SASL bind authorisation error`	The SASL bind failed because the authorization credentials were incorrect.

IMAP4 monitor

The IMAP4 monitor works with the SMTP monitor to test the availability and response time of an IMAP4 email service.

The following table lists the IMAP4 monitor files.

Table 75. IMAP4 monitor files

Monitor files	Name or location
Monitor executable	`nco_m_imap4`
Properties file	`$ISHOME/etc/props/imap4.props`
Rules file	`$ISHOME/etc/rules/imap4.rules`
Log file	`$ISHOME/log/imap4.log`

Guidelines for configuring IMAP4 monitor

The IMAP4 monitor works with the SMTP monitor by monitoring the mailbox to which the SMTP monitor sends test messages, and measuring that amount of time taken to deliver those messages.

Note: Ensure that the system clocks on the monitor host computer and the mail server are synchronized for the delivery time calculation to work correctly.

When the IMAP4 monitor has read the contents of the mailbox, it generates two different types of events:

- Message-specific events

  The IMAP4 monitor creates a message-specific event for each email message that it downloads from the mailbox. In this type of event, the monitor sets the `$message` element to `Message Successfully Downloaded`. The `$timeToDeliver` element is calculated as the time taken for the message to travel between the SMTP monitor that issued it and the mailbox that received it. The `$hopCount` element indicates the number of hosts the message hopped through to arrive at the mailbox.

- Summary events

  The monitor creates a summary event when it has processed all the messages in the mailbox. In this type of event, the `$message` element indicates the total number of messages successfully downloaded from the mailbox and the `$totaltime` element indicates the time taken to complete the requests. The `$totaltime` is in seconds.

Secure mail

The IMAP4 monitor supports connections to secure mail services. It can connect using SSL/TLS, or the STARTTLS command. When defining a profile element, use the securitytype field to select the appropriate security. If the mail server requires a client-side certificate for SSL encryption, use the SSL properties to specify a certificate file, key file, key password and cipher suite.

Client-side certificates

The IMAP4 monitor enables you to monitor servers that require client-side certificates for mutual authentication.

Specify the SSL certificate file, key file, and key password when creating a profile element.

Certificates must be in Privacy Enhanced Mail (PEM) format. If your certificate is in another format, you must convert it to PEM format. Certificates can be converted using software such as openSSL, which is available from http://www.openssl.org.

Note: If you always use the same certificate, key, and password in all profile elements, specify them using monitor properties instead of defining them in every profile element you create.

Mail boxes

After the IMAP4 monitor processed information contained in an email message sent by the SMTP monitor, it deletes it from the mailbox. You can use any existing mailbox to store email messages between the two monitors, even if the mailbox belongs to a real user. However, it’s recommended that you create a special mailbox account for service testing.

Configuring IMAP4 monitor service tests

Use the IMAP4 monitor configuration parameters to define service tests.

When you configure the monitor, default values are shown for the timeout and poll interval parameters. These defaults are 30 and 300 seconds respectively. Other defaults listed in the table aren’t shown during configuration but are applied when the configuration details are saved if no value has been specified.

Table 76. IMAP4 monitor configuration

Field	Description
server	The IP address of the mail server. Example is `test.mycompany.com`
description	A text field for providing descriptive information on the element.
port	The IP Port of the IMAP4 Server. Default: 143

Table 76. IMAP4 monitor configuration (continued)

Field	Description
securitytype	The type of secure connection opened with the mail server: • NONE - Connect without security. • SSL - Send an SSLv2 hello, then negotiate SSLv2, SSLv3 or TLSv1. • STARTTLS - Connect without security, issue a STARTTLS command, then establish a connection over TLSv1. Default: NONE
username	The name of the mailbox.
password	The password used to log in to the mailbox, if necessary.
authenticationtype	The method of authentication to use (STANDARD or CRAM_MD5) Default: STANDARD
sharedsecret	The shared secret for CRAM_MD5 authentication if applicable.
timeout	The time, in seconds, to wait for the server to respond. Default: 30
poll	The time, in seconds, between each poll. Default: 300
failureretests	The number of times to retest before indicating a failure. Default: 0
retestinterval	The time, in seconds, to wait between each failure retest. Default: 10

Regular expression matching

You can perform a regular expression search on the information being downloaded by entering up to 50 different regular expressions. The monitor attempts to match the contents retrieved to each of the regular expressions.

If a match for a specified regular expression is found, the matched lines (or as much as can fit in the monitor's internal buffer) are returned in the corresponding `$regexpMatchn` element. If the regular expression matches more than once in the information downloaded, only the first match is returned. The status of each regular expression test is indicated by the `$regexpStatusn` elements. You can use the regular expression matches and their status information as criteria for service level classifications.

For information about regular expression syntax, see Table 50 on page 322.

Monitor elements

In addition to the test results common to all elements, the IMAP4 monitor generates a set of test results containing data specific to IMAP4 service tests.

The following table describes the additional elements for the IMAP4 monitor.

Elements indicated by an asterisk (*) are available as attributes. The names of the attributes are shown within brackets. Absence of an asterisk indicates there's no equivalent attribute. Attributes shown in bracket but without an element indicate that they are only available as attributes, there's no equivalent element.

Table 77. IMAP4 monitor elements

Element	Description
$authentication	The type of user authentication method required by the IMAP4 server (Standard or CRAM-MD5).
$bytesPerSec	The average number of bytes transferred each second.
$bytesTransferred	The number of bytes uploaded or downloaded.
$connectTime	The time taken to connect to the IMAP4 server.
$downloadTime* (DownloadTime)	The time taken to download the file.
$hopCount	The number of hosts the message hopped through to reach the mailbox.
$inEvent	Indicates that this event is part of a series of events. 1 indicates that it’s not the final event. 0 indicates that it’s the final event.
$lookupTime*(LookupTime)	The time taken to obtain the IP address of the host server.
$port*(Port)	The port on which the service is monitored.
$responseTime* (ResponseTime)	The time between when the connection is established and the first byte of data is received.
$security	The type of secure connection opened with the mail server specified when adding an IMAP element (NONE, STARTTLS or SSL).
$sentTo*(SentTo)	The email address used by the SMTP monitor to send the original message.
$smtpServer	The name of the SMTP server from which the email was sent.
$SSLHandshakeTime* (SslHandshakeTime)	The time taken to establish the SSL connection.
$timeToDeliver	The time taken for an email message to travel between an SMTP monitor and its destination mailbox.
$user*(ImapUser)	The username (account name) used by the monitor to log in to the IMAP4 server.

Status message

The IMAP monitor provides status messages in the ResultMessage attribute when using IBM Application Performance Management. These messages indicate the result of the test.

The following table describes the IMAP4 status messages.

Table 78. IMAP4 monitor status messages

Message	Description
Message successfully downloaded	The message was successfully downloaded.
Downloaded x messages	Indicates how many messages were successfully downloaded from the mailbox.
Server not IMAP4rev1 compliant	The IMAP4 server doesn’t comply with the IMAP4 specification (RFC2060).

Table 78. IMAP4 monitor status messages (continued)

Message	Description
`Server does not support STARTTLS capability`	The server isn't configured correctly.
`Unable to log into server`	The monitor cannot log in to the IMAP server.
`Unrecognised response to STATUS command`	The monitor doesn't recognize the value returned by the server.
`Unrecognised response to FETCH INTERNALDATE command`	
`Failed to obtain Actual-Time-Sent header`	The monitor didn't obtain the expected response from the server.
`Failed to obtain Actually-To header`	
`Failed to obtain SMTP-Server header`	

Properties

The properties and command-line options specific to the IMAP4 monitor are described in the following table.

Table 79. IMAP4 monitor properties

Property name	Property parameter	Description
`Originator`	string	Specifies the `From` field to match when retrieving test email messages sent by the SMTP monitor. The monitor retrieves only messages where the `From` field matches the string in Originator. The IMAP4 Originator must match the Originator in the SMTP monitor. Default: `SMTP-Monitor`
`SSLCertificate File`	string	The path and filename of the digital certificate file used if no certificate is explicitly specified for an HTTPS element during its creation. If the path isn't absolute, the monitor interprets it relative to the working directory (`$ISHOME/platform/arch/bin`).
`SSLCipherSuite`	string	The cipher suite to use for SSL operations. For a description of possible values, see Cipher Suites. Default: `RC4:3DES:DES:+EXP`
`SSLDisableTLS`	integer	Disables TLSv1 for heritage support. Default: `0` - TLSv1 is enabled. 1 - TLSv1 is disabled.
`SSLKeyFile`	string	The file containing the SSL private key.
`SSLKeyPassword`	string	The password used to encrypt the SSL private key.

Cipher suites

The SSLCipherSuite property specifies the cipher suite used by the IMAP4 monitor. For more information about SSL settings, see "SSL setting in Internet Service Monitoring" on page 428.

NTP Monitor

The Network Time Protocol (NTP) monitor queries an NTP server by using UDP (User Datagram Protocol) to determine whether the server is supplying the correct time.

NTP uses Coordinated Universal Time to synchronize computer clocks to millisecond.

The following table lists the NTP monitor files.

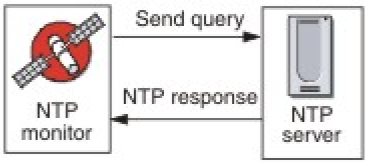
Table 80. NTP monitor files

Monitor Files	Name and Location
Monitor executable	`nco_m_ntp`
Properties file	`$ISHOME/etc/props/ntp.props`
Rules file	`$ISHOME/etc/ntp.rules`
Log file	`$ISHOME/log/ntp.log`

Guidelines for configuring NTP monitor

The NTP monitor acquires data by sending a query to an NTP server, which returns a UDP response packet with the current time (as seen by the NTP server).

The following image shows an example of the messages that are exchanged between the monitor and the NTP server.



Configuring NTP monitor service tests

Use the NTP monitor configuration parameters to define service tests.

The following table describes the NTP configurations:

Table 81. NTP configuration

Field	Description
server	The host name of the NTP server. Example is `ntp.mycompany.com`.
description	A text field for providing descriptive information on the element. Example is `NTP monitor`.
port	The port on the NTP server to use. Default: 123
localip	Specifies the IP address of the network interface on the host system to which the monitor binds when it performs the test. If the monitor's `IpAddress` property is set, it overrides the value of this field. Example is `102.168.n.n`.

Table 81. NTP configuration (continued)

Field	Description
version	The version of the NTP server to use (1, 2, 3, or 4). Default: 1
timeout	The time, in seconds, to wait for the server to respond. Default: 10
retries	The number of times the monitor retries to contact the NTP server. Default: 0
poll	The time, in seconds, between each poll. Default: 300
failureretests	The number of times to retest before failure is indicated. Default: 0
retestinterval	The time, in seconds, to wait between each failure retest. Default: 10

Service level classification

Service level classifications define the rules for determining the level of service that is provided over NTP.

Available service level classification options for the NTP monitor are:

```
totalTime
responseTime
lookupTime
offset
adjustedOffset
message
```

In service level classifications:

- Specify more service level classifications by manually entering the name of the monitor element. The name must match the name that is shown for the element in the Monitor elements section.
- `message` can be any message that is forwarded in the `$message` element toIBM Application Performance Management server if used in any widget. For a list of possible values, see Status messages.
- The operand is a string or a positive number.

Monitor elements

In addition to the test results common to all elements, the NTP monitor generates a set of test results that contains data specific to NTP service tests.

The following table describes the additional elements for the NTP monitor.

Table 82. NTP monitor elements

Element	Description
`$adjustedOffset`	The time offset from the server in seconds.

Table 82. NTP monitor elements (continued)

Element	Description
$localIP	The local IP address the monitor is configured to use. It might be blank on a system with only one interface.
$lookupTime* (LookupTime)	The time taken to obtain the IP address of the host server.
$ntpVersionIn	Protocol version used in the response from the server.
$ntpVersionOut	Protocol version that is used for sending.
$offset	The difference in time between the NTP server and the system that runs the monitor in seconds.
$port* (Port)	The port on the NTP server to use.
$responseTime* (ResponseTime)	The time between the monitor to connect to the NTP server and receive a response.
$retries	The number of times to resend a request if no response IDs are received.

Status messages

The NTP monitor provides status messages in the ResultMessage attribute when you use IBM Application Performance Management. These messages indicate the result of the test.

The following table describes the NTP status messages.

Table 83. NTP monitor status messages

Message	Description
Successful query	The NTP server gave the expected response.
Connection failed	Unable to initialize a UDP socket.
Failed to send request to NTP server	Unable to write to UDP socket.
No response from server	The NTP server did not respond.

NNTP Monitor

The NNTP monitor tests the availability of an NNTP service by reading from and posting to a newsgroup.

The following table lists the NNTP monitor files.

Table 84. NNTP monitor files

Monitor files	Name or location
Monitor executable	nco_m_nntp

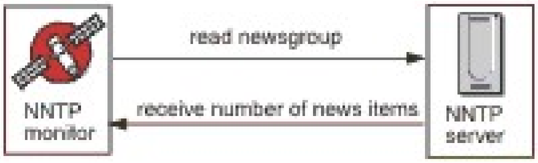
Table 84. NNTP monitor files (continued)

Monitor files	Name or location
Properties file	`$ISHOME/etc/props/nntp.props`
Rules file	`$ISHOME/etc/rules/nntp.rules`
Log file	`$ISHOME/log/nntp.log`

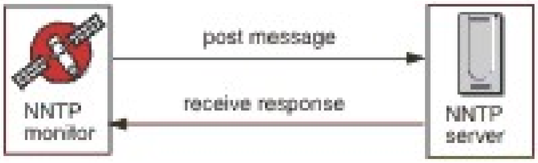
Guidelines for configuring the NNTP monitor

The NNTP monitor tests NNTP services by posting to and reading from an NNTP server. Each profile element that you create for the monitor performs either a read operation or a post operation.

In a read operation, the monitor connects to the NNTP service to check whether a particular Internet newsgroup exists. If the newsgroup exists, the monitor records the number of news items in it. It also attempts to record the last news item that is added to the newsgroup. The following image shows the read operation.



In a post operation, the monitor checks that the newsgroup exists and then attempts to write a test message to it. The subject of the test message is `NNTP Monitor Test Message`. The following image shows the post operation.



Each profile element specifies a `username` and `password` that is supplied by the monitor when it is accessing an NNTP server. The monitor uses the plain text authentication system.

```
AUTHINFO USER username
AUTHINFO PASS password
```

Where the `username` and `password` are specified in the monitor profile element.

Properties

Properties options specific to the NNTP monitor are described in the following table.

Table 85. NNTP monitor properties options

Property name	Property parameter	Description
OutputDirectory	string	Specifies the output directory to use if **`OutputResult`** is true (set to 1). Default: `$ISHOME/var`

Table 85. NNTP monitor properties options (continued)		
Property name	**Property parameter**	**Description**
OutputResult	0|1	Specifies that the monitor can save the data that it receives from the service. 0 - disabled 1 - enabled

Configuring NNTP monitor service tests

Use the NNTP monitor configuration parameters to define service tests.

The following table lists the NNTP monitor configurations.

Table 86. NNTP monitor configuration

Field	**Description**
server	The IP address of the NNTP server. For example, `news.mycompany.com`.
newsgroup	The name of the newsgroup the monitor uses for posting and reading test messages. For example, `mycompany.test`.
description	A text field for providing descriptive information on the element. For example, `READ`.
port	The port number of the NNTP server. Default: `119`
username	The username used to authenticate with the NNTP server.
password	The password of the username used to authenticate with the NNTP server.
action	Indicates whether to post or retrieve an article. It can be `READ` or `POST`. Default: `POST`
timeout	The time, in seconds, to wait for the server to respond. Default: `30`
poll	The time, in seconds, between each poll. Default: `300`
failureretests	The number of times to retest before failure is indicated. Default: `0`
retestinterval	The time, in seconds, to wait between each failure retest. Default: `10`

Matching regular expression

You can perform a regular expression search on the information that is being downloaded by entering up to 50 different regular expressions. The NNTP monitor attempts to match the contents that are retrieved to each of the regular expressions.

If a match for a specified regular expression is found, the matched lines (or as much as can fit in the monitor's internal buffer) are returned in the corresponding `$regexpMatchn` element. If the regular expression matches more than once in the information downloaded, only the first match is returned. The status of each regular expression test is indicated by the `$regexpStatusn` elements. You can use the regular expression matches and their status information as criteria for service level classifications.

For more information, see Table 50 on page 322.

Service level classifications

Service level classifications define the rules for determining the level of service that is provided over NNTP.

Available service level classification options for the NNTP monitor are:

```
totalTime
lookupTime
connectTime
transferTime
responseTime
status
bytesTransferred
bytesPerSec
newsItems
expected
lastLineReceived
checksum
previousChecksum
regexpMatch1 to 3
regexpStatus1 to 3
message
```

In service level classifications:

- Specify more service level classifications by manually entering the name of the monitor element. The name must match the name that is shown for the element in the Monitor elements section.
- `message` can be any message that is forwarded in the `$message` element to IBM Application Performance Management server if used in a widget. For a list of possible values, see Status messages.
- The operand is a string or a positive number.
- `status` codes `220` and `240` indicate success. See the NNTP protocol for other status codes returned by the operation.
- `egexpStatus`*n* might have the following values:
  - `NONE`: No regular expression checking is configured
  - `MATCHED`: A match was found for the regular expression
  - `FAILED`: A match was not found for the regular expression
- Evaluate regular expression matches that uses test expressions of the format:

  ```
  regexpMatchn [contains|!contains] expression
  ```

  Use `contains` and `!contains` operators in place of = and `!`= because `regexpMatch`*n* normally contains the entire line that matches the regular expression instead of just the matching portion, so the = and `!`= operators often do not match the expression.
- The `Checksum` and `PreviousChecksum` elements do not normally provide meaningful values for service level classifications because checksum values are not known when the profile element is created (the monitor calculates checksum values while tests are in progress). The `$checksum` and `$previousChecksum` monitor elements are intended for alert enrichment by using the monitor's rules file.

Monitor elements

In addition to the test results common to all elements, the NNTP monitor generates a set of test results that contains data specific to NNTP service tests.

The following table describes the additional elements for the NNTP monitor.

Table 87. NNTP monitor elements

Element	Description
$action* (NntpAction)	The action taken by the monitor. It can be READ or POST.
$bytesPerSec	The average number of bytes transferred each second.
$bytesTransferred	The number of bytes uploaded or downloaded.
$checksum	The Checksum element does not normally provide meaningful values for service level classifications because checksum values are not known when the profile element is created (the monitor calculates checksum values while tests are in progress). The $checksum and $previousChecksum monitor elements are intended for alert enrichment by using the monitor's rules file.
$connectTime* (ConnectTime)	The time taken to establish a connection to the NNTP server.
$downloadTime	The time taken to download the file.
$group* (NntpGroup)	The name of the monitored newsgroup.
$lastLineReceived	This element is only set if the $message element contains the message Expect Failed. If it is set, it contains the NNTP server's response.
$lookupTime* (LookupTime)	The time taken to look up the server IP address.
$newsItems	The number of news items in the news group.
$password	The password used to authenticate the monitor.
$previousChecksum	The PreviousChecksum element does not normally provide meaningful values for service level classifications because checksum values are not known when the profile element is created (the monitor calculates checksum values while tests are in progress). The $previousChecksum and $checksum monitor elements are intended for alert enrichment by using the monitor's rules file.
$responseTime* (ResponseTime)	The time taken, after a connection is created, until the first byte of the target article is received.

Table 87. NNTP monitor elements (continued)

Element	Description
`$status`	The status code returned by the NNTP server.
`$transferTime*` (`TransferTime`)	Sets the value to `$uploadTime` or `$downloadTime`.
`$uploadTime`	The time taken to upload the file.
`$username`	The username used to authenticate the monitor.
If `$message` contains `$ExpectFailed`	
`$expected`	The text in the connection that the monitor was waiting for when the connection failed.
`$lastLineReceived`	The last line of text in the connection that the monitor received from the NNTP server.

Status messages

The NNTP monitor provides status messages in the `ResultMessage` attribute when using IBM Application Performance Management. These messages indicate the result of the test.

The following table describes the NNTP status messages.

Table 88. NNTP monitor status messages

Message	Description
`Article Posted`	The NNTP POST action succeeded.
`Article Retrieved`	The NNTP READ action succeeded.
`Not Found`	The article might not be located.
`Expect failed`	The NNTP request failed.
`Timed out waiting to read`	A data connection to the server was established, but it stopped responding.
`Connection failed`	The monitor failed to connect to the server. For more information, see the log file.
`Connection closed by foreign host`	The remote host closed the connection before the monitor expected it to.

POP3 Monitor

The POP3 monitor works along with the SMTP monitor to test the availability and response time of a POP3 email service.

The following table lists the POP3 monitor files.

Table 89. POP3 monitor files

Monitor files	Name or location
Monitor executable	`nco_m_pop3`
Properties file	`$ISHOME/etc/props/pop3.props`
Rules file	`$ISHOME/etc/rules/pop3.rules`
Log file	`$ISHOME/log/pop3.log`

Guidelines for configuring POP3 monitor

The POP3 monitor operates along with the SMTP monitor by monitoring the mailbox to which the SMTP monitor sends test messages, and measuring that amount of time taken to deliver those messages.

Note: Ensure that the system clocks on the monitor host computer and the mail server are synchronized for the delivery time calculation to work correctly.

When the POP3 monitor has read the contents of the mailbox, it generates two different types of events:

- Message-specific events

  The POP3 monitor creates a message-specific event for each email message that it downloads from the mailbox. In this type of event, the monitor sets the `$message` element to `Message Successfully Downloaded`. The `$timeToDeliver` element is calculated as the time taken for the message to travel between the SMTP monitor that issued it and the mailbox that received it. The `$hopCount` element indicates the number of hosts the message hopped through to arrive at the mailbox.

- Summary events

  The monitor creates a summary event when it has processed all the messages in the mailbox. In this type of event, the `$message` element indicates the total number of messages successfully downloaded from the mailbox and the `$totaltime` element indicates the time taken to complete the requests. The `$totaltime` is in seconds.

Secure mail

The POP3 monitor supports connections to secure mail services. It can connect using SSL/TLS, or the STARTTLS command. When defining a POP3 monitor element, use the Security Type field to select the appropriate security. If the mail server requires a client-side certificate for SSL encryption, use the SSL*name* properties or command line options to specify a certificate file, key file, key password and cipher suite.

Client-side certificate

The POP3 monitor enables you to monitor servers that require client-side certificates for mutual authentication. Specify the SSL certificate file, key file, and key password while creating a profile element. Certificates must be in Privacy Enhanced Mail (PEM) format. If your certificate is in another format, you must convert it to PEM format. Certificates can be converted using software such as openSSL, which is available from http://www.openssl.org.

Note: If you always use the same certificate, key, and password in all profile elements, specify them using monitor properties instead of defining them in every profile element you create.

Configuring POP3 monitor tests

Note: Monitor the operation of the mail server `mail.mycompany.com` by configuring the SMTP monitor to send messages to a test mailbox, and configuring the POP3 monitor to retrieve the messages. The test mailbox has the address `ismtest@mycompany.com` and credentials `ismtest/secret1`. Use a connection timeout of 20 seconds, 2 retests on failure, and a retest interval of 5 seconds at each end, and test the services every ten minutes. Use the default service level classifications provided by the profile elements.

Table 90. POP3 monitor configuration

Field	**Description**
server	The IP address of the mail server. Example is `mail.mycompany.com`
description	A text field for providing descriptive information on the element.
port	The port number of the mail server. Default: `110`
securitytype	The type of secure connection opened with the mail server: • `NONE` - Connect without security • `SSL` - Send an SSLv2 `hello`, then negotiate SSLv2, SSLv3 or TLSv1 • `STARTTLS` - Connect without security, issue a STLS command, then establish a connection over TLSv1. This is the most secure security type. • `NONE` - Connect without security Default: `NONE`
username	The name of the mailbox.
password	The password used to log in to the mailbox, if necessary.
authenticationtype	The method of authentication to use and the label is Authentication Type: • `STANDARD` - Uses a user/pass exchange where the password isn't encrypted. This is appropriate for intermittent use of POP3. • `APOP` - Use where the POP3 client connects to the server regularly. This offers a higher level of security than standard. Ensure that you specify an APOP Shared Secret if you select `APOP`. Note that not all servers support APOP. Default: `STANDARD`.
sharedsecret	The shared secret for APOP authentication, applicable only if you're using the APOP authentication type. The string should be at least eight characters long and is obscured in the user interface.
timeout	The time, in seconds, to wait for the server to respond. Default: `30`
poll	The time, in seconds, between each poll. Default: `300`
failureretests	The number of times to retest before indicating a failure. Default: `0`
retestinterval	The time, in seconds, to wait between each failure retest. Default: `10`

Table 90. POP3 monitor configuration (continued)

Field	Description
verifycertificate	The verification certificate of the server. Default: `Disabled`

Use the POP3 monitor configuration parameters to define service tests.

Regular expression matching

You can perform a regular expression search on the information being downloaded by entering up to 50 different regular expressions. The monitor attempts to match the contents retrieved to each of the regular expressions.

If a match for a specified regular expression is found, the matched lines (or as much as can fit in the monitor's internal buffer) are returned in the corresponding `$regexpMatchn` element. If the regular expression matches more than once in the information downloaded, only the first match is returned. The status of each regular expression test is indicated by the `$regexpStatusn` elements. You can use the regular expression matches and their status information as criteria for service level classifications.

For information about regular expression syntax, see Table 50 on page 322.

Monitor elements

In addition to the test results common to all elements, the POP3 monitor generates a set of test results containing data specific to POP3 service tests.

Table 1 describes the additional elements for the POP3 monitor.

Elements indicated by an asterisk (*) are available as attributes. The names of the attributes are shown within brackets. Absence of an asterisk indicates there's no equivalent attribute. Attributes shown in bracket but without an element indicate that they are only available as attributes, there's no equivalent element.

Table 91. IMAP4 monitor elements

Element	Description
`$authentication`	The type of user authentication method required by the IMAP4 server (Standard or CRAM-MD5).
`$bytesPerSec`	The average number of bytes transferred each second.
`$bytesTransferred`	The number of bytes uploaded or downloaded.
`$connectTime`	The time taken to connect to the IMAP4 server.
$downloadTime* (`DownloadTime`)	The time taken to download the file.
`$hopCount`	The number of hosts the message hopped through to reach the mailbox.
`$inEvent`	Indicates that this event is part of a series of events. `1` indicates that it isn't the final event, `0` indicates that it's the final event.
`$lookupTime*`(`LookupTime`)	The time taken to obtain the IP address of the host server.
`$port*`(`Port`)	The port on which the service is monitored.
`$responseTime*` (`ResponseTime`)	The time between when the connection is established and the first byte of data is received.

Table 91. IMAP4 monitor elements (continued)

Element	Description
`$security`	The type of secure connection opened with the mail server specified when adding an IMAP element (NONE, STARTTLS or SSL).
`$sentTo*(SentTo)`	The email address used by the SMTP monitor to send the original message.
`$smtpServer`	The name of the SMTP server from which the email was sent.

`$SSLHandshakeTime*`
`(SslHandshakeTime)`
	The time taken to establish the SSL connection.

`$timeToDeliver`	The time taken for an email message to travel between an SMTP monitor and its destination mailbox.
`$user*(ImapUser)`	The username (account name) used by the monitor to log in to the IMAP4 server.

Status message

The POP3 monitor provides status messages in the `ResultMessage` attribute when using IBM Application Performance Management. These messages indicate the result of the test.

The following table describes the POP3 status messages.

Table 92. POP3 monitor status messages

Message	Description
`Message successfully downloaded`	The POP3 request was successful.
`Downloaded` *x* `messages`	Indicates how many messages were downloaded from the mailbox.
`Timed out waiting to read/write`	A data connection to the server was established, but it has stopped responding.
`Connection closed by foreign host`	The remote host closed the connection before the monitor expected.
`Connection failed`	The monitor failed to connect to the server. See the log file for more information.
`APOP not supported by the server`	The APOP authentication method isn’t supported by the server. Use the Standard authentication type instead.
`APOP service not available`	The APOP server implementation isn’t supported by the monitor. Use the Standard authentication type instead.
`Server does not support STLS capability`	The server doesn’t support STARTTLS. Use a different security type.

Properties

Properties specific to the POP3 monitor are described in the following table.

Table 93. POP3 monitor properties and command-line options

Property name	Property parameter	Description
SSLCertificate File	string	The path and filename of the digital certificate file used if no certificate is explicitly specified for a POP3 element during its creation. If the path isn't absolute, the monitor interprets it relative to the working directory ($ISHOME/platform/*arch*/bin).
SSLCipherSuite	string	The cipher suite to use for SSL operations. Default: RC4:3DES:DES:+EXP. See Cipher suites for a description of the possible values.
SSLDisableTLS	integer	Disables TLSv1 for legacy support. Default: 0 - TLSv1 is enabled. Set to 1 to disable TLSv1.
SSLKeyFile	string	The file containing the SSL private key.
SSLKeyPassword	string	The password used to encrypt the SSL private key.

Cipher suites

The SSLCipherSuite property specifies the cipher suite used by the POP3 monitor. For more information about SSL settings, see "SSL setting in Internet Service Monitoring" on page 428.

RADIUS Monitor

Remote Authentication Dial-In User Service (RADIUS) provides authentication for remote access to services. The RADIUS monitor simulates a client system that access a RADIUS service and returns data about the performance of the service.

The following table lists the RADIUS monitor files.

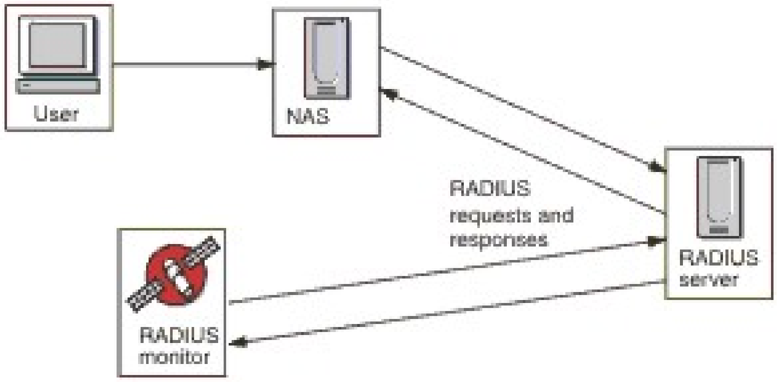
Table 94. RADIUS monitor files

Monitor Files	Name and Location
Monitor executable	nco_m_radius
Properties file	$ISHOME/etc/props/radius.props
Rules file	$ISHOME/etc/rules/radius.rules
Log file	$ISHOME/log/http.log

Guidelines for configuring Radius monitor

The RADIUS monitor simulates the operation of a Network Access Server (NAS), sending requests to a RADIUS server.

The RADIUS monitor uses UDP to send requests to the RADIUS server and then generates events that contain the results of those requests and data about the server performance. The following image shows the operation of the monitor.



The monitor can test both the authentication and accounting operations of RADIUS servers:

- Access-Requests by using Password Authentication Procedure (PAP)
- Access-Requests by using Challenge-Handshake Authentication Protocol (CHAP)
- Accounting-Requests: Start, Stop, Accounting On, and Accounting Off

Properties

The properties options specific to the RADIUS monitor are described in the following table.

Table 95. RADIUS monitor properties options

Property name	Property parameter	Description
`FramedServiceRequest`	`0`|`1`	When this property is set to `1`, the monitor selects the Framed service type set in Access-Requests. `0` - disabled `1` - enabled

Configuring Radius monitor service tests

Use the RADIUS monitor configuration parameters to define service tests.

The following table describes the Radius monitor configurations:

Table 96. RADIUS monitor configuration

Field	Description
server	The IP address of the RADIUS server.
sharedsecret	The shared secret used to authenticate the monitor.
username	The username supplied by the monitor to authenticate the RADIUS server.
password	The password supplied by the monitor to authenticate the RADIUS server.
description	A text field for providing descriptive information on the element.

Table 96. RADIUS monitor configuration (continued)

Field	**Description**
requesttype	Specifies the type of request sent to the RADIUS server: • `Authenticate (CHAP)` • `Authenticate (PAP)` • `Accounting` Default: `Authenticate(CHAP)`
port	The port to use to connect to the RADIUS server. Default: `1812`
localip	Specifies the IP address of the network interface on the host system to which the monitor binds when it performs the test. If the monitor's `IpAddress` property is set, it overrides the value of this field.
loginhost	Sets the value of the `Login-IP-Host` attribute in the Access-Request.
calledstation	Sets the value of the `Called-Station-Id` attribute in the Access-Request.
callingstation	Sets the value of the `Calling-Station-Id` attribute in the Access-Request.
accountsessionid	Sets the value of the `Acct-Session-Id` attribute in Accounting-Request packets sent to the accounting server. **Note:** This field applies to the `Accounting` request type only.
accountstatustype	Sets the value of the `Acct-Status-Type` attribute in Accounting-Request packets sent to the accounting server: • `Start` • `Stop` • `Accounting On` • `Accounting Off` **Note:** This field applies to the `Accounting` request type only. Default: `Start`
accountsessiontime	Sets the value of the `Acct-Session-Time` attribute (in seconds) in accounting-request packets sent to the accounting server. **Note:** This field applies to the `Accounting` request type only.
nasip	The `NAS-IP-Address` attribute that is sent by the RADIUS monitor as part of an Access-Request packet.
nasport	The `NAS-Port` attribute that is sent by the RADIUS monitor as part of an access request packet.

Table 96. RADIUS monitor configuration (continued)

Field	Description
timeout	The time, in seconds, to wait for the server to respond. Default: `10`
retries	The number of times to retry to connect to the RADIUS server if there is a problem. Default: `0`
poll	The time, in seconds, between each poll. Default: `300`
failureretests	The number of times to retest before failure is indicated. Default: `0`
retestinterval	The time, in seconds, to wait between each failure retest. Default: `10`

Service level classification

Service level classifications define the rules for determining the level of service that is provided by RADIUS service.

Available service level classification options for the RADIUS monitor are:

```
totalTime
lookupTime
responseTime
message
```

In service level classifications:

- Specify more service level classifications by manually entering the name of the monitor element. The name must match the name that is shown for the element in the Monitor elements section.
- `message` can be any message that is forwarded in the `$message` element to IBM Application Performance Management server if used in any widget. For a list of possible values, see Status messages.
- The operand is a string or a positive number.

Monitor elements

In addition to the test results common to all elements, the RADIUS monitor generates a set of test results that contain data specific to RADIUS service tests.

The following table describes the additional elements for the RADIUS monitor.

Table 97. RADIUS monitor elements

Element	Description
`$accountSessionId`	Unique identifier used to match start and stop records.
`$accountSessionTime`	When `accountStatusType` is set to `Stop`, this field shows the amount of time that the user receives the service, in seconds.

Table 97. RADIUS monitor elements (continued)

Element	Description
$accountStatusType	Indicates whether it is the start of the user's service (start) or the end (stop).
$calledStationId	The RADIUS monitor sends the calledStationId as part of an Access-Request packet. It is used if the RADIUS server requires it and not used if callingStationId is used.
$callingStationId	The RADIUS monitor sends callingStationId as part of an Access-Request packet. It is used if the RADIUS server requires it and not used if calledStationId is used.
$localIP	The local IP address the monitor is configured to use. It might be blank on a system with only one interface.
$loginIPHost* (LoginIpHost)	The RADIUS monitor sends loginIPHost as part of an Access-Request Packet. It might be required by servers that are being monitored.
$lookupTime* (LookupTime)	The time taken to obtain the IP address of the host server.
$nasPort* (NasPort)	NAS Port parameter that is sent by the RADIUS monitor as part of an Access-Request packet. Default: 0.
$password	The password used to authenticate the monitor.
$port* (Port)	The port on which the service is monitored.
$requestType	Indicates the request type that is selected for the element, either PAP, CHAP, or Accounting.
$responseTime	The time that is taken between sending a request to the RADIUS server and receiving a reply from it.
$retries	The maximum number of retries.
$secret	The shared secret password taken from the configuration file.
$username* (RadiusUser)	The user name used to authenticate the monitor.

Status messages

The RADIUS monitor provides status messages in the ResultMessage attribute when using IBM Application Performance Management. These messages indicate the result of the test.

The following table describes the RADIUS monitor status messages

Table 98. RADIUS monitor status messages

Message	Description
`CHAP authentication - Access granted`	The monitor was authenticated (using CHAP). It is only returned if the CHAP request type was used.
`PAP authentication - Access granted`	The monitor was authenticated (using PAP). It is only returned if the PAP request type was used.
`Accounting response received`	An accounting response was received from the server. The transaction is continued.
`Connection failed`	The server name that is specified is invalid.
`Failed to send request to RADIUS server`	Might not write UDP packet to the network. No further error information is available.
`No response from server`	The RADIUS server is not responding.
`Incorrect identifier returned`	There was a response from the server to a request that was not sent from the monitor.
`Invalid response authenticator`	The response contained an authorization that was not expected. It might be caused by an incorrect shared secret or password.
`Unrecognized response`	The server did not recognize the packet that is sent.
`PAP authentication - Access denied`	The monitor was not authenticated (using PAP).
`CHAP authentication - Access denied`	The monitor was not authenticated (using CHAP).

RPING Monitor

The RPING monitor tests the availability of network devices by pinging them remotely from a router. It provides maximum, minimum, and average round-trip time performance data.

The monitor supports Cisco, Juniper routers, and RFC2925-compliant routers.

The following table lists the RPING Monitor files.

Table 99. RPING monitor files

Monitor files	Name or location
Monitor executable	`nco_m_rping`
Properties file	`$ISHOME/etc/ims/props/rping.props`
Rules file	`$ISHOME/etc/ims/rules/rping.rules`
Log file	`$ISHOME/log/rping.log`

Table 99. RPING monitor files (continued)

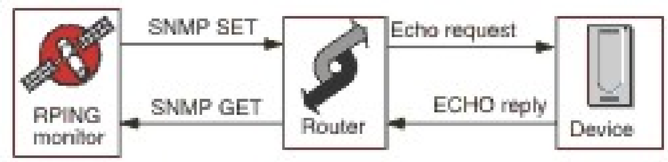
Monitor files	Name or location
Script files	`$ISHOME/scripts/rping/cisco.s` (SNMP script for Cisco routers) `$ISHOME/scripts/rping/juniper.s` (SNMP script for Juniper routers) `$ISHOME/scripts/rping/rfc2925.s` (SNMP script for RFC2925-compliant routers)

Guidelines for configuring RPING monitor

The RPING monitor acquires data by configuring the router to ping a network device, then periodically polling the router to obtain the results of the pings.

The monitor configures the ping tests by using an SNMP SET command to create a control row in the router's ping MIB, then it retrieves the ping data from the MIB by using SNMP GET commands. All communication with the router is over SNMP.

The following image shows an example of the messages that are exchanged between the monitor and the network device.



Enabling remote Ping request on Cisco routers

By default, remote ping SNMP requests on Cisco routers are disabled. However, for the RPING monitor to do an SNMP SET request and to start pinging, this request must be enabled.

To enable the request, log in to the Cisco router and enter the following commands:

```
enable
config terminal
snmp-server community communitystring rw
write mem
logout
```

The communitystring configured in the router must match the string that you enter in the R/W Community String field of any RPING profile elements that are created for that router. The write mem line ensures that the settings are saved when the router is rebooted.

Enabling remote Ping request on Juniper routers

By default, remote ping SNMP requests that on Juniper routers are disabled. For the RPING monitor to operate by using a Juniper router, you must enable SNMP requests.

To enable SNMP request on the router, ensure that the SNMP section of the JUNOS configuration match:

```
[edit snmp]
view ping-mib-view {
  oid .1.3.6.1.2.1.80 include; # pingMIB
  oid jnxPingMIB include; # jnxPingMIB
}
community communitystring {
  authorization read-write;
  view ping-mib-view;
}
```

The communitystring configured in the router must match the string that you enter in the communitystring field of any RPING profile elements that are configured for that router.

Properties

Properties options specific to the RPING monitor are described in the following table.

Table 100. RPING properties options

Property name	Property parameter	Description
MibDir	string	The directory that contains MIB files used by the monitor. Default: `$ISHOME/mibs`.

Configuring the RPING monitor service tests

Use the RPING monitor configuration parameters to define service tests.

Table 101. RPING monitor configuration

Field	Description
server	The name or IP address of the router. For example, `rt1.mycompany.com`.
routertype	The type of router: - CISCO - Juniper - RFC2925
host	The name or IP address of the server you want the router to ping.
communitystring	Specifies the SNMP community string that is used to communicate with the router. For example, `server1.mycompany.com`.
description	A text field for providing descriptive information about the element. For example is, `RPING monitor`.
vpn	The optional name of a VPN to use for sending pings. The router uses the VPN specified instead of the default route configured.
version	The SNMP version to use: 1 - SNMPv1 2 - SNMPv2c 3 - SNMPv3 Default: 2
numberofpings	The number of pings to send. Default: 5
packetsize	The size of the packets to send, in bytes. Default: 64
packettimeout	The time to wait between pings in seconds. Default: 500

Table 101. RPING monitor configuration (continued)

Field	Description
securityname†	The username for the SNMP session.
authenticationphrase†	The authentication password for the user.
privacyphrase†	The privacy password for the user.
authenticationprotocol†	The protocol to use to authenticate the user: • MD5 • SHA1 Default: MD5
privacyprotocol†	The protocol to use for encrypting the session. Default: DES
timeout	The time, in seconds, between each poll. Default: 10
retries	The number of times the monitor retries to contact the server. Default: 3
poll	The time to wait between pings in seconds. Default: 300
failureretests	The number of times to retest before failure is indicated. Default: 0
retestinterval	The time, in seconds, to wait between each failure retest. Default: 10
† Applicable only to SNMPv3.	

Service level classification

Service level classifications define the rules for determining the level of service that is provided over RPING.

Available service level classification options for the RPING monitor are:

```
totalTime
lookupTime
numPacketSent
numPacketsRecv
maxRTT
minRTT
averageRTT
respondPercent
message
```

In service level classifications:

- Specify more service level classifications by manually entering the name of the monitor element. The name must match the name that is shown for the element in the Monitor elements section.

- `message` can be any message that is forwarded in the **`$message`** element to IBM Application Performance Management server if used in any widget. For a list of possible values, see Status messages.
- The operand is a string or a positive number.

Monitor elements

In addition to the test results common to all elements, the RPING monitor generates a set of test results that contain data specific to RPING service tests.

The following table lists the additional elements for the RPING monitor.

Table 102. RPING monitor elements

Element	Description
$authProto	The authentication protocol as specified when the element was created.
(AverageRTT)	The average round-trip time in seconds.
$community	The SNMP community string for the router.
$communityString	The SNMP community string used to communicate with the router.
(MaxRTT)	The maximum round-trip time in seconds.
(MinRTT)	The minimum round-trip time in seconds.
$numPacketSent	The number of packets sent by the monitor.
$numPings	The number of pings sent, as specified when the RPING element was added.
$packetSize	The size of packets to send.
$packetTimeout	The time to wait between sending packets.
$privProto	The privacy protocol as specified when the element was created.
$remoteHost* (RemoteHost)	The name or IP address of the server you want the router to ping.
(RespondPercent)	The percentage of pings sent for which there was a response.
$routerMan* (RouterName)	The router type selected when the RPING element was added: • CISCO • Juniper • RFC2925
$securityName	The security user name as specified when the element was created.
(SnmpVersion)	The version of SNMP used to send SNMP packets (version 1, 2c, or 3).
(SourceRouter)	The name or IP address of the router.
$timeout	The number of seconds in which the server must respond. Taken from the configuration file.

Table 102. RPING monitor elements (continued)

Element	Description
$vpn* (Vpn)	The name of the VPN specified in the **vpn** field of the RPING profile element.

Status messages

The RPING monitor provides status messages in the **ResultMessage** attribute when using IBM Application Performance Management. These messages indicate the result of the test.

The following table describes the RPING status messages.

Table 103. RPING monitor status messages

Message	Description
`Got Response`	The monitor received a response from the Cisco device.
`Error in packet - exiting thread`	There was an error in one of the packets.
`Timed out while trying initial sets`	There was no response from the router when you try to create the `rowEntry` field.
`Internal Error`	Error in the router.
`Host poll did not finish`	The network device did not finish the pings.

`Response Failed`
`Operation Failed`
	The router was unable to ping the network device.

`Timed out on Get requests`	The monitor timed out when you try to get the results from the router.

RTSP Monitor

The Real Time Streaming Protocol (RTSP) monitor tests audio and video stream playback on streaming servers. It gathers information about media files, and initiates stream playback, pause, and the end of a streaming session.

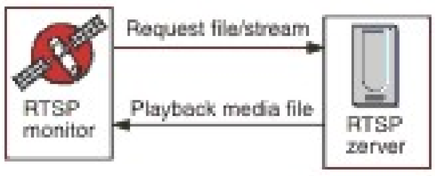
The following table lists the RTSP monitor files.

Table 104. RTSP monitor files

Monitor files	Name or location
Monitor executable	`nco_m_rtsp`
Properties file	`$ISHOME/etc/props/rtsp.props`
Rules file	`$ISHOME/etc/rules/rtsp.rules`
Log file	`$ISHOME/log/rtsp.log`

Guidelines for configuring the RTSP monitor

The RTSP monitor connects to the streaming server in either DESCRIBE or PLAY mode. The monitor downloads information or statistics that are delivered by genuine RTSP servers such as Darwin.



DESCRIBE Mode

In the DESCRIBE mode, the RTSP monitor connects to the streaming server and requests information about the audio and video files and streams.

The server returns a status code where a value of `200` indicates a file that can be downloaded, and where other values indicate that why the requested file cannot be played.

However, the statistics that relate to playback are not reported in this mode, the basic function of the servers that support RTSP can be tested.

PLAY Mode

In PLAY mode, the RTSP monitor connects to the streaming server in the same way as in DESCRIBE mode, and then streams the file to provide statistics on requested downloads.

Properties

Properties options specific to the RTSP monitor are described in the following table.

Table 105. RTSP monitor properties options

Property name	Property parameter	Description
StreamingSocket BufferSize	integer	The size of the streaming socket buffer, with a range of 8 to 64 KB. Default: 8

Configuring the RTSP monitor service tests

Use the RTSP monitor configuration parameters to define service tests.

The following table lists the RTSP monitor configurations:

Table 106. RTSP monitor configuration

Field	Description
server	The target system that runs the streaming server. For example, `rtsp.mymusic.com`.
remotefile	The file that gets downloaded. For example, `singalong.mp3`.
description	A text field for providing descriptive information on the element. For example, RTSP `monitor`.
port	The port that the monitor connects to on the target system. Default: 554

Table 106. RTSP monitor configuration (continued)

Field	Description
action	The action that the server performs on the stream: • DESCRIBE • PLAY Default: DESCRIBE
duration	The portion of the stream, in seconds, that the server plays back. Default: 5
maxbandwidth	The maximum bandwidth, n bits per second, that is used for streaming. Default: 1500000
timeout	The time, in seconds, to wait for the RTSP server to respond. Default: 10
poll	The time, in seconds, between each poll. Default: 300
failureretests	The number of times to retest before failure is indicated. Default: 0
retestinterval	The time, in seconds, to wait between each failure retest. Default: 10

Service level classifications

Service level classifications define the rules for determining the level of service that is provided over RTSP.

Available service level classification options for the RTSP monitor are:

```
totalTime
lookupTime
connectTime
responseTime
sdpDownloadTime
playbackTime
status
percentPacketsLost
message
```

In service level classifications:

- Specify more service level classifications by manually entering the name of the monitor element. The name must match the name that is shown for the element in the monitor elements section.
- `message` can be any message that is forwarded in the **`$message`** element to IBM Application Performance Management server if used in any widget. For a list of possible values, see.Status messages.
- The operand is a string or a positive number.
- A status code of 200 indicates success. See the RTSP protocol for other status codes returned by the operation.

Monitor elements

In addition to the test results common to all elements, the RTSP monitor generates a set of test results that contain data specific to RTSP service tests.

The following table describes the additional elements for the RTSP monitor.

Table 107. RTSP monitor elements

Element	Description
$action	The action taken by the monitor.
$averageBandwidth	The average total bandwidth, in bits.
$bytesReceived	The total number of bytes received.
$connectTime* (`ConnectTime`)	The time taken to establish a connection with the target server.
$describeStageStatus	Status code for a stage of the RTSP conversation.
`$filename`	The name of the media file.
$lookupTime* (`LookupTime`)	The time taken to obtain the IP address of the host server.
$maxBandwidth	The maximum bandwidth by using the configuration interface.
$mediaResponseTime	The time taken by the server to start streaming the requested file.
$numberOfStreams	The number of streams embedded in the media.
$percentPacketsLost	The percentage of packets lost.
$playbackTime* (`PlaybackTime`)	The time that represents the sum of the `setupResponseTime` and the `mediaResponseTime`.
$playStageStatus	Status code for a stage of the RTSP conversation.
$port	The port used to access the monitor's server.
$responseTime* (ResponseTime)	The time from when the connection is established to when the first byte of data is received.
$sdpDownloadTime* (SdpDownloadTimed)	The time taken to download data about the media file.
$setupResponseTime	The time that represents part of the `playbackTime`. **Note:** The element is only generated when the RTSP monitor is operating in PLAY mode.
$setupStageStatus	Status code for a stage of the RTSP conversation.

Table 107. RTSP monitor elements (continued)

Element	Description
$status	The status code returned by the RTSP server.
$streamingTime	The time taken by the server to complete streaming the requested file.
$streamLength	The length of the longest stream in the media file.
$teardownStageStatus	Status code for a stage of the RTSP conversation.
$totalBandwidthRequired	The total bandwidth, in kilobits per second.
$totalPacketsLost	The total number of packets lost.
$totalPacketsReceived	The number of packets received.

Status messages

The RTSP monitor provides status messages in the **ResultMessage** attribute when using IBM Application Performance Management. These messages indicate the result of the test.

Table 108. RTSP monitor status messages

Message	Description
`OK`	The request succeeded.
`Connection failed`	The monitor failed to connect to the server. For more information, see the log file.
`Connection closed by foreign host`	The connection to the RTSP server was broken.
`Timed out waiting to read/write`	A data connection to the RTSP server was established, but a problem occurred.
`Play failed - no streams`	The monitor received a response, but no audio or video was available for playback.
`select() failed on RTSP socket (PLAY stage)`	The socket was closed from the remote server, or it timed out waiting for a response.
`RTSP Server response not in expected format`	The response from the server was in a format that the monitor does not support.
`Redirection requested by server not supported by client`	The response from the server is not supported by the client.
`Server cannot fulfill client request`	The request is failed and no further information is available.

Table 108. RTSP monitor status messages (continued)

Message	**Description**
`Server Error`	There was a problem with the server and the request failed. A code of `500` or greater was returned by the server. For more information, see the RTSP protocol (RFC 2326).
`RTSP response header CSeq doesn't match request CSeq`	The RTSP server is misconfigured and is not functioning correctly.
`Corrupted RTSP server response`	
`Corrupted session description`	
`RTSP SETUP response CSeq doesn't match request CSeq`	
`RTSP SETUP response, incomplete Session string`	
`RTSP SETUP response, Session ID has changed within the same session`	
`RTSP SETUP response does not contain server ports to connect to`	
`RTSP SETUP response does not contain server port pair to connect to`	
`RTSP PLAY response CSeq doesn't match request CSeq`	
`RTSP PLAY response, incomplete Session string`	
`RTSP PLAY response, Session ID has changed within the same session`	
`RTSP PLAY response, incomplete RTP-Info string`	
`RTSP PLAY response does not valid RTP seqnum in RTP-Info response`	
`RTSP PLAY response does not valid RTP time in RTP-Info response`	

SAA monitor

The Cisco Service Assurance Agent (SAA) is a performance monitoring agent for Cisco products for IOS version 12.2(2) and above.

The SAA monitor uses the Cisco Service Assurance Agent facility to test various timings between Cisco routers.

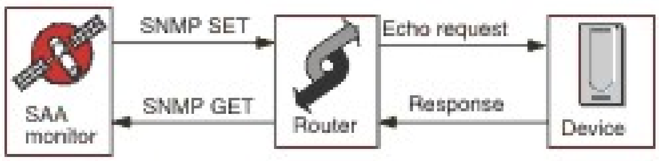
Table 109. SAA monitor summary

Monitor files	Name or location
Executable name	`nco_m_saa`
Properties file	`$ISHOME/etc/props/saa.props`
Rules file	`$ISHOME/etc/rules/saa.rules`
Log file	`$ISHOME/log/saa.log`
Scripts directory	`$ISHOME/scripts/saa/`

Guidelines for configuring the SAA monitor

The SAA monitor configures a router's SAA to test the availability of another network device or service by using timed echo request or responses that are defined in the Cisco Response Time Monitor's Management Information Base (MIB). The monitor uses Simple Network Management Protocol to communicate with the Service Assurance Agent.

The following image demonstrates the operation of the SAA monitor.



Operation

The SAA monitor configures the Service Assurance Agent to run echo tests, called probes, on other network devices. You can configure a range of different probes, each of which uses a different protocol.

All probes can operate against any IP enabled target, except Jitter, which requires another SAA capable Cisco responder router.

Each monitor profile element initiates an Service Assurance Agent probe on a router at startup and with each successive poll, it collects result information and reschedules the probe. If a probe stops unexpectedly, the monitor restarts it immediately. When the probe tests are complete, it goes into inactive state until the next monitor poll. In next monitor poll, the result data is collected and another test cycle begins. During each poll, the monitor checks the probe’s state. If the probe is still running, the monitor stops it and then polls the Management Information Base (MIB) for result data and error information from the last cycle. It then reschedules the probe, resets the statistical data, and reactivates the probe, which runs unattended until the next monitor poll.

To prevent the possibility of leaving uncontrolled processes on the router, the monitor starts probes with a predefined life span that is extended at each monitor poll. If the monitor is terminated, it continues to run until its life span expires. Thereafter, it moves into inactive state until an age-out time is reached and router ends the process.

It is not necessary to preconfigured IOS and Service Assurance Agent because the monitor automatically configures, controls, and cleans up after the probes at run time. This includes configuring responder routers that are needed by some probe types.

Probe persistence

The ProbePersist monitor property controls probe persistence across monitor polls. If probe persistence is not enabled, probes start at each poll and ends immediately after producing the test results.

Router load

At times, probe operations can be affected by router load. The StatusWait property provides probes with the time to change from one state to another before an operation is considered to be failed.

Probe types

The types of probes available with the SAA monitor are listed as follows:

- DHCP
- DLSW
- DNS
- FTP
- HTTP Get Requests
- ICMP Echo
- ICMP Path Echo
- Jitter
- UDP Echo
- SNA-Echo
- VOIP

Echo probes perform tests based on a timeframe, whereas Jitter, VOIP, and HTTP probes perform tests by single operation.

SAA properties

Yo must set the properties of the SAA monitor.

The following table describes the properties of SAA monitor.

Table 110. SAA monitor properties

Property name	Property parameter	Description
AgeOut	integer	The maximum number of seconds that a probe remains inactive before it stops. Default is 600.
MibDir	string	The directory that is used for MIB files. Default path is `$ISHOME/mibs`
ProbeLife	integer	The maximum number of seconds that a probe remains active when it's unattended. Default is 600.
ProbePersist	0|1	The probes can run in two modes. They perform a single test cycle per monitor poll or they are started once and are rescheduled at each poll. • 0 indicates single test cycle • 1 indicates reschedule at each poll
StatusWait	integer	The number of seconds a monitor waits for a probe to complete any action before it fails.

Configuring SAA monitor service tests

You must configure SAA monitor parameters to define service tests.

The following table described the SAA monitor configuration fields.

Table 111. SAA monitor configuration

Field	Description
server	The name or IP address of the Cisco router.
communitystring	The SNMP community string for the router.
probetype	The SAA probe type applicable to the profile element.
description	A text field for providing descriptive information for the element.
Active	Indicates whether the profile element is active.
port	The port used to access the router. Default port is 161.
version	The SNMP version to use: • 1 is used for SNMPv1 • 2 is used for SNMPv2c • 3 is used for SNMPv3 Default is 1.
probeid	Specifies a value that is used to generate the probe control row index.
securityname†	The username for the SNMP session.
authenticationphrase†	The authentication password for the user.
privacyphrase†	The privacy password for the user.
authenticationprotocol†	The protocols that are used for authenticating the users are as follows: • MD5 • SHA1 Default is MD5.
privacyprotocol†	The protocol to use for encrypting the session. This is DES.
timeout	The time, in seconds, to wait for the router to respond. Default is 5.
retries	The number of times the monitor retry to contact the router before it quits. Default is 0.
poll	The time, in seconds, between each poll. Default is 300
failureretests	The number of times to retest before it indicates a failure. Default is 0.
retestinterval	The time, in seconds, to wait between each retest on failure. Default is 10.

Note: † Applicable only to SNMPv3.

Probe type configuration

Probe configuration is different for each type of probe, and Internet Service Monitoring agent provides a set of configuration fields specific to each type. To create a profile element, select a probe type, then provide the configuration appropriate for that type. For information about individual configuration items, see the Cisco Response Time Monitor MIB document.

Service level classification

Service level classification defines the rules for determining the level of service that is provided by a network device.

Available service level classification options for the SAA monitor are as follows:

```
totalTime
errTotal
numRTT
minRTT
maxRTT
avgRTT
minPosJitterSD
maxPosJitterSD
minNegJitterSD
maxNegJitterSD
minPosJitterDS
maxPosJitterDS
minNegJitterDS
maxNegJitterDS
packetLossSD
packetLossDS
packetOutOfSequence
packetMIA
packetLateArrival
minDelaySD
maxDelaySD
minDelayDS
maxDelayDS
avgPosJitterSD
avgPosJitterDS
avgNegjitterSD
avgNegJitterDS
avgDelaySD
avgDelayDS
devPosJitterSD
devPosJitterDS
devNegJitterSD
devNegJitterDS
devDelaySD
devDelayDS
MOS
ICPIF
mMinRTT
httpRTT
dnsRTT
tcpConnectRTT
transactionRTT
message
```

In service level classifications:

- Specify more service level classifications by manually entering the name of the monitor element. The name must match the name that is shown for the element in the Monitor elements section.
- `message` can be any message that is forwarded in the **`$message`** element toIBM Application Performance Management server if used in any widget. For a list of possible values, see "Status messages" on page 397.
- The operand is a string or a positive number.

Monitor elements
In addition to the test results common to all elements, the SAA monitor generates a set of test results that contains data specific to the type of probe in use.

DHCP probes
DHCP probes generate multiple elements.

The following table describes the DHCP probe elements.

Table 112. DHCP probe elements

Element	Description
$authProto	The authentication protocol as specified when the element was created.
$community	The community used to send SNMP requests to the SAA.
$port	The port used to connect to the SAA.
$privProto	The privacy protocol as specified when the element was created.
$probeType	dhcp
$securityName	The security user name as specified when the element was created.
$snmpVersion (SnmpVersion)	The version of SNMP used to send SNMP packets (version 1, 2c, or 3).
(SourceRouter)	The name of the router used to send DHCP requests.
$totalRTT † (TotalRTT)	The total round-trip time taken to get an IP from the DHCP server in seconds.

Note: † indicates that the element is available for service level classifications.

DLSW probes
DLSW probes generate multiple elements.

The following table describes DLSW probe elements.

Table 113. DLSW probe elements

Element	Description
$authProto	The authentication protocol as specified when the element was created.
$avgRTT* † (AverageRTT)	The average round-trip time in seconds.
$community	The community used to send SNMP requests to the SAA.
$errTotal* † (ErrorTotal)	The total number of packets in error.
$maxRTT* † (MaximumRTT)	The highest round-trip time in seconds.
$minRTT* † (MinimumRTT)	The lowest round-trip time in seconds.

Table 113. DLSW probe elements (continued)

Element	Description
$numRTT†	The number of successful round trips.
$port	The port used to connect to the SAA.
$privProto	The privacy protocol as specified when the element was created.
$probeType †	dlsw
$securityName	The security user name as specified when the element was created.
$snmpVersion (SnmpVersion)	The version of SNMP used to send SNMP packets (version 1, 2c, or 3).
(SourceRouter)	The router used to perform the SAA test.
$sumOfRTT (TotalRTT)	The sum of all round-trip times in seconds.
(TargetHost)	The name or IP address of the host on which the target SAA is running.

Note: † indicates that the element is available for service level classifications.

DNS probes
DNS probes generate multiple elements.

The following table describes DNS probe elements.

Table 114. DNS probe elements

Element	Description
$authProto	The authentication protocol as specified when the element was created.
$community	The community used to send SNMP requests to the SAA.
$dnsHost (Host)	The host to resolve from server.
$dnsServer	The IP of the DNS server.
(HostLookup)	The IP address of the host.
$port	The port used to connect to the SAA.
$privProto	The privacy protocol as specified when the element was created.
$probeType	dns
$securityName	The security user name as specified when the element was created.
$snmpVersion (SnmpVersion)	The version of SNMP used to send SNMP packets (version 1, 2c, or 3).
(SourceRouter)	The name of the router used to send DNS requests.
$totalRTT † (TotalRTT)	The total round-trip time for the DNS lookup in seconds.

Note: † indicates that the element is available for service level classifications.

FTP probes
FTP probes generate multiple elements.

The following table describes FTP probe elements.

Table 115. FTP probe elements

Element	Description
$activePassive	The connection type used in the test, either `Active` or `Passive`. Default: `Passive`
$authProto	The authentication protocol as specified when the element was created.
$community	The community used to send SNMP requests to the SAA.
$errorStatus	The result string indicating the status of the test (from the `rttMonLatestRttOperSense` MIB object).
$ftpFile	The name of the test file retrieved during the test.
$ftpUrl (FtpUrl)	The URL used in the FTP test.
$port	The port used to connect to the SAA.
$privProto	The privacy protocol as specified when the element was created.
$securityName	The security user name as specified when the element was created.
$snmpVersion (SnmpVersion)	The version of SNMP used to send SNMP packets (version 1, 2c, or 3).
(SourceRouter)	The name of the router used to send FTP requests.
$totalRTT (TotalRTT)	The completion time of the test (from the `rttMonLatestRttOperCompletionTime` MIB object) in seconds.

HTTP-Get probes
HTTP-Get probes generate multiple elements.

The following table describes the HTTP-Get probe elements.

Table 116. HTTP-Get probe elements

Element	Description
$authProto	The authentication protocol as specified when the element was created.
$community	The community used to send SNMP requests to the SAA.
$dnsRTT † (DnsRTT)	The round-trip time to perform the DNS query in seconds.
$httpRTT † (HttpRTT)	The round-trip time to perform the HTTP operation in seconds.

Table 116. HTTP-Get probe elements (continued)

Element	Description
(HttpUrl)	The URL that is monitored.
$messageBodyBytes	The size of the message body received.
$numRTT†	The number of successful round trips.
$port	The port used to connect to the SAA.
$privProto	The privacy protocol as specified when the element was created.
$probeType	http-get
$securityName	The security user name as specified when the element was created.
(SourceRouter)	The name of the router used to send HTTP requests.
$snmpVersion (SnmpVersion)	The version of SNMP used to send SNMP packets (version 1, 2c, or 3).
$targetHost	The name of the host for the service that is being tested.
$tcpConnectRTT † (TcpConnectRTT)	The round-trip time to connect to the HTTP server in seconds.
$transactionRTT † (TransactionRTT)	The round-trip time to download the object specified by the URL in seconds.

Note: † indicates that the element is available for service level classifications.

ICMP-Echo probes
ICMP-Echo probes generate multiple elements.

The following table describes the ICMP-Echo probe elements.

Table 117. ICMP-Echo probe elements

Element	Description
$authProto	The authentication protocol as specified when the element was created.
$avgRTT † (AverageRTT)	The average round-trip time in seconds.
$community	The community used to send SNMP requests to the SAA.
$errBusies	The number of pings that are failed because a previous incomplete ping.
$errDisconnects	The number of pings that are failed through disconnects.
$ErrDrops	The number of pings failed because an internal resource was not available.
$errNoConnects	The number of pings that are failed because a connection to the target might not be established.
$errSequences	The number of pings failed because an unexpected sequence ID was received.

Table 117. ICMP-Echo probe elements (continued)

Element	Description
$errTimeouts	The number of pings failed through timeouts.
$errTotal † (ErrorTotal)	The total number of packets in error.
$errVerifies	The number of pings failed because the data received was not the same as the data expected.
$maxRTT † (MaximumRTT)	The highest round-trip time in seconds.
$minRTT † (MinimumRTT)	The lowest round-trip time in seconds.
$numRTT †	The number of successful round trips.
$port	The port used to connect to the SAA.
$privProto	The privacy protocol as specified when the element was created.
$probeType	The probe type must be as follows: • icmp-echo • icmp-echo-path • udp-echo
$securityName	The security user name as specified when the element was created.
$snmpVersion (SnmpVersion)	The version of SNMP used to send SNMP packets (version 1, 2c, or 3).
(SourceRouter)	The name of the router used to send ICMP requests.
$sumOfRTT	The sum of all round-trip times in seconds.
$targetHost (Host)	The host name of the service that is being monitored.
$tos (Tos)	The type of service value.
$vpn (Vpn)	The name of the VPN.

Note: † indicates that the element is available for service level classifications.

ICMP-Path-Echo probes
ICMP-Patch-Echo probes generate multiple elements.

The following table describes the ICMP-Patch-Echo probe elements.

Table 118. ICMP-Path-Echo probe elements	
Element	**Description**
$authProto	The authentication protocol as specified when the element was created.
$avgRTT † (AverageRTT)	The average round-trip time in seconds.
$community	The community used to send SNMP requests to the SAA.
(HopHost*One to Eight*)	The first to eighth host that visit using ICMP Echo Path.
$maxRTT † (MaximumRTT)	The highest round-trip time in seconds.
$minRTT † (MinimumRTT)	The lowest round-trip time in seconds.
$numRTT †	The number of successful round trips.
$port	The port used to connect to the SAA.
`$privProto`	The privacy protocol as specified when the element was created.
$probeType	The probe type is as follows: • `icmp-echo` • `icmp-echo-path`
$securityName	The security user name as specified when the element was created.
$snmpVersion (SnmpVersion)	The version of SNMP used to send SNMP packets (version 1, 2c, or 3).
(SourceRouter)	The name of the router used to send ICMP requests.
$sumOfRTT	The sum of all round-trip times in seconds.
$targetHost	The name of the host for the service that is being tested.
$tos (Tos)	The type of service value.
$vpn (Vpn)	The name of the VPN.

Note: † indicates that the element is available for service level classifications.

Jitter probes
Jitter probes generate multiple elements.

The following table describes the Jitter probe elements.

Table 119. Jitter probe elements

Element	Description
$authProto	The authentication protocol as specified when the element was created.
$avgDelayDS†	The average delay from destination to source in seconds.
$avgDelaySD†	The average delay from source to destination in seconds.
$avgNegJitterDS†	The average negative Jitter from destination to source in seconds.
$avgNegJitterSD†	The average negative Jitter from source to destination in seconds.
$avgPosJitterDS†	The average positive Jitter from destination to source in seconds.
$avgPosJitterSD†	The average positive Jitter from source to destination in seconds.
$avgRTT † (AverageRTT)	The average round-trip time in seconds.
$community	The community used to send SNMP requests to the SAA.
$devDelayDS†	The standard deviation of delay from destination to source.
$devDelaySD†	The standard deviation of delay from source to destination.
$devNegJitterDS†	The standard deviation of negative Jitter from destination to source.
$devNegJitterSD†	The standard deviation of negative Jitter from source to destination.
$devPosJitterDS†	The standard deviation of positive Jitter from destination to source.
$devPosJitterSD†	The standard deviation of positive Jitter from source to destination.
$errDescription	A description of the error.
$errTotal (ErrorTotal)	The total number of packets in error.
$maxDelayDS †	The maximum delay from destination to source in seconds.
$maxDelaySD †	The maximum delay from source to destination in seconds.
$maxNegJitterDS †	The maximum negative Jitter value from destination to source in seconds.
$maxNegJitterSD †	The maximum negative Jitter value from source to destination in seconds.
$maxPosJitterDS †	The maximum positive Jitter value from destination to source in seconds.
$maxPosJitterSD †	The maximum positive Jitter value from source to destination in seconds.
$maxRTT † (MaximumRTT)	The highest round-trip time in seconds.
$minDelayDS †	The minimum delay from destination to source in seconds.
$minDelaySD †	The minimum delay from source to destination in seconds.

Table 119. Jitter probe elements (continued)

Element	Description
$minNegJitterDS †	The minimum negative Jitter value from destination to source in seconds.
$minNegJitterSD †	The minimum negative Jitter value from source to destination in seconds.
$minPosJitterDS †	The minimum positive Jitter value from destination to source in seconds.
$minPosJitterSD †	The minimum positive Jitter value from source to destination in seconds.
$minRTT † (MinimumRTT)	The lowest round-trip time in seconds.
$numNegJitterDS	The number of negative Jitter values from destination to source.
$numNegJitterSD	The number of negative Jitter values from source to destination.
$numOW	The number of one-way operations for delay.
$numPosJitterDS	The number of positive Jitter values from destination to source.
$numPosJitterSD	The number of positive Jitter values from source to destination.
$numRTT †	The number of successful round trips.
$packetLateArrival †	The number of packets that arrived after timeout.
$packetLossDS †	The number of packets that are lost from destination to source.
$packetLossSD †	The number of packets that are lost from source to destination.
$packetMIA†	The number of packets that are lost where the direction is unknown.
$packetOutOfSequence†	The number of packets returned out of order.
$port	The port used to connect to the SAA.
$privProto	The privacy protocol as specified when the element was created.
$probeType†	Jitter
(ResponderRouter)	The name of the router that is used to respond to the Jitter requests.
$securityName	The security user name as specified when the element was created.
(SourceRouter)	The name of the router that is used to send Jitter requests.
$snmpVersion (SnmpVersion)	The version of SNMP used to send SNMP packets (version 1, 2c, or 3).
$sum2DelayDS	The sum of squares of delays from destination to source.
$sum2DelaySD	The sum of squares of delays from source to destination.
$sum2NegJitterDS	The sum of squares of all negative Jitter values.
$sum2NegJitterSD	The sum of squares of all negative Jitter values.
$sum2PosJitterDS	The sum of squares of all positive Jitter values.

Table 119. Jitter probe elements (continued)

Element	Description
$sum2PosJitterSD	The sum of squares of all positive Jitter values.
$sum2Rtt†	The sum of squares of the round-trip values in seconds.
$sumDelayDS	The sum of delays from destination to source in seconds.
$sumDelaySD	The sum of delays from source to destination in seconds.
$sumNegJitterDS	The sum of all negative Jitter values in seconds.
$sumNegJitterSD	The sum of negative Jitter values in seconds.
$sumPosJitterDS	The sum of all positive Jitter values in seconds.
$sumPosJitterSD	The sum of all positive Jitter values in seconds.
$sumRTT	The sum of all round trips in seconds.
$targetHost	The name of the host for the service that is being tested.
$tos (Tos)	The type of service value.
$vpn (\Vpn)	The name of the VPN.

Note: † indicates that the element is available for service level classifications.

SNA-Echo probes
SNA-Echo probes (SNA-RU-Echo, SNA-LU0-Echo, SNA-LU2-Echo, SNA-LU62-Echo, and SNA-LU62Native-Echo) generate the elements that are listed in the following table.

The following table describes the JSNA-Echo probe elements.

Table 120. SNA-Echo probe elements

Element	Description
$authProto	The authentication protocol as specified when the element was created.
$avgRTT † (AverageRTT)	The average round-trip time in seconds.
$community	The community used to send SNMP requests to the SAA.
$errTotal †	The total number of packets in error.
$maxRTT † (MaximumRTT)	The highest round-trip time in seconds.
$minRTT † (MinimumRTT)	The lowest round-trip time in seconds.
$numRTT†	The number of successful round trips.
$port	The port used to connect to the SAA.

Table 120. SNA-Echo probe elements (continued)

Element	Description
$privProto	The privacy protocol as specified when the element was created.
$probeType (ProbeType)	`sna-`*`name`*`-echo`
$securityName	The security user name as specified when the element was created.
(SourceRouter)	The name of the router that is used to send SNA requests.
$snmpVersion (SnmpVersion)	The version of SNMP used to send SNMP packets (version 1, 2c, or 3).
$sumOfRTT (TotalRTT)	The sum of all round-trip times in seconds.
(TargetHost)	The host target for the SNA echo request.

Note: † indicates that the element is available for service level classifications.

UDP-Echo probes
UDP-Echo probes generate the elements listed in the following table.

The following table describes the UDP-Echo probe elements.

Table 121. UDP-Echo probe elements

Element	Description
`$authProto`	The authentication protocol as specified when the element was created.
$avgRTT † (AverageRTT)	The average round-trip time in seconds.
$community	The community used to send SNMP requests to the SAA.
$errBusies	The number of pings failed because a previous incomplete ping.
$ErrDrops	The number of pings failed because internal resource was not available.
$errTimeouts	The number of pings failed through timeouts.
$errTotal † (ErrorTotal)	The total number of packets in error.
$errVerifies	The number of pings failed because the data received was not the same as the data expected.
$maxRTT † (MaximumRTT)	The highest round-trip time in seconds.
$minRTT † (MinimumRTT)	The lowest round-trip time in seconds.

Table 121. UDP-Echo probe elements (continued)

Element	Description
$numRTT †	The number of successful round trips.
$port	The port used to connect to the SAA.
$privProto	The privacy protocol as specified when the element was created.
$probeType	`udp-echo`
`$securityName`	The security user name as specified when the element was created.
$snmpVersion* (SnmpVersion)	The version of SNMP used to send SNMP packets (version 1, 2c, or 3).
$sumOfRTT	The sum of all round-trip times (in seconds).
$targetHost (Host)	The host name of the service that is monitored.
$tos (Tos)	The type of service value.
$vpn (Vpn)	The name of the VPN.

Note: † indicates that the element is available for service level classifications.

VOIP probes
VOIP probes generate the same elements as Jitter probes. In addition, they generate the elements that are listed in the following table.

The following table describes VOIP probe elements.

Table 122. VOIP probe elements

Element	Description
$authProto	The authentication protocol as specified when the element was created.
$avgRTT † (AverageRTT)	The average round-trip time in seconds.
$community	The community used to send SNMP requests to the SAA.
$errTotal † (ErrorTotal)	The total number of packets in error.
$ICPIF †	The ICPIF value.
$maxRTT † (MaximumRTT)	The highest round-trip time in seconds.
$minRTT † (MinimumRTT)	The lowest round-trip time in seconds.

Table 122. VOIP probe elements (continued)

Element	Description
$MOS †	The value of the Mean Opinion Score (MOS) from the test.
$port	The port used to connect to the SAA.
$privProto	The privacy protocol as specified when the element was created.
$probeType †	voip
(ResponderRouter)	The name of the router that is used to respond to the VOIP requests.
$securityName	The security user name as specified when the element was created.
$snmpVersion (SnmpVersion)	The version of SNMP used to send SNMP packets (version 1, 2c, or 3).
(SourceRouter)	The name of the router that is used to send the VOIP requests.
(Tos)	The type of service value.
(Vpn)	The name of the VPN.

Note: † indicates that the element is available for service level classifications.

Status messages
The SAA monitor provides status messages in the **ResultMessage** attribute when using IBM Application Performance Management. These messages indicate the result of the test.

The following table describes the status messages for SAA monitors.

Table 123. SAA monitor status messages

Message	Description
`Success`	The probe operation succeeded.
`Operation failed`	The probe operation failed.
`Invalid status`	The probe operation failed with an invalid status.

SIP Monitor

The SIP monitor checks the availability of Session Initiation Protocol (SIP) servers, including the time that is taken to register and authenticate end points. The monitor initiates a SIP session so that SIP requests and SIP responses can be monitored.

The following table lists the SIP monitor files.

Table 124. SIP monitor file summary

Monitor files	Name or location
Monitor executable	`nco_m_sip`
Properties file	`$ISHOME/etc/props/sip.props`
Rules file	`$ISHOME/etc/rules/sip.rules`
Log file	`$ISHOME/log/sip.log`

Guidelines for configuring the SIP monitor

The SIP monitor tests the availability of a SIP server by sending a request to the URI of a SIP enabled device, over the SIP server, and receiving, also over the SIP server, responses from the SIP device.

The SIP monitor acts as a User Agent Client (UAC); it initiates the connections that are used to test SIP services. The User Agent Server (UAS), the receiver or target of the call can be any SIP-enabled device such as a computer that runs a soft phone, or a message bank.

When testing a SIP server, the monitor undertakes the following sequence of actions:

1. Register with the SIP server by using the credentials supplied in the profile element.
2. Send an OPTIONS request to the UAS.
3. Send an INVITE request to the UAS.

   Register a successful test result if the UAS accepts the request.
4. Send a BYE request to the UAS and end the connection with the UAS.
5. Unregister from the SIP server, with immediate expiry.

The monitor records the duration of each action that is performed in the test.

Properties

The properties options specific to the SIP monitor are described in the following table.

Table 125. SIP monitor properties options

Property name	Property parameter	Description
ShowZeroes	0|1	Specifies the display of SIP statistics with zero values. 0 - disabled 1 - enabled
Transports	string	Lists local protocol port transports separated by a space that is TCP or UDP for the protocol. Wildcard port numbers are allowed. Default: UDP:*.

Cipher suites

The **`SSLCipherSuite`** property specifies the cipher suite that is used by the SIP monitor. For more information, see "SSL setting in Internet Service Monitoring" on page 428.

Configuring the SIP monitor service tests

Use the SIP monitor configuration parameters to define service tests.

Table 126. SIP monitor configuration

Field	Description
server	Specifies the name of the server to be tested. For example, `sip1.mycompany.com`.
serverport	The port through which the SIP monitor can reach the server to be tested.
username	Specifies the extension number or account identity of the SIP monitor that calls. For example, `jblogg`.
target	Specifies the extension number of a SIP enabled device that is used to make a call. For example, 5551234.
password	Specifies the password for the username.

Table 126. SIP monitor configuration (continued)

Field	Description
description	A text field for providing descriptive information on the element. For example, `SIP monitor`.
proxy	The host name of the proxy server. Example is, `proxy.mycompany.com`.
proxyport	The port through which the SIP monitor can reach the proxy server.
timeout	The time, in seconds, to wait for the server to respond. Default: 30
poll	The time, in seconds, between each poll. Default: 300
failureretests	The number of times to retest before failure is indicated. Default: 0
retestinterval	The time, in seconds, to wait between each retest on failure. Default: 10

Service level classification

Service level classifications define the rules for determining the level of service that is provided over SIP.

Available service level classification options for the SIP monitor are:

```
totalTime
message
```

In service level classifications:

- Specify more service level classifications by manually entering the name of the monitor element. The name must match the name that is shown for the element in the Monitor elements section.
- `message` can be any message that is forwarded in the **`$message`** element to IBM Application Performance Management server if used in any widget. For a list of possible values, see Status messages.
- The operand is a string or a positive number.

Monitor elements

In addition to the test results common to all elements, the SIP monitor generates a set of test results that contain data specific to SIP service tests.

The following table describes the additional elements for the SIP monitor.

Table 127. SIP monitor elements

Element	Description
$AcceptReg	The number of accepted SIP registration requests.
$AuthTime* (AuthenticationTime)	The time taken to authorize both the SIP monitor and the SIP enabled device.

Table 127. SIP monitor elements (continued)

Element	**Description**
$authAttempts	The number of times the monitor needed to resend a request to include its credentials.
(CallSetupTime)	The time taken to set up a call.
$Invalid	The number of invalid sent and received requests.
$InvalidReg	The number of invalid SIP registration requests.
$lastMethod	The last method the monitor saw other than BYE or ACK.
$lastSequence [METHOD]	The last sequence that is received for a method.
$lastStatus [METHOD]	The last status that is received for a method or overall.
$method *METHOD*	Tally of messages that are seen for a method.
$optionsTime* (OptionsTime)	The time taken to negotiate an options change (OPTIONS to 200 OK).
$postDialTime* (PostDialTime)	The time taken to receive a ringing signal after the dialing (INVITE to 180 Ringing).
$RegTime* (RegistrationTime)	The time taken to register both the SIP monitor and the SIP enabled device.
$registrationTime	The time taken to register with the server (REGISTER to 200 OK).
$Requests	The number of SIP Request messages that are received and sent.
($RequestsSent)	The number of SIP Request messages sent.

`${request|response}[Sent|`
`Received|Transmitted|Total]`
`[METHOD][STATUS]`
	Tally of messages seen for various categories, for example, `requestSentINVITE = 1`, `responseReceived = 10`, and `responseReceivedBYE200 = 1`.

$Responses	The number of SIP Response messages that are received and sent.
($ResponseReceived)	The number of SIP Response messages received.

Table 127. SIP monitor elements (continued)

Element	Description
$sessionAnswered	• 1 - if the call is answered • 0 - if the call is unanswered
$sessionCreated	• 1 - if a session is established • 0 - if a session is not established
$sessionTerminated	• 1 - if the session ends • 0 - if the session does not end.
$shutdownTime* (`ShutdownTime`)	The time taken to end the connection (BYE to 200 OK).
$terminatedReason* (TerminatedReason)	The reason for the connection closure.
(Username)	The user name used to log in to the SIP server.
(Target)	The target to open the session to.

Status messages

The SIP monitor provides status messages in the **`ResultMessage`** attribute when using IBM Application Performance Management. These messages indicate the result of the test.

The following table describes the SIP monitor status messages.

Table 128. SIP monitor status messages

Message	Description
`Register timed out`	The monitor failed to register to the server.
`Invite timed out`	The INVITE message timed out.
`OK`	The request and response succeeded.
`n operation status description`	• *n* is the message number sequence. • operation is type of message. • status is the status code. • description is a plain text description of the status. For example, `1 INVITE 200 OK`.

SIP responses

The SIP monitor supports the following types of responses. Each response has a 3-digit code:

- Informational Responses (100 - 199)
- Successful Responses (200 - 299)
- Redirection Responses (300 - 399)
- Client Failure Responses (400 - 499)
- Server Failure Responses (500 - 599)

- Global Failure Responses (600 - 699)

The following table lists the common SIP responses.

Table 129. Common SIP responses

Response	Description
100 Trying	The message is received by the SIP enabled device but yet to be processed.
180 Ringing	The message is received and processed by the SIP enabled device. The device is ringing to alert the user.
200 OK	This code is returned upon successful completion of a method. For example, the call is registered with the server or the user answered the call.
401 Unauthorized	The user is not authorized.
407 Proxy Authentication Required	This code is similar to 401, but indicates that the user must first authenticate.
408 Request Timeout	The user did not answer the call.

For a full list of SIP responses, see RFC3261.

SMTP Monitor

The SMTP monitor works in along with the IMAP4 or POP3 monitors to test the performance of an email service.

The following table lists the SMTP monitor files.

Table 130. SMTP monitor files

Monitor files	Name or location
Monitor executable	`nco_m_smtp`
Properties file	`$ISHOME/etc/props/smtp.props`
Rules file	`$ISHOME/etc/rules/smtp.rules`
Log file	`$ISHOME/log/smtp.log`

Guidelines for monitoring SMTP monitor

The SMTP monitor operates along with the POP3 or IMAP4 monitors. It periodically sends an email message to a mailbox on the target server and records the time taken to issue the send email request. The POP3 or IMAP4 monitor then reads the messages from the mailbox and uses them to calculate the response time and availability of the email service.

Note: The SMTP monitor operates along with the POP3 or IMAP4 monitors. It periodically sends an email message to a mailbox on the target server and records the time taken to issue the send email request. The POP3 or IMAP4 monitor then reads the messages from the mailbox and uses them to calculate the response time and availability of the email service.

Mailboxes

You may configure the monitor to send email messages to any existing mailbox, even if the mailbox belongs to a real user. However, it’s recommended that you create a special mailbox account for service testing. The email parameter specifies the recipient mailbox. By default, the monitor sends test messages with the subject line `SMTP Monitor Test Message`.If required, you can configure SMTP profile elements without a mailbox name. In this configuration, the monitor simply checks that the SMTP service is accepting connections.

Secure mails

The SMTP monitor supports connections to secure mail services. It can connect using SSL/TLS, or the STARTTLS command. When defining an SMTP monitor element, use the Security Type field to select the appropriate security. If the mail server requires a client-side certificate for SSL encryption, use the SSLname properties or command line options to specify a certificate file, key file, key password and cipher suite.

Client-side certificate

The SMTP monitor enables you to monitor servers that require client-side certificates for mutual authentication. You specify the SSL certificate file, key file, and key password when creating a profile element. Certificates must be in Privacy Enhanced Mail (PEM) format. If the certificate is in another format, you must convert it to PEM format. Certificates can be converted using software such as openSSL, which is available from http://www.openssl.org.

Note: If you always use the same certificate, key, and password in all profile elements, specify them using monitor properties instead of defining them in every profile element you create.

Configure the SMTP monitor service tests

Use the SMTP monitor configuration parameters to define service tests. When you configure the monitor, default values are shown for the timeout and poll interval parameters. These defaults are 30 and 300 seconds respectively. If no value is specified other defaults listed in the table aren't shown during configuration but are applied when the configuration details are saved.

Table 131. SMTP monitor configuration

Field	Description
server	The IP address of the mail server. Example is `mail.mycompany.com`
description	A text field for providing descriptive information on the element.
port	The port number of the mail server. Default: 25 If you use a server other than an SMTP server, update the port on which to connect to the server. For example, if you use an IMAP4 server over SSL for Microsoft Exchange, specify port 465.
securitytype	The type of secure connection opened with the mail server: • NONE - Connect without security • SSL - Send an SSLv2 hello, then negotiate SSLv2, SSLv3 or TLSv1 • STARTTLS - Connect without security, issue a STARTTLS command, then establish a connection over TLSv1 Default: NONE
username	The username used to log in to the SMTP server. Used with PLAIN or CRAM-MD5 authentication.
password	The password used to log in to the SMTP server. Used with PLAIN or CRAM-MD5 authentication.

Table 131. SMTP monitor configuration (continued)

Field	Description
authenticationtype	The method to authenticate the monitor to the SMTP server. The available options are: • NONE - No authentication is attempted • PLAIN - Plain text username and password authentication • CRAM-MD5- CRAM-MD5 authentication is used Default value is NONE.
sharedsecret	The shared secret key for CRAM-MD5 authentication.
email	The email address of the mailbox used by the SMTP and POP3 monitors.
timeout	The time, in seconds, to wait for the SMTP server to respond. Default: 30
poll	The time, in seconds, between each poll. Default: 300
failureretests	The number of times to retest before indicating a failure. Default: 0
retestinterval	The time, in seconds, to wait between each failure retest. Default: 10

Note: Monitor the availability of the mail server `mail.mycompany.com` by attempting to connect to it at 10-minute intervals. Use a connection timeout of 30 seconds and, if the connection fails, retry three times with 5 seconds between each retry.

Monitor element

In addition to the test results common to all elements, the SMTP monitor generates a set of test results containing data specific to SMTP service tests.

The following table describes the additional elements for the SMTP monitor.

Table 132. SMTP monitor elements

Element	Description
$authentication	The type of user authentication method required by the SMTP server (Standard or APOP).
$bytesPerSec	The average number of bytes transferred each second.
$bytesTransferred	The number of bytes uploaded or downloaded.
$connectTime* (ConnectTime)	The time taken to connect to the SMTP server.
$email* (EmailAddress)	The email address of the mailbox to which the monitor to sends test email.

Table 132. SMTP monitor elements (continued)

Element	Description
$lookupTime* (LookupTime)	The time taken to obtain the IP address of the host server.
$port* (Port)	The port on which the service is monitored.
$responseTime* (ResponseTime)	The time taken, after a connection is created, until the first byte of the test email can be sent to the SMTP server.
$security	The type of secure connection opened with the mail server (NONE, STARTTLS or SSL) as set in the profile element **securitytype** field.
$SSLHandshakeTime* (SslHandshakeTime)	The time taken to establish the SSL connection.
$status* (ResultStatus)	The status code returned by the SMTP server.
$uploadTime* (UploadTime)	The time taken to upload the file.
$user* (SmtpUser)	The username (account name) used by the monitor to log in to the SMTP server.

Status message

The SMTP monitor provides status messages in the $message element when using IBM Application Performance Management. These messages indicate the result of the test.

Properties

Properties specific to the SMTP monitor are described in the following table.

Table 133. SMTP monitor properties and command-line options

Property name	Property parameter	Description
MailMessage Path	string	Path to a file containing text to send in the test email. A default message is sent if this isn’t set.
Originator	string	Specifies the From field to set when sending the test email. Ensure that this matches the corresponding string in the IMAP4 monitor. Default: SMTP-Monitor.
SSLCertificate File	string	The path and filename of the digital certificate file used if no certificate is explicitly specified for an SMTP element during its creation. If the path isn’t absolute, the monitor interprets it relative to the working directory ($ISHOME/platform/arch/bin).

Table 133. SMTP monitor properties and command-line options (continued)

Property name	Property parameter	Description
SSLCipherSuite	string	The cipher suite to use for SSL operations. For a description of possible values, see Cipher suites. Default: RC4:3DES:DES:+EXP
SSLDisableTLS	integer	Disables TLSv1 for legacy support. Default: 0 - TLSv1 is enabled. 1 TLSv1 is disabled.
SSLKeyFile	string	The file containing the SSL private key.
SSLKeyPassword	string	The password used to encrypt the SSL private key.
UseBody	integer	Specifies where the monitor writes tracking information in the mail message, either in the mail header or mail body. Default: 0 - information is included in the mail header. 1 - write information to the mail body.

Cipher suites

The SSLCipherSuite property specifies the cipher suite used by the SMTP monitor. For more information about SSL settings, see "SSL setting in Internet Service Monitoring" on page 428.

SNMP Monitor

The SNMP monitor tests the SNMP enabled devices for performance and fault data.

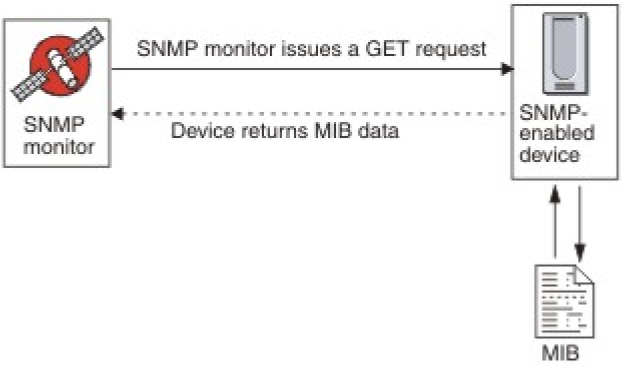
The following table lists the SNMP monitor files.

Table 134. SNMP monitor summary

Monitor files	Name or location
Monitor executable	nco_m_snmp
Properties file	$ISHOME/etc/props/snmp.props
Rules file	$ISHOME/etc/rules/snmp.rules
Log file	$ISHOME/log/snmp.log

Guidelines for configuration SNMP monitor

The SNMP monitor acquires data from SNMP enabled devices by sending SNMP GET requests for one or more objects that are contained in a device's MIB. The device then returns the MIB data to the SNMP monitor. The SNMP monitor supports SNMP versions 1, 2c, and 3.



Properties

Properties options specific to the SNMP monitor are described in the following table.

Table 135. SNMP monitor properties options

Property name	Property parameter	Description
InvalidBps Value	integer	Specifies an integer value that is substituted for bits per second (Bps) value calculations when only one data point is available.
MibDir	string	Specifies the directory that contains MIB documents that are used by the monitor. Default: `$ISHOME/mibs`.
StripQuotes	`0|1`	Strips quote characters from integer data. `0` - disabled `1` - enabled
Rollover Threshold	integer	The value that a delta must meet or exceed if a rollover happens before a router reset occurs. Default: `0` (never rollover)

Configuring the SNMP monitor service tests

Use the SNMP monitor configuration parameters to define service tests.

Table 136. SNMP monitor configuration

Field	Description
server	The server to send SNMP GET requests to.
objectgroupname	The text name for the group of OIDs to include in the GET request.
communitystring	The SNMP read/write community string for the SNMP server on the client. **Note:** Use a caret character (^) in community names advisedly, see Community names for further information.
description	A text field for providing descriptive information on the element.

Table 136. SNMP monitor configuration (continued)

Field	Description
port	The port on the server to use. Default: 161
version	The SNMP version to use: 1 - SNMPv1 2 - SNMPv2c 3 - SNMPv3 Default: 1
securityname†	The username for the SNMP session.
authenticationphrase†	The authentication password for the user.
privacyphrase†	The privacy password for the user.
authenticationprotocol†	The protocol to use to authenticate the user: • MD5 • SHA1 Default: MD5
privacyprotocol†	The protocol to use for encrypting the session. Default: DES
timeout	The time, in seconds, to wait for the server to respond. Default: 20
poll	The time, in seconds, between each poll. Default: 300
retries	The number of times the monitor retries to contact the server before it quits. Default: 0
failureretests	The number of times to retest before failure is indicated. Default: 0
retestinterval	The time, in seconds, to wait between each failure retest. Default: 10

Table 136. SNMP monitor configuration (continued)

Field	Description
hostnamelookuppreference	Determines which IP version, IPv6, or IPv4, is applied to the supplied host name. Options are: • `default` sets the monitor to use monitor-wide properties settings. This is the default. • 4Then6 selects IPv4 and then IPv6. Uses IPv4 addresses if they are available. If no IPv4 addresses are found, IPv6 addresses are used. • 6Then4 selects IPv6 and then IPv4. Uses IPv6 addresses if they are available. If no IPv6 addresses are found, IPv4 addresses are used. • `4Only` selects IPv4 only. Uses IPv4 addresses only. If there are no IPv4 addresses, the poll returns an error. • `6Only` selects IPv6 only. Uses IPv6 addresses only. If there are no IPv6 addresses, the poll returns an error. • `6Or4` selects either IPv4 or IPv6. Uses the first address that is returned from the host name.
† Applicable only to SNMPv3.	

Community names

Internet Service Monitoring uses the caret character (^) as an escape character as it sends information to the target device. If a community name contains a carat, you must enter two carats in a row (^^) for the name to be correct at the router. For example, for the community name a$^&b to be correct when sent to the device use a$^^&b.

Service level classifications

Service level classifications define the rules for determining the level of service.

Available service level classification options for the SNMP monitor are:

```
totalTime
message
```

In service level classifications.

- Specify more service level classifications by manually entering the name of the monitor element. The name must match the name that is shown for the element in the Monitor elements section.
- `message` can be any message that is forwarded in the **`$message`** element to IBM Application Performance Management server if used in any widget. For a list of possible values, see Status messages.
- The operand is a string or a positive number.
- oidName is the name that is assigned to a MIB object in the OID Name field defined in the OID group.

Monitor elements

In addition to the test results common to all elements, the SNMP monitor generates a set of test results that contain data specific to SNMP service tests.

The following table describes the additional elements for the SNMP monitor.

Table 137. SNMP monitor elements

Element	Description
$community	The SNMP community string for the SNMP server on the client.

Table 137. SNMP monitor elements (continued)

Element	Description
$numOids	The number of OIDs used in the query.
$oidGroupName* (OidGroup)	The name of the OID group. The OID group contains the OIDs that the monitor is polling.
$oidName*0 to n** (OIDName*Zero to Nine*)	The name of the first to the last MIB object in the OID group. It is indicated by a number when using Netcool/OMNIbus and by alphabetic text (zero to nine) when using IBM Application Performance Management.
$oidNames	The names of each OID separated by a vertical bar (|).
$oidReturnValues*0 to n** (snmpResult*Zero to Nine*)	The data that is returned by the SNMP GET command for the first to the last MIB object in the OID group. This is indicated by a number when using Netcool/OMNIbus and by alphabetic text (zero to nine) when using IBM Application Performance Management.
$oidUnit*0 to n*	The units for the first to the last MIB object in the OID group indicated by a number.
$oidUnits	The units for each OID, separated by a vertical bar character (|).
$port	The port on which the service is monitored.
$snmpVersion* (SnmpVersion)	Version of SNMP used to send SNMP packets configured in the profile (Version 1, 2c or 3).

Status message

The SNMP monitor provides status messages in the `ResultMessage` attribute when using IBM Application Performance Management. These messages indicate the result of the test.

The following table describes the SNMP monitor status messages.

Table 138. SNMP monitor status messages

Message	Description
`Successful Get`	The query of the SNMP agent was successful.

`Failed to open snmp sessions`
`SNMP session - start failed`
	Unable to initialize an SNMP session.

`Error in packet`	Unable to create a valid SNMP packet.
`Timed out while waiting for response`	No response received from the SNMP agent.
`Internal Error`	This was an internal error in the monitor. For more information, contact IBM Technical Support.
`Error Processing OID`	There was an error processing one of the `OID`s.

Table 138. SNMP monitor status messages (continued)

Message	Description
`ERROR: Too Many OIDs`	The monitor is set to request too many `OIDs` at one time. The maximum is `100`.
`ERROR: PDU received mismatch with PDU sent`	The protocol data unit (PDU) received by the monitor did not match the PDU sent to the server.

SOAP Monitor

The SOAP monitor checks the availability and response time of the SOAP interface (SOAP 1.0 and 1.1). It can also monitor the validity of SOAP inputs (requests) and SOAP outputs (responses).

The SOAP monitor supports the following message encoding styles:

- RPC - Encoded
- Document - Literal Unwrapped
- Document - Literal Wrapped

The following table lists the SOAP monitor files.

Table 139. SOAP monitor file summary

Monitor files	Name or location
Monitor executable	`nco_m_soap`
Properties file	`$ISHOME/etc/props/soap.props`
Rules file	`$ISHOME/etc/rules/soap.rules`
Log file	`$ISHOME/log/soap.log`

Guidelines for configuring the SOAP monitor

The SOAP monitor tests the operation of a SOAP service by sending the target SOAP interface a request that contains a set of inputs, and then receiving and analyzing the outputs that are contained in the response that is received from the interface. When a request is sent to the SOAP interface, the request can either succeed or fail. A request succeeds if a response is received and the values in the response message match the specified output values. A request fails if no response is received or a response is received but the values in its message do not match the output values.

The SOAP inputs and outputs that are contained in requests and responses depend on the functions of the SOAP service under test, when you design a test for a SOAP service, must specify inputs and outputs appropriate to that service. The inputs consist of the names of the data to be sent and their assigned input values. The outputs consist of the names of the data to be received and their expected output values. These data names are sourced from a local Web Service Description Language (WSDL) file, which you specify when you configure the SOAP monitor. The input and output data names must match the names and data types in the WSDL file. The data names must also be in the same order as in the WSDL file. If the names do not match, or the order is incorrect, an error message is generated when the monitor tries to poll the SOAP interface.

The format of inputs is:

```
dataname:datatype=assigned_value, dataname:datatype=assigned_value, ...
```

The format of outputs is:

```
dataname:datatype=expected_value, dataname:datatype=expected_value, ...
```

SOAP data types

The SOAP monitor supports simple, array, and user-defined data types. Simple data types include Integer, String, and Boolean. Arrays might contain simple data types and other array and user-defined data types.

Table 140. Simple data types

Simple data types			
anyURI	float	language	Qname
boolean	gDay	long	short
byte	gMonth	Name	string
date	gMonthDay	NCName	time
dateTime	gYear	negativeInteger	token
decimal	gYearMonth	NMTOKEN	unsignedByte
double	ID	NMTOKENS	unsignedInt
duration	IDREFS	nonNegativeInteger	unsignedLong
ENTITIES	int	nonPostiveInteger	unsignedShort
ENTITY	integer	normalizedString	

SOAP authentication

If the SOAP interface that you want to monitor requires basic HTTP authentication, specify credentials for accessing the interface in the SOAP profile element when using Internet Service Monitoring Configuration tool.

To set the required SOAP authentication parameters:

1. In the Internet Service Monitoring Configuration tool, select the profile element for which you want to add authentication information.
2. On the **Advanced** tab, click in the **Value** field for the `username` parameter and enter the required value.
3. Click in the **Value** field for the `password` parameter and enter the required value. The password is encrypted.
4. Click **OK**.

If authentication is no longer required, delete the values for the **`username`** and **`password`** parameters.

Properties

The properties options specific to the SOAP monitor are described in the following table.

Table 141. SOAP monitor properties options

Property name	Property parameter	Description	Default
SoapParser	string	XML parsing library.	`$ISHOME/platform/$ARCH/bin/AxisXMLParserXerces.dll`
SoapTransport	string	SOAP transport library.	`$ISHOME/platform/$ARCH/bin/HTTPTransport.dll`
SoapChannel	string	SOAP channel library	`$ISHOME/platform/$ARCH/bin/HTTPChannel.dll`
SoapSecureChannel	string	SOAP secure channel library.	`$ISHOME/platform/$ARCH/bin/HTTPSSLChannel.dll`
SoapClientLog	string	The name of the extra SOAP client log file.	`$ISHOME/log/SoapClient.log`

Cipher suites

The `SSLCipherSuite` property specifies the cipher suite that is used by the SOAP monitor.

For more information, see "SSL setting in Internet Service Monitoring" on page 428.

Configuring the SOAP monitor services tests

Use the SOAP monitor configuration parameters to define service tests.

Table 142. SOAP monitor configuration

Element	Description
wsdl	The path to a local copy of the WSDL file.
operation	The name of the SOAP operation.
operationnamespace	The namespace of the SOAP operation.
location	The URL of the SOAP service to be monitored.
description	A text field for providing descriptive information on the element.
timeout	The time, in seconds, to wait for the SOAP service to respond. Default: `10`
poll	The time, in seconds, between each poll. Default: `300`
failureretests	The number of times to retest before failure is indicated. Default: `0`
retestinterval	The time, in seconds, to wait between each failure retest. Default: `10`
Soap parameters	

Table 142. SOAP monitor configuration (continued)

Element	Description

inputs	Provides access to the name, type, and value fields, including attributes, for SOAP inputs. Use simple, complex, or array soap parameters. For example:
	• Simple: `symbol:string="IBM"`
	• Complex:
	`outer:{item1:string,item2:string}`**`(aaa:string='bbb')`**
	`={item1`**`(attr:string='ccc')`**`='',`
	`item2`**`(attr:string='ddd',attr2:string='eee')`**`='fff'}`
	In this example the attributes in parentheses, marked in bold, are optional.
	• Array: `input:int[]=[1,2,3,4]`

outputs	Provides access to the name, type, and value fields, including attributes, for SOAP outputs. Use simple, complex, or array soap parameters. For more information about syntax, see the examples for SOAP parameter inputs.

Service level classification

Service level classifications define the rules for determining the level of service that is provided by the SOAP interface.

Available service level classification options for the SOAP monitor are:

```
totalTime
message
```

In service level classifications:

- Specify more service level classifications by manually entering the name of the monitor element. The name must match the name that is shown for the element in the Monitor elements section.
- `message` can be any message in the **`$message`** element to IBM Application Performance Management server if used in any widget. For a list of possible values, see Status messages.

Monitor elements

In addition to the test results common to all elements, the SOAP monitor generates a set of test results that contain data specific to SOAP service tests.

The following table lists the additional elements for the SOAP monitor.

Table 143. SOAP monitor elements

Element	Description
(Location)	The URL of the SOAP service that is monitored.
(Operation)	The name of the SOAP service that is monitored.
$outputMatch	`Success` if returned value matches the output value, else `Failure`.
$responseValueName	The value name received in the SOAP response.
$soapname	The container name in the SOAP response. Only applicable to array and user-defined complex data types.
$soaptype	The container type in the SOAP response. Only applicable to array and user-defined complex data types.

Table 143. SOAP monitor elements (continued)

Element	Description
(WSDL)	The path to a local copy of the WSDL file.

Status messages

The SOAP monitor provides status messages in the **ResultMessage** attribute when using IBM Application Performance Management. These messages indicate the result of the test.

The messages are either `Success` if the returned values match the output values or an error message. The error message contains a description of the error.

Example

Monitor the availability of the SOAP interface at 5-minute intervals. If the SOAP interface is unavailable, repeat the test at most two times, with 5 seconds between each repeated test. Send a request that adds 1 + 2 and check that the response contains the value of 3.

Create a SOAP profile element and set the fields that are shown in the following table.

Table 144. SOAP profile element example

Configuration field	Value
wsdl	`c:\%ISMHOME%\etc\SOAP.wsdl`
operation	add
operationnamespace	`http://localhost/SOAP/Calculator`
location	`http://serverA/SOAP/Calculator`
description	basic Calculator SOAP monitor
Active	Selected
timeout	30
poll	300
failureretests	2
retestinterval	5
inputs	[in0=1,in1=2]
outputs	[addReturn=3]

TCPPort Monitor

The TCPPort monitor provides coverage for services that are not tested by the other monitors. It detects and responds to commands or strings on a TCP port. This monitor is particularly useful for monitoring bespoke services.

The following table lists the TCPPort monitor files.

Table 145. TCPPort monitor files

Monitor files	Name or location
Monitor executable	`nco_m_tcpport`

Table 145. TCPPort monitor files (continued)

Monitor files	Name or location
Properties file	`$ISHOME/etc/props/tcpport.props`
Rules file	`$ISHOME/etc/rules/tcpport.rules`
Log file	`$ISHOME/log/tcpport.log`

Guidelines for configuring the TCPPort monitor

The TCPPort monitor tests TCP-based services by connecting to the service, monitoring messages, received from the service and sending responses to it.

To configure a test, you define a sequence of expected messages and responses that comprise a normal interaction on that service.

For example, a standard interaction for a telnet service involves the following sequence:

- The telnet service sends a login message, prompting for a username.
- The client sends a response that contains a username.
- The telnet service sends a message that prompts a password.
- The client sends a response that contains a password.
- If the login attempt is successful, the telnet service sends some form of greeting message.

The monitor's **`WaitForn`** and **Sendn** properties, which are specified define the expected messages and the responses to those messages. These properties in the monitor properties file, define how the monitor interacts with the TCP service:

- **`WaitForn`** properties are regular expressions. The monitor uses them to match messages that are received on the monitored port.
- **Sendn** properties are literal strings that the monitor writes to the port.

Note: If required, you can insert control characters into these properties by using a text editor that supports control character insertion.

The format for defining `WaitForn` and Sendn properties is:

```
WaitFor1: 1st received message
Send1: 1st response
WaitFor2: 2nd received message
Send2: 2nd response
...
WaitFor5: 5th received message
Send5: 5th response
```

When the monitor reaches the first unset **`WaitFor`** property, it stops sending and receiving. If the **`MonitorDisconnect`** property is set to 0, the monitored service must close the connection that is opened by the monitor otherwise the monitor reports the message `Timed out waiting to read` in its **`$message`** element. With many services, connection can be closed by sending a `quit` command. If **`MonitorDisconnect`** is set to 1, the monitor disconnects after the last Send or `WaitFor` command completes, or the timeout is reached, whichever occurs first.

Configuring the TCPPort monitor service test

Use the TCPPort monitor configuration parameters to define service tests.

Table 146. TCPPort configuration

Field	Description
server	The IP address of the system on which the target service is running. Example is `server.mycompany.com`

Table 146. TCPPort configuration (continued)

Field	Description
port	The port on which to connect to the target service.
description	A text field for providing descriptive information on the element.
timeout	The time, in seconds, to wait for the server to respond. Default: `30`
poll	The time, in seconds, between each poll. Default: `300`
failureretests	The number of times to retest before failure is indicated. Default: `0`
retestinterval	The time, in seconds, to wait between each failure retest. Default: `10`

Note: Monitor the availability of the `telnet` service that runs on the host `server.mycompany.com` on port 23. Use the credentials `user` or `guest` to log in to the server and close the connection immediately after login. Run the test at 5-minute intervals, and set a 10-second timeout on connection attempts.

1.Add the following entries to the TCPPort properties file:

```
WaitFor1: ".*[Ll]ogin:"
Send1: "user"
WaitFor2: ".*[Pp]assword:"
Send2: "guest"
WaitFor3: ".*%"
Send3: "exit"
```

2.Start or Restart the TCPPort monitor.

Regular expression matches

Perform a regular expression search on the information that is downloaded by entering up to 50 different regular expressions. The TCPPort monitor attempts to match the contents that are retrieved to each of the regular expressions. If a match for a specified regular expression is found, the matched lines (or as much as can fit in the monitor's internal buffer) are returned in the corresponding `$regexpMatchn` element. If the regular expression matches more than once in the information downloaded, only the first match is returned. The status of each regular expression test is indicated by the `$regexpStatusn` elements. You can use the regular expression matches and their status information as criteria for service level classifications.

For more information, see Table 50 on page 322.

Monitor elements

The following table describes the additional elements for the TCPPort monitor.

Elements indicated by an asterisk (*) are available as attributes. The names of the attributes are shown within brackets. Absence of an asterisk indicates there's no equivalent attribute. Attributes that are shown in bracket but without an element indicates that they are only available as attributes, there's no equivalent element.

In addition to the test results common to all elements, the TCPPort monitor generates a set of test results that contain data specific to TCPPort service tests.

Table 147. TCPPort monitor elements

Element	Description
`$bytesPerSec`	The average number of bytes transferred each second.
`$bytesTransferred`	The number of bytes uploaded or downloaded.
`$connectTime*(ConnectTime)`	The time taken to establish a connection with the target server.
`$downloadTime*(DownloadTime)`	The time taken to download data.
`$lastlineThere's`	The contents of the last line received from the target server.
`$lookupTime*(LookupTime)`	The time taken to obtain the IP address of the host server.
`$networkError`	Contains any network errors during the connection.

`$port*`
`(Port)`
	The port on the target server the monitor tried to connect to.

`$waitingFor`	If the connection terminates before the monitor completes its sequence of waits and sends, this element contains the contents of the last `WaitFor` property.

Status message

The TCPPort monitor provides status messages in the `ResultMessage` attribute when using IBM Application Performance Management. These messages indicate the result of the test.

The following table describes the TCPPort status messages.

Table 148. TCPPort monitor status messages

Message	Description
`OK`	The request succeeded.
`Timed out waiting to read/write`	A data connection to the server was established, but it does not respond.
`Connection closed unexpectedly`	The connection to the server was broken.
`Connection failed`	The monitor failed to connect to the server. For more information, see the log file.
`Network connect error`	There's a problem with the network.
`Network error whilst reading`	

Properties

Properties options specific to the TCPPort monitor are described in the following table.

Table 149. TCPPort properties

Property name	Property parameter	Description
Monitor Disconnect	0|1	Specifies that the monitor must disconnect itself after the last Send or WaitFor command. If the last command is a Send, the monitor disconnects immediately after the string is sent. If the last command is a WaitFor, the monitor disconnects as soon as the monitor receives a match or when the poll timeout is exceeded. 0 - disabled (monitor doesn't connect) 1 - enabled
OutputDirectory	string	Specifies the output directory to use if the OutputResult is saved. Default: $ISHOME/var.
OutputResult	0|1	Specifies that the monitor must save the data that it receives from the service. 0 - disabled 1 - enabled
Send	*n*	Literal string that the monitor writes to the port. See Guidelines for configuring TCPPort monitor. n is a number in the range 1 - 30 inclusive.
singleLineMatch	0|1	Specifies that the monitor should return a single line match when a regular expression is matched. 0 - disabled (multiple lines are matched) 1 - enabled (single line is matched)
WaitFor	*n*	Regular expression used to match commands or strings on the monitored port. For more information, see Guidelines for configuring TCPPort monitor. n is a number 1 - 30 inclusive.

Cipher suites

The SSLCipherSuite property specifies the cipher suite that is used by the TCPPORT monitor. For more information about SSL settings, see "SSL setting in Internet Service Monitoring" on page 428.

TFTP Monitor

The TFTP monitor measures the performance of the Trivial File Transfer Protocol (TFTP) service between two systems.

The following table lists the TFTP monitor files.

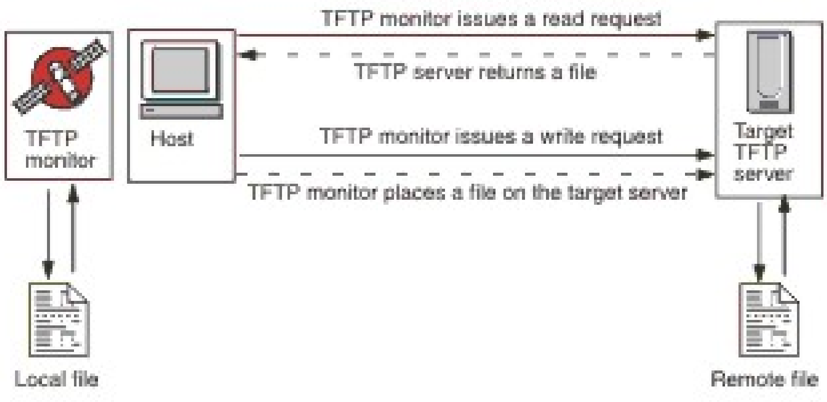
Table 150. FTP monitor summary

Monitor files	Name or location
Executable name	nco_m_tftp

Table 150. FTP monitor summary (continued)	
Monitor files	**Name or location**
Properties file	`$ISHOME/etc/props/tftp.props`
Rules file	`$ISHOME/etc/rules/tftp.rules`
Log file	`$ISHOME/log/tftp.log`

Guidelines for configuring TFTP monitor

The TFTP monitor transfers files between the host system and the target server by using TFTP READ or WRITE requests, then records the response time and data transfer rate. Use it to ensure that your TFTP server is up and running and transferring files at an acceptable rate.


TFTP monitor issues a read request
TFTP server returns a file
TFTP monitor issues a write request
TFTP monitor places a file on the target server
TFTP monitor
Host
Target TFTP server
Local file
Remote file

To upload a file, the monitor sends the TFTP WRITE request (WRQ), and to download a file it sends the TFTP READ request (RRQ). In the TFTP clients, the upload operation is PUT and download operation is GET.

The TFTP monitor supports the octet (binary) and netascii file transfer modes.

Configuring TFTP monitor services tests

Use the TFTP monitor configuration parameters to define service tests.

Table 151. TFTP monitor configuration	
Field	**Description**
server	The IP address of the target TFTP server, or the system you want to transfer files to or from.
localfile	For GET operations, this field specifies the name and path to which the file is downloaded. For PUT operations, this field specifies the name and path of the file that is uploaded to the server.
remotefile	For GET operations, this field specifies the name and path of the file that is downloaded from the server. For PUT operations, this field specifies the name and path to which the file is uploaded on the server.
description	A text field for providing descriptive information about the TFTP monitor.

Table 151. TFTP monitor configuration (continued)

Field	Description
port	The port that the TFTP server uses. Default: 69
localip	The IP address of the host's network interface on which the monitor opens the TFTP connection. If this field is empty, the monitor uses the interface that is specified by the `IpAddress` property
localport	The port that the monitor uses to establish the TFTP connection. If the value of this field is `0`, the monitor selects a suitable port.
command	The TFTP command for the monitor to use: • `GET` - Download a file from the target server to the monitor host. • `PUT` - Upload a file from the monitor host to the target server. Default: `GET`
transfermode	Specifies the format in which the monitor transfers the file: • `OCTET` (8-bit) • `NETASCII` Default: `OCTET`
timeout	The time, in seconds, to wait for the TFTP server to respond. Default: `10`
retries	The number of times that the monitor attempts to transfer a file before it quits. Default: 3
poll	The time, in seconds, between each poll. Do not set this value too low as constant polling might overwhelm the service. Default: `300`
failureretests	The number of times that the monitor retests the TFTP server after an initial failure before failure is indicated. Default: `0`
retestinterval	The time, in seconds, to wait between each failure retest. Default: `10`

Service level classifications

Service level classifications define the rules for determining the level of service that is provided by a TFTP server.

Available service level classification options for the TFTP monitor are:

```
totalTime
lookupTime
responseTime
```

```
transferTime
bytesTransferred
bytesPerSec
checksum
message
```

In service level classifications:

- Specify more service level classifications by manually entering the name of the monitor element. The name must match the name that is shown for the element in the Monitor elements section.
- `message` can be any message that is forwarded in the `$message` element to IBM Application Performance Management server if used in any widget. For a list of possible values, see Status messages.
- The operand is a string or a positive number.
- The `checksum` element does not normally provide meaningful results for service level classifications. Its value is not known when the profile element is created. The monitor calculates checksum values while tests are in progress. This element is intended for alert enrichment by using rules files.

Monitor elements

In addition to the test results common to all elements, the TFTP monitor generates a set of test results that contain data specific to TFTP service tests.

The following table describes the additional elements for the TFTP monitor.

Table 152. TFTP monitor elements

Element	Description
`$bytesPerSec*` (`BytesPerSec`)	The average number of bytes transferred each second.
`$bytesTransferred*` (`BytesTransferred`)	The number of bytes uploaded or downloaded.
`$checksum`	The checksum value of the downloaded data. It is generated by the monitor and is provided for further processing by using rules files.

`$command*`
`(TftpCommand)`
	The TFTP command that is issued by the monitor (GET or PUT).

`$localFile*` (`TftpLocalFile`)	Full path name of the file stored on the local host. This element is taken from the configuration file.
`$localIP`	The local IP address the monitor is configured to use. It might be blank on a system with only one interface.
`$lookupTime*` (`LookupTime`)	The time taken to obtain the IP address of the host server.
`$remoteFile*` (`TftpRemoteFile`)	Full path name of the file stored on the remote host (the FTP server). This element is taken from the configuration file.
(`TransferTime`)	The time taken to transfer the file.

Table 152. TFTP monitor elements (continued)

Element	Description
(TftpConnection)	The format in which the monitor transferred the file. It is either OCTET (8 bit) or NETASCII.

Status messages

The TFTP monitor provides status messages in the ResultMessage attribute when using IBM Application Performance Management. These messages indicate the result of the test.

Table 153. TFTP monitor status messages

Message	Description
OK	The TFTP request succeeded.
FAILED: connect failed	The monitor failed to connect to the server. Check that the server is running.
FAILED: internal tftp monitor error	There is a problem with the monitor, possibly caused by insufficient memory.
FAILED: A send/wait timed out	The TFTP request failed. There might be a problem with the network.
FAILED: An unspecific error condition. The transfer should be aborted	
FAILED: Received a short or malformed packet	
FAILED: local file open/read/write failed	
FAILED: unrecognized status from transfer attempt	

Example

Test the availability of the TFTP server tftp.mycompany.com by uploading the file $ISHOME/etc/testfiles/upload.txt to /ism/test/upload_result.txt. Use netascii mode to upload the file at 20-minute intervals

Classify the service level by using the following criteria:

- If the upload is not successful, the service level is Failed
- If the total time of the transfer is greater than 10 seconds, the service level is Marginal
- Otherwise, the service level is Good

Create a TFTP monitor profile element and set the configuration as shown in the following table.

Table 154. TFTP profile element example

Configuration field	Values
server	tftp.mycompany.com
localfile	$ISHOME/etc/ism/testfiles/upload.txt

Table 154. TFTP profile element example (continued)

Configuration field	Values
remotefile	`/ism/test/upload_result.txt`
description	TFTP test
Active	Selected
command	PUT
transfermode	NETASCII
poll	1200

statement	`If (Message != OK) then status Failed`
	`else if (TotalTime > 10) then status`
	`Marginal`
	`else status Good`

TRANSX Monitor

The TRANSX monitor simulates the actions of a real internet user by running a series of activities, which it performs by using other Internet Service Monitors.

For example, configure TRANSX to access pages of a website by using the HTTP monitor, download some files, send or receive by using the POP3 and SMTP monitors.

The following table lists the TRANSX monitor files.

Table 155. TRANSX monitor files

Monitor files	Name or location
Monitor executable	`nco_m_transaction`
Properties file	`$ISHOME/etc/props/transx.props`
Rules file	`$ISHOME/etc/rules/transx.rules`
Log file	`$ISHOME/log/transx.log`

Properties

Table 156. TRANSX monitor properties

Property name	Property parameter	Description
`CompleteTransax`	`0|1`	Specifies that the transaction continues even if a step fails. • `0` - disabled (doesn't continue) • `1` - enabled (continues)
`DetailedTimings`	`0|1`	Specifies that the TRANSX monitor produces data logs that contains fine-grained timings for each step. Fine grained timings that are produced in the data logs are pre-set, and can't be modified. • `0` - disabled • `1` - enabled

Table 156. TRANSX monitor properties (continued)

Property name	Property parameter	Description
`MultipleEvents`	<u>0</u>|1	Specifies whether the monitor generates multiple events for transaction results: • `0` - disabled (monitor generates only one event containing the results of all steps and summary results) • `1` - enabled (monitor generates one event for every step in the transaction and a final summary event)
`StepPause`	integer	Specifies the length of the pause, in seconds, between execution of each transaction step. The length of the pause doesn’t affect the value of a transaction's `$totalTime` element. `$totalTime` represents the sum of a transaction's `$stepXTime` elements. Default: `0`

The following table describes the properties specific to the SMTP monitor.

Guidelines for configuring the TRANSX monitor

The TRANSX monitor tests services by simulating a set of activities that comprise a typical user experience. The set of activities is called a `transaction`, and each activity in the transaction is called a `transaction step`.

TRANSX profile elements define the transactions. Each transaction step configures an Internet service monitor, such as HTTP, to perform the operation for that step. You configure transaction steps through the Edit button on the Steps tab of the TRANSX profile element.

The steps are configured in the same way as you configure any other profile elements. For example, the configuration details for a step that involves the HTTP monitor can include Head/Form parameters, Proxy server parameters, regular expressions, and service level classifications.

When the TRANSX monitor tests a step in the transaction, it records the time that is taken and the level of service for the step.

Note: The TRANSX monitor requires root privileges if any transaction steps use another monitor, such as ICMP, that requires root privileges.

Handling dynamic content with HTTP and HTTPS monitors

Many websites use the dynamic content to provide functions such as session- or region-based interactions. When used together with the TRANSX monitor, the HTTP and HTTPS monitors can test web pages that contain dynamic content, such as session IDs, region codes, or dates and times that are embedded in links, whose values may be different each time the transaction is tested.

The dynamic content features provided by the HTTP and HTTPS monitors when running in transaction mode enables to identify dynamic content in the form of name-value pairs, called `dynamic page elements`, embedded in URLs, or defined in HTML `form` elements, which the monitor then extracts from a page during each test, ensuring that the appropriate dynamic value is used every time that a transaction is tested.

For example, consider a website's Login page, `http://www.mycompany.com/login`, that contains a link for logging on to the website. The link URL for the login action `http://www.mycompany.com/doLogin?sessionID=id` include a session ID for the Login transaction. In this example, the name-value pair `sessionID=id` is a dynamic page element; the value of `id` changes each time the Login page is accessed. To test the Login page as part of a transaction, you would configure the transaction to obtain and use the value of `sessionID` each time the transaction is tested and insert it into the login action URL.

The TRANSX monitor passes dynamic page elements from one transaction step to the next. In the website Login example, the transaction's first step would access the Login page to obtain the session ID then pass it to a second step, which submits the login request containing that ID. The operations that are performed in these steps are:

1. Access the page containing the dynamic page elements, for example `http://www.mycompany.com/login`
2. Submit the action by using the dynamic page elements, for example `http://www.mycompany.com/doLogin?sessionID=@@030671`

When you identify a dynamic page element in a transaction step, it's passed on to the next transaction step, which inserts the name-value pair into its request. The element is passed on to each subsequent HTTP or HTTPS transaction until you explicitly remove it.

Adding and removing dynamic pages

Each HTTP or HTTPS transaction step consists of a request, which returns an HTML page. When running a transaction step that contains dynamic page elements, the monitor parses the HTML page to locate each element’s name-value pair and passes it to the next transaction step, which inserts them into its HTTP or HTTPS request.

To specify that a step uses dynamic page elements, set the parameter type for the element to DYNAMIC. Then, specify each dynamic element that is to be extracted and passed on to later steps. Identify each dynamic element by entering its name, for example, `sessionID` and select Add To as the value. Add To indicates that the element is to be passed on to later steps.

Note: To obtain the name of a dynamic page element, view the HTML source of the page on which it’s located.

Dynamic page elements are passed from one transaction step to the next. If a page element is no longer required to be passed on to a next step, set the Value of the element to Remove From. Manually update the subsequent transaction steps to ensure that the correct page elements are processed.

Use the following guidelines for adding and removing dynamic page elements:

- If a step doesn’t use any dynamic page elements, don’t select DYNAMIC as the parameter type.
- If a step requires a dynamic page element, the step that retrieves the page on which the dynamic element appears must specify the name of the element and the value Add To.
- If a step doesn’t require a dynamic element that is passed on from prior steps, set the value of the previous step to Remove From.

GET and POST

In GET methods, all dynamic page elements are inserted into the request URL automatically. In POST methods, you must specify each dynamic element as a FORM parameter on the Parameters tab.

The monitor automatically inserts the dynamic value for each form when it runs the transaction step.

Creating transaction

You define transactions by creating TRANSX profile elements and transaction steps by using the Internet Service Monitoring user interface. For more information, see “Creating transactions” on page 428.

Configuring the TANSX monitor service test

Table 157. TRANSX monitor configuration

Field	Description
transxname	A name for the transaction.
description	A text field for providing descriptive information on the element.
poll	The time, in seconds, between each poll. Default: `300`

Note: Monitor the availability of a website by using a sequence of web browsing, file downloads, and sent email messages.

1. Create a TRANSX profile element.
2. Create an HTTP transaction step to monitor the availability of a website.
3. Create an FTP transaction step to monitor a file download.
4. Create a POP3 or SMTP transaction step to monitor email.

See to the documentation for each monitor for further information.

Monitor elements

The TRANSX monitor generates events containing the results of each transaction. These events contain the results of the entire transaction as well as those of the individual transaction steps.

However, by default, the monitor places all transaction and step results in a single event, by using the `MultipleEvents` property that you can configure the monitor to create individual events for each transaction step and a summary event for the entire transaction. Table 1 lists the TRANSX summary elements.

Elements indicated by an asterisk (*) are available as attributes. The names of the attributes are shown within brackets. Absence of an asterisk indicates there's no equivalent attribute. Attributes that are shown in bracket but without an element indicates that they are only available as attributes, there's no equivalent element.

Table 158. TRANSX summary monitor elements

Element	Description
`$numberOfSteps*(NumberOfSteps)`	The number of steps in the transaction.
`$stepDescriptions`	A list of the descriptions for each step, which is separated by a vertical bar character (|).
`$stepTimes*(Step`*`1 to 10`*`TotalTime)`	The timing data returned by each step (1 - 10).
`$stepUnits`	A list of the units for each step, usually seconds, each separated by a vertical bar character (|).
`(TransName)`	The name of the transaction as specified when configuring the transaction.
`(TransStepDescription)`	The description of the transaction step as specified when configuring the step.

Status message

The TRANSX monitor provides status messages in the `ResultMessage` attribute when using IBM Application Performance Management. These messages indicate the result of the test.

The following table describes the status messages.

Table 159. TRANSX monitor status messages

Message	Description
`Successfully completed transaction`	The transaction was completed successfully.
`Error in transaction`	There was a failure in one of the steps of the transaction.
`Service Level Failed, ending transactionService Level Failed`	The service level of one of the steps returned a failed response, which caused the transaction to stop.

Creating transactions
Transactions can be defined by creating TRANSX profile elements and transaction steps using the Internet Service Monitoring agent user interface.

Procedure

To create a transaction using the interface:

1. Click **System Configuration icon**. Under which click **Agent Configuration**.

   The agent configuration window opens.
2. Click **ISM** tab to configure Internet Service Monitoring agent.
3. Click **(+)** plus icon to create a new profile. Enter the **Profile Name** and **Description**.
4. Click **Next**.
5. Click **TRANSX** Monitor from monitor drop-down list to select TRANSX Monitor.
6. Click **Next**.
7. Enter the mandatory parameters.
8. In the **advance** tab specify the **poll interval**.
9. Click **(+)** plus icon on the Steps tab.
10. Click the monitor that needs to be selected from the monitor drop-down list.
11. Click **Select** to configure the transaction step.
    a) Specify the mandatory and optional parameters the same way as previously entered to configure the profile elements.
    b) If creating dynamic steps for the HTTP or HTTPS monitor, set the Name and Value pairs on the Parameters tab and select DYNAMIC as the parameter type.
12. Click **Add**.
13. Click refresh icon on the steps grid.
14. Repeat steps 1 - 13 for each additional transaction help.
15. Click **Add** to finish.
16. Click **Done** to save.

Results

Note: To specify a pause between each transaction step, use the `StepPause` property.

SSL setting in Internet Service Monitoring
Internet Service Monitoring uses OpenSSL to communicate securely with typically remote internet services using various monitors, for example, the HTTPS monitor communicates with a secured HTTPD.

Internet Service Monitoring agent also uses OpenSSL between the monitors and the Databridge and between the Internet Service Monitoring agent (KIS) and the Databridge. Specify the cipher suite that your application uses in the `SSLCipherSuite` property.

The Databridge should be configured to securely communicate with the monitors and the Internet Service Monitoring agent so that every monitor shares a common set of Databridge related properties to manage secure communication with the Databridge. Some monitors also share a similar, but different set of related properties to manage secure communication with their respective internet services under test.

The following monitors support monitoring of secured internet services:

- HTTPS
- IMAP4
- POP3
- SMTP

These monitors use certificates. All certificates are stored in X509 format in Privacy Enhanced Mail .Pemfiles in `$ISMHOME/certificates`. The certificate for the Databridge is also stored in the same location. For this reason, the following properties are shared by all monitors, the Databridge, and the Internet Service Monitoring agent:

- SSLTrustStore (Default: $ISMHOME/certificates/trust.pem)
- SSLTrustStorePath (Default: $ISMHOME/certificates/)

As all communication between monitors and the Databridge, and between selected monitors and their secured internet services are built on the same version of OpenSSL, they share characteristics. For example, the highest level of security Internet Service Monitoring can provide is a function of the highest level provided by the underlying OpenSSL. The lowest level of security provided is similarly dependent on the underlying OpenSSL.

If the Internet Service Monitoring agent is updated, and that update includes an update to the underlying OpenSSL, the internet services being monitored might be impacted. For example:

1. HTTPS monitor in Internet Service Monitoring agent V7.x.1 is monitoring a secured HTTPD server.
2. Apply a new version of Internet Service Monitoring agent that contains an updated version of OpenSSL, which means the HTTPS monitor is now V7.x.2.
3. You notice that the HTTPS monitor is now failing to monitor the secured HTTPD.

The security level of the HTTPD server is less than the minimum supported by the newly updated Internet Service Monitoring agent V7.x.2. Even though the configuration of the HTTPS monitor hasn't changed, its behavior has, because it depends on the underlying OpenSSL layer. The newer Internet Service Monitoring agent/HTTPS Monitor/OpenSSL combination is more secure than the old combination, and you now you need to raise the security level of the remote HTTPD server.

Monitoring secured internet services presents you with a dilemma. Should the security level of Internet Service Monitoring agent be so low that it can monitor weakly protected Internet services; or should it be as high as the minimum currently recommended settings? If the former is selected, then a weakened Internet Service Monitoring agent might compromise security, possibly at both ends.

The same version of OpenSSL is used by all monitors. All of these monitors share a common set of monitor properties for configuring the underlying OpenSSL, which are described in the following table.

Table 160. OpenSSL related monitor properties

Property name	Property parameter	Description
`SSLCipherSuite`	string	Specifies the cipher suites to use for SSL operations between the monitor and the internet service being monitored. Values for this property should be in the form recommended by OpenSSL. Default: `AES:3DES:DES:!EXP:!DHE:!EDH`
`SSLDisableSSLv2`	0|1	Determines which type of secure connection to make when monitoring a secured internet service. `0` – SSLv2 is allowed `1` – SSLv2 is NOT allowed Default: `1` (SSLv2 NOT allowed).
`SSLDisableSSLv3`	0|1	Determines which type of secure connection to make when monitoring a secured internet service. `0` – SSLv3 is allowed `1` – SSLv3 is NOT allowed Default: `1` (SSLv3 NOT allowed).
`SSLDisableTLS`	0|1	Determines which type of secure connection to make when monitoring a secured internet service. `0` – TLSv1.0 is allowed `1` – TLSv1.0 is NOT allowed Default: `0` (TLSv1.0 is allowed).
`SSLDisableTLS11`	0|1	Determines which type of secure connection to make when monitoring a secured internet service. `0` – TLSv1.1 is allowed `1` – TLSv1.1 is NOT allowed Default: 0 (TLSv1.1 is allowed).
`SSLDisableTLS12`	0|1	Determines which type of secure connection to make when monitoring a secured internet service. `0` – TLSv1.2 is allowed `1` – TLSv1.2 is NOT allowed Default: 0 (TLSv1.2 is allowed).

Table 160. OpenSSL related monitor properties (continued)

Property name	Property parameter	Description
`SSLCertificateFile`	string	The path and filename of the public digital certificate file used by the monitor. When a monitor attempts to set up a secured connection to an internet service, the latter may optionally request that the monitor provide its client side certificate, allowing the internet service to verify the monitor or client (client-side certificate verification). The certificate must be in Privacy Enhanced Mail (PEM) format. For the HTTPS monitor, this value can be specified for each HTTPS element at creation time. However, if the HTTPS monitor is going to use the same certificate for all elements, the value in the `HTTPS.props` file is used. For IMAP, LDAP, POP3, SIP, SMTP and SOAP monitors, the value is set monitor wide. If the path isn't absolute, the monitor interprets it relative to the working directory, `$ISMHOME/certificates`. Default: “”
`SSLKeyFile`	string	The path and filename of the file containing the private key used by the monitor. The monitor uses this file to encrypt messages it sends to others. The receivers use the monitor's public digital certificate to decrypt the message. Default: `monitoryKey.pem`
`SSLKeyPassword`	string	The password used to encrypt the SSL private key. Default: “”
`SSLTrustStoreFile`	string	The fully qualified name of the file that stores all the X509 public certificates of the internet services that are being monitored, as a concatenated list. Revoked certificates (CRLs) are also stored here as a concatenated list. The Databridge can also store its public certificate here. This property appears in the `bridge.props` file. Certificates are stored in Privacy Enhanced Mail (PEM) format. Convert certificates obtained in other formats to PEM format using OpenSSL software available from http://www.openssl.org. Default: **“`$ISMHOME/certificates/trust.pem`”**

Table 160. OpenSSL related monitor properties (continued)

Property name	Property parameter	Description
`SSLTrustStorePath`	string	The location of the `.pem` files containing the X509 certificates of the secure internet service being monitored. Revoked certificates (CRLs) are also stored here. The Databridge can also store its public certificate here. This property appears in the`bridge.props` file. If new certificates are added to this directory, run the openssl `rehash` command to scan the directory and calculate a hash for each certificate. If both `SSLTrustStoreFile` and `SSLTrustStorePath` properties are used, OpenSSL uses both properties to locate trusted certificates. Default: "`$ISMHOME/certificates/`"
`VerifyCertificate Preference`	0|1	Enables or Disables the verification of the certificate provided by the internet service being monitored against the certificate revocation list (CRL). Default: 0 - disabled

Cipher suites

The cipher suites available to the Internet Service Monitoring are a subset of those allowed by OpenSSL. The set of cipher suites allowed by OpenSSL changes over time. As new vulnerabilities are discovered and best practices evolve, access to specific or general types of cipher suites may be restricted or removed entirely by OpenSSL. As these later versions of OpenSSL are included in later versions of the ISM, there's a flow on effect, which may impact the configuration and operation of the monitors.

Use the SSLCipherSuite monitor-wide property to specify the cipher suites allowed by a monitor from all the ciphers suites available using keywords. To specify multiple suites, use a colon separated list of keywords. For example, the default SSLCipherSuite property is `AES:3DES:DES:!EXP:!DHE:!EDH`. This selection means that cipher suites that include AES, 3DES, and DES are allowed, but excludes any cipher suites that use EXP (Export (short key lengths)), DHE (Diffie Hellman Exchange), or EDH (Ephemeral Diffie Hellman) key exchanges. Additionally, when the secure connection is made between the monitor and the internet service, AES is used first, followed by 3DES, then DES if necessary. The syntax for the cipher suite lists for Internet Service Monitoring agent is the same as for OpenSSL.

To pick the correct set of cipher suites for a monitor, consider what the underlying OpenSSL supports, the range of ciphers that the internet service being monitored supports, and the security standards of your organization. You may not be able to monitor a secure external site that has a level of security less than that which Internet Service Monitoring or OpenSSL will tolerate. In some cases, a monitor that was once able to monitor an internet service, may fail after upgrading Internet Service Monitoring because the security levels are incompatible.

The following table lists a subset of cipher suites equivalent to the default value for SSLCiperSuite of `AES:3DES:DES:!EXP:!DHE:!EDH` with their properties. In the table, you'll see the following terms:

- Cipher Suite Name: Describes the cipher suite using a name constructed from keywords.
- Protocol: Describes the version of the protocol supported.
- Key Exchange: Describes the key exchange system used for encryption and decryption.

- Encryption & Key Length: Describes the type of encryption algorithm used and the length of the key (in bits) used.
- MAC: Describes the Message Authentication Code used to ensure that the data hasn't been tampered with.

Table 161. Cipher suite name and property values AES:3DES:DES:!EXP:!DHE:!EDH

Cipher Suite Name	Protocol	Key Exchange	Authentication	Encryption & Key Length	Message Authentication Code
ECDHE-RSA-AES256-GCM-SHA384	TLSv1.2	ECDH	RSA	AESGCM(256)	AEAD
ECDHE-ECDSA-AES256-GCM-SHA384	TLSv1.2	ECDH	ECDSA	AESGCM(256)	AEAD
ECDHE-RSA-AES256-SHA384	TLSv1.2	ECDH	RSA	AES(256)	SHA384
ECDHE-ECDSA-AES256-SHA384	TLSv1.2	ECDH	ECDSA	AES(256)	SHA384
ECDHE-RSA-AES256-SHA	SSLv3	ECDH	RSA	AES(256)	SHA1
ECDHE-ECDSA-AES256-SHA	SSLv3	ECDH	ECDSA	AES(256)	SHA1
SRP-DSS-AES-256-CBC-SHA	SSLv3	SRP	DSS	AES(256)	SHA1
SRP-RSA-AES-256-CBC-SHA	SSLv3	SRP	RSA	AES(256)	SHA1
SRP-AES-256-CBC-SHA	SSLv3	SRP	SRP	AES(256)	SHA1
DH-DSS-AES256-GCM-SHA384	TLSv1.2	DH/DSS	DH	AESGCM(256)	AEAD
...followed by 61 more rows					

The following table lists a subset of cipher suites equivalent to the value for `SSLCiperSuite` of `AES:3DES:DES:!EXP:!DHE:!EDH:!SSLv2:!SSLv3` with their properties. Some protocols are now eliminated and the overall set of cipher suites has been reduced 71 - 31.

Table 162. Cipher suite name and property values ***AES:3DES:DES:!EXP:!DHE:!EDH:!SSLv2:!SSLv3***

Cipher Suite Name	Protocol	Key Exchange	Authentication	Encryption & Key Length	Message Authentication Code
ECDHE-ECDSA-AES256-GCM-SHA384	TLSv1.2	ECDH	ECDSA	AESGCM(256)	AEAD
ECDHE-RSA-AES256-SHA384	TLSv1.2	ECDH	RSA	AES(256)	SHA384
ECDHE-ECDSA-AES256-SHA384	TLSv1.2	ECDH	ECDSA	AES(256)	SHA384
DH-DSS-AES256-GCM-SHA384	TLSv1.2	DH/DSS	DH	AESGCM(256)	AEAD

*Table 162. Cipher suite name and property values **`AES:3DES:DES:!EXP:!DHE:!EDH:!SSLv2:!SSLv3`** (continued)*

Cipher Suite Name	Protocol	Key Exchange	Authentication	Encryption & Key Length	Message Authentication Code
DH-RSA-AES256-GCM-SHA384	TLSv1.2	DH/RSA	DH	AESGCM(256)	AEAD
DH-RSA-AES256-SHA256	TLSv1.2	DH/RSA	DH	AES(256)	SHA256
DH-DSS-AES256-SHA256	TLSv1.2	DH/DSS	DH	AES(256)	SHA256
ADH-AES256-GCM-SHA384	TLSv1.2	DH	None	AESGCM(256)	AEAD
ADH-AES256-SHA256	TLSv1.2	DH	None	AES(256)	SHA256
ECDH-RSA-AES256-GCM-SHA384	TLSv1.2	ECDH/RSA	ECDH	AESGCM(256)	AEAD
...followed by 21 more rows					

Reducing vulnerability

In future releases, the DHE and EDH ciphers will be disabled by default because of vulnerabilities. For previous versions of Internet Service Monitoring agent, you may need to disable the DHE and EDH ciphers in all monitors. To disable the DHE and EDH ciphers, update the `SSLCipherSuite` and `BridgeSSLCipherSet` monitor properties.

For example, to disable DHE and EDH ciphers in the HTTPS monitor, update the `https.props` file to include the following properties:

```
SSLCipherSuite: AES:3DES:DES:!DES-CBC-SHA:!EXP:!DHE:!EDH
BridgeSSLCipherSet: AES:3DES:DES:!DES-CBC-SHA:!EXP:!DHE:!ED
```

Ensure that you verify that this configuration change doesn't cause any compatibility issues. If you change the default setting after applying this fix, you may expose yourself to a security vulnerability. You should review your entire environment to identify other areas where you have enabled the Diffie-Hellman key-exchange protocol used in TLS and take appropriate mitigation and remediation actions.

Protocol selection

You can select from a range of historic and current secure communication protocols. They can be individually selected using a set of boolean monitor properties:

- SSLDisableSSLv2
- SSLDisableSSLv3
- SSLDisableTLS
- SSLDisableTLS11
- SSLDisableTLS12
- BridgeSSLDisableSSLv2
- BridgeSSLDisableSSLv3

You should disable SSLv2 and SSLv3. These protocols have been compromised and have several known vulnerabilities. They are disabled by default and are only provided for legacy purposes.

Internet Service Monitoring enables TLS by default. If you know that the internet services you're monitoring aren't using TLS 1.0 and have already uplifted to TLS 1.1 or TLS 1.2, you should disable the unused protocols in Internet Service Monitoring.

The Databridge component communicates with the Internet Service Monitoring agent and with each of the monitors. By default, this communication is encrypted and TLS is the preferred protocol.

Key trust stores and certificates

Internet Service Monitoring stores its certificates in a user-defined file in a user-defined location. All certificates must be stored in Privacy Enhanced Mail (PEM) format. Ensure that public certificates obtained from other organizations are converted to PEM format. Conversion software is available at http://www.openssl.org.

Trusted certificates specified using the `SSLTrustStoreFile` property are stored in the file as a concatenated list.

It is good practice to store Certificate Revocation Lists (CRLs) in the trust store, against which certificates can be validated. Certificate Authorities have systems in place to generate lists of revoked certificates and have distribution systems in place to make them publicly available. Then, if a certificate is compromised, it will be revoked.

Databridge security settings

All monitors communicate with the Databridge, so all monitors have a common set of properties that should be set to manage communication between the monitors and the Databridge. By default, communication is encrypted. The default encryption protocol is TLS. Unlike monitor properties, there is no mechanism to control if a particular version of TLS is enabled or disabled. All the monitors should have the same values for the Databridge properties, otherwise there will be communication issues. Similarly, the properties set in the Databridge .props file should be consistent with those in the monitors. The Databridge also communicates with the Internet Service Monitoring agent, which has its own .props file. Some of the values in the agent .props are Databridge-related and like monitors, must have values that are consistent with those in the Databridge `.props` file.

Table 163. OpenSSL related Databridge properties

Property name	Property parameter	Description
`BridgeSSLEncryption`	0|1	Determines whether communication with the Databridge is encrypted or not. This covers all communication from Databridge to Monitors and Internet Service Monitoring agent. `0` – not encrypted `1` – encrypted **Restriction:** Set the same value on the Internet Service Monitoring agent, all monitors, and the Databridge.
`BridgeSSLCipherSet`	string	Specifies the cipher suites to use for SSL operations to and from the Databridge. Values for this property should be in the form recommended by OpenSSL. **Restriction:** Set the same value on the Internet Service Monitoring agent, agent, all monitors, and the Databridge. Default: `AES:3DES:DES:!EXP:!DHE:!EDH`

Table 163. OpenSSL related Databridge properties (continued)

Property name	Property parameter	Description
BridgeSSLDisableSSLv2	0|1	Determines which type of secure connection to make to and from the Databridge. 0 – SSLv2 and SSLv3 are allowed 1 – SSLv2 is NOT allowed **Restriction:** Set the same value on the Internet Service Monitoring agent, all monitors, and the Databridge. Default: 1 (SSLv2 NOT allowed).
BridgeSSLDisableSSLv3	0|1	Determines which type of secure connection to make to and from the Databridge. 0 – SSLv3 is allowed 1 – SSLv3 is NOT allowed **Restriction:** Set the same value on the Internet Service Monitoring agent, all monitors, and the Databridge. Default: 1 (SSLv3 NOT allowed).
BridgeSSLCertificateFile	string	The path and filename of the digital Databridge SSL certificate. Default: `$ISMHOME/certificates/bridgeCert.pem`
BridgeSSLKeyFile	string	The path and filename of the Databridge SSL private key file. Default: `$ISMHOME/certificates/bridgeKey.pem`
BridgeSSLKeyPassword	string	The password used to encrypt the Databridge SSL private key. Default: **Tivoli**

Table 163. OpenSSL related Databridge properties (continued)

Property name	Property parameter	Description
BridgeSSLTrustStore	string	The path and file name of the Trusted certificate file for authentication. This is only required when using the **BridgeSSLAuthenticatePeer** property. Default: `$ISMHOME/certificates/trust.pem` If you want to configure SSL authentication between a monitor and Databridge, or between the Databridge and the agent, set `BridgeSSLAuthenticatePeer` to `1` and restart the Databridge. This action authenticates the certificates from the server. You can store certificates in both the `SSLTrustStoreFile` and the `SSLTrustStorePath`. Defaults: • `SSLTrustStoreFile`, `$ISMHOME/certificates/trust.pem` • `SSLTrustStorePath`, `$ISMHOME/certificates/` To add new certificates, complete one of the following steps: • Add a certificate to the end of the list in the `SSLTrustStoreFile` text file • Add a new certificate to the `SSLTrustStorePath` directory, and run the OpenSSL **`c_rehash certificate_dir`** command to hash the certificates
SSLTrustStoreFile	string	This property is used by secure monitors and the Databridge. See Table 160 on page 430 for more information.
SSLTrustStorePath	string	This property is used by secure monitors and the Databridge. See Table 160 on page 430 for more information.
BridgeSSLAuthenticatePeer	0|1	Specifies whether the Databridge should cross-authenticate with other Internet Service Monitoring components. `0` – disabled `1` – enabled If a monitor contacts the Databridge, it must authenticate with the Databridge, and the Databridge must authenticate with the monitor. If the Internet Service monitoring agent contacts the Databridge, it must authenticate with the Databridge, and the Databridge must authenticate with the agent. Certificates for the Databridge are stored in the BridgeSSLTrustStore. Default: 0 - disabled

Internet Service Monitoring agent properties

The Internet Service Monitoring agent has its own properties file ,which contains a set of security properties and settings. The agent properties file doesn’t communicate with the monitors, but it does communicate with the Databridge, so the security settings in the agent's `.props` file manage communication between the agent and the Databridge.

Configuring the agent on Windows systems

You can configure the agent on Windows systems by using the **IBM Performance Management** window.

Procedure

Configure Internet Service Monitoring agent Agent on the user system as follows.

To manually configure the Internet Service Monitoring agent on user systems:

1. On the **IBM Performance Management dashboard,** click **System configuration** > **Agent configuration** that is listed under **System Configuration**.
2. Click **ISM** tab to open the Internet Service Monitoring Agent dashboard.

Configuring Databridge

Configuring the Databridge involves setting properties for the Databridge that control its operation such as the connection of the component modules and the internet service monitors.

Operation and configuration

The Databridge and its component modules are configured through properties files.

The properties determine the operation of the Databridge and its component modules that sends test results to IBM Cloud Application Performance Management for reporting on Internet Service Monitoring Agent dashboard.

Configuring the Databridge

The Databridge must be configured to receive data from the Internet Service Monitors and to forward that data to its component modules for further processing.

The following table lists the files associated with the Databridge. The **`Properties file`**, **`Store And Forward file`**, and **`Log file`** are described in more detail in the appropriate sections.

Table 164. Databridge files and their location

Databridge file	Location or name
Executable file	`$ISHOME/platform/arch/bin/nco_m_bridge`
Properties file	`$ISHOME/etc/props/bridge.props`
Store and Forward file	Name and location are specified by properties in the `bridge.props` file. The default name and location is `$ISHOME/var/sm_bridge.saf`
Log file	`$ISHOME/log/bridge.log`
Error log file	`$ISHOME/log/bridge.err`

Store And Forward file

If the Databridge is unable to forward data to Netcool/OMNIbus , it stores all of the data it would normally send in a Store And Forward (SAF) file. When Netcool/OMNIbus becomes available again, it processes all of the events stored in the SAF file.

The QFile and QSize properties in the Databridge properties file determine the name, location and operation of the store and forward processing.

Log file

The Databridge sends daily messages about its operations to a message log file. By default, the name of this file is `$ISHOME/log/bridge.log`. It is updated at 12 midnight. The Databridge properties MsgDailyLog and MsgTimeLog control the operation of message logging.

Starting the Databridge

Starting the Databridge using the Windows Services console.

Procedure

Note: If the ObjectServer module is connected to the Databridge, ensure that its target system is running before starting the Databridge. If any of the Databridge modules fails to initialize correctly, the Databridge won't start.

1. From the Windows desktop, click **Start** > **Administrative Tools** > **Services**.
2. From the list of services, select the service named NCO BRIDGE Internet Service Monitor and click **Start** from the menu.

Connecting modules

The Databridge properties file defines the modules to connect to the Databridge.

About this task

Each Module n `SharedLib` and Module n `PropFile` property pair defines the connection for one module. Modules are loaded in order of definition, starting from Module0.

Procedure

1. To connect individual modules to the Databridge:
   a) In the Databridge properties file, identify the next available Module n `SharedLib` and Module n `PropFile` property pair.
   b) Set Module n `SharedLib` to the name of the module's shared library (its binary implementation).
   c) Set Module n `PropFile` to the full path of the module's properties file.

   In this example, lines 1 and 2 connect the ObjectServer module, lines 3 and 4 connect the Datalog module, lines 5 and 6 connect the IBM Application Performance Management (`pipe`) module. The Datalog module doesn't have a properties file, so the entry for the properties file has the value `""`.
2. To disable a module:
   a) Set the corresponding Module n `SharedLib` property to `"NONE"` and the Module n `PropFile` property to `""`. All other modules that have a value higher than n are also ignored.

Connecting monitors

Internet service monitors get connected to the Databridge over TCP. Each monitor has a set of properties that configures the connection to the Databridge.

About this task

To connect a monitor to the Databridge, set the value of the `BridgePort` property defined in the monitor's properties file to the value of the `SocketPort` property defined in the Databridge properties file. The default value of each monitor's `BridgePort` property and the Databridge's `SocketPort` property is `9510`.

The databridge supports SSL encryption of the test results that it receives from the monitors. To encrypt a monitor's test results, set the values of the `BridgeSSL` properties defined in the monitor's properties file to the values of the `BridgeSSL` properties defined in the Databridge properties file. To encrypt all monitors' test results, all monitors must have the same `BridgeSSL` properties.

Configuring Databridge module

The Databridge directs test results to the Internet Service Monitoring Agent. The monitoring agent converts this data to the required format and distributes it to the IBM Application Performance

Management Server. Configure the Databridge module and the Internet service monitoring agent through their respective properties files.

Configure the operation of the Databridge by modifying the property values defined in the module properties file.

The module properties file is named `pipe_module.props`. This file is located in the `$ISHOME/etc/props/` directory.

Following table lists the properties available for the module. If changes are made to the properties, restart the Databridge for those changes to take effect.

Table 165. Databridge module properties

Property name	Type	Description
TEMAHOST	string	The name of the host running the monitoring agent. Default: `localhost`
TEMAPORT	integer	The port number used by the host. Default: `9520`

You configure the operation of the Internet service monitoring agent by modifying the property values defined in the monitoring agent properties file.

The monitoring agent properties file is named `kisagent.props`. This file is located in the `$ISMHOME/etc/props/` directory.

The following table lists the properties available for the monitoring agent.

Table 166. Monitoring agent properties

Property name	Type	Description
TEMAPORT	integer	The port number used by the host. This must be the same as the port number for the TEMAPORT property listed in the module properties file. Default: `9520`
`ObsoleteDuration`	integer	The time, in seconds, after which any data that hasn't been updated is deleted from the monitoring agent's memory. Data might not be updated when, for example, a profile element has been stopped or a network failure has occurred. **Note:** Don't set the `ObsoleteDuration` time to a value less than the poll interval because this results in loss of data between poll intervals. Default: `900`

Table 166. Monitoring agent properties (continued)

Property name	Type	Description
AggDuration	integer	The time, in seconds, after which the monitoring agent stops data from being aggregated and reported on the agent dashboard. Any data that is older than the specified time is deleted from the monitoring agent's memory. Old data is calculated by comparing the interval between the start and the current time to the aggregate duration time. If the interval is greater than the aggregate duration time, 10 percent of the old data is removed and the start time is increased by 1/10th of the interval. The monitoring agent calculates this every 5 minutes. Default: 3600
ManageServices	0|1	Starts and stops all monitors and the Databridge when the Internet Service Monitoring agent is started or stopped. 1 is enabled, and 0 is disabled. Default: 1

The connection between the Internet service monitoring agent and the Databridge module is created when you install Internet Service Monitoring.

Enabling Netcool/OMNIbus

Follow these steps to enable the Tivoli Netcool/OMNIbus to send the events from Internet Service Monitoring agent to Netcool/OMNIbus.

Before you begin

Ensure that you installed IBM Tivoli Netcool/OMNIbus.

Procedure

Complete the following steps to enable the Netcool/OMNIbus:

1. Stop the Internet Service Monitoring agent by using the following command:

   ```
   $CANDLEHOME/bin/ism-agent.sh stop
   ```

2. Open the bridge.props file that is placed at $ISMHOME/etc/props path and update the file with following code snippet:

   ```
   Module0SharedLib : "libSMModulePipe"
   Module0PropFile : "$ISMHOME/etc/props/pipe_module.props"
   Module1SharedLib : "libSMModuleObjectServer"
   Module1PropFile : "$ISMHOME/etc/props/objectserver.props"
   ```

3. Modify the permission of 8.1.0 directory placed at $ISMHOME/objectserver path as follows:

   ```
   cd  $ISMHOME/objectserver/
   chmod -R 777 8.1.0
   ```

   Note: Modify the permission of all the files inside 8.1.0 directory by using chmod -R 777 <file-name> command. Where <file-name> is the name of file inside 8.1.0 directory.

4. Modify the omni.dat file placed at $ISMHOME/objectserver/8.1.0/etc path to configure Netcool/OMNIbus server address.
5. Run nco_igen from the following location:

```
cd $ISMHOME/objectserver/8.1.0/bin
./nco_igen
```

6. Start the Internet Service Monitoring agent by using the following command:

   ```
   $CANDLEHOME/bin/ism-agent.sh start
   ```

7. Verify that Internet Service Monitoring agent, Databridge and all the monitors are in running state.

   To check the status of Databridge and monitors, run the following command:

   ```
   ps -aef|grep -i nco_*
   ```

   To check the status of Internet Service Monitoring agent, run the following command:

   ```
   ps -aef|grep -i kis
   ```

8. Use the IBM Tivoli Netcool/OMNIbus user interface to verify that Databridge sends the data to Netcool/OMNIbus server.

   Data must be displayed for Internet Service Monitoring agent on the IBM Application Performance Management user interface.

Migration support

Support of two standalone utilities Properties Migration and Profile Migration are available for Internet Service Monitoring agent version 8.1.4.0.12 and onward. These utilities are developed for migration of agent properties and Internet Services configurations from legacy versions of Internet Service Monitoring agent such as 2.4 and ITM to APM version of the agent.

For more information about the utilities, refer Properties Migration and Profiles Migration utilities of Internet Service Monitoring.

Configuring J2SE monitoring

To collect resource monitoring and diagnostic data from the on-premises Java applications that are being monitored, you must configure the J2SE data collector.

Before you begin

Install one of the supported Java runtimes:

- Oracle Java Platform Standard Edition 7 (Java SE Development Kit 7)

  Remember: This Java Runtime does not support the J2SE data collector image that is configured with HTTPS protocol.

- Oracle Java Platform Standard Edition 7 (Java SE Runtime Environment 7)

  Remember: This Java Runtime does not support the J2SE data collector image that is configured with HTTPS protocol.

- Oracle Java Platform Standard Edition 8 (Java SE Development Kit 8)
- Oracle Java Platform Standard Edition 8 (Java SE Runtime Environment 8)
- IBM SDK, Java Technology Edition, Version 7
- IBM SDK, Java Technology Edition, Version 8

Important: For Windows server 2016, install JDK 8, update 131 (Java SE Development Kit 8u131), or Java SE Development Kit 7, and update 80 (JDK 7u80).

For more information about system requirements, see Software Product Compatibility Reports for J2SE data collector.

About this task

You can configure the J2SE data collector on Windows, Linux, and AIX systems.

The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent and data collector version list and what's new for each version, see the "Change history" on page 54.

Procedure

1. Copy the following files from the APM installer to a directory:

   Important: The directory path must not have any spaces.

   - Windows Copy the `gdc.zip` file from the APM installer to a directory, and extract it.
   - Linux AIX Copy the `gdc-apd.tar.gz` file from the APM installer to a directory, and extract it.
   - Linux AIX Provide read/write and execute permissions to the user for the `j2se_dc` folder. Execute permission is provided to run scripts and JAR files in the folder. Read/write permission is provided because the deep-dive diagnostics files are generated in this folder.
2. On the command line, go to *`DCHOME`*`\.gdc\<toolkit_version>\bin`

   Where *toolkit_verion* is,

   - For V8.1.4.0 and earlier, *toolkit_verion* is 7.3.0.5.0.
   - For V8.1.4.0.1 and later, *toolkit_verion* is 7.3.0.14.0.
3. Run the following command:

   Windows `config.bat`

   Linux AIX `config.sh`
4. When you are prompted, specify the path to Java Home, and press **Enter**.
   For example,

   Windows `C:\Program Files\jre7`

   Linux AIX `/opt/ibm/java`
5. Complete the following steps for the agent version that you use:

   - For V8.1.4.0.2 and earlier, complete the following steps:

     a. When you are prompted, enter the full name (qualified name) of the main class of application, and press **Enter**. The main class is the entry point of the application that needs to be monitored. Example: `testapp.TemperatureConveter`

     b. When you are prompted, enter a distinct application alias name, and press **Enter**. The name that you enter here is used to create the instance name on the APM dashboard.

        Windows The `dcstartup.bat` file is generated at the following location: *`DCHOME`*`\.gdc\`*`toolkit_verion`*`\runtime\j2se`*`application_alias.hostname.application_alias`*. This file is the script to run your application along with the data collector.

        Linux AIX The `dcstartup.sh` file is generated at the following location: *`DCHOME`*`/.gdc/`*`toolkit_verion`*`/runtime/j2se`*`application_alias.hostname.application_alias`*. This file is the script to run your application along with the data collector.
   - For V8.1.4.0.3 to V8.1.4.0.5, complete the following steps

     a. When prompted, enter the home directory of Java application. For example, `/root/J2seApp/`

     b. Select a Java application from the list that is provided to you and then Exit.

        – `com.ibm.SampleApplication`

        – `com.ibm.DBApplication`

      - com.ibm.SpringBootApplication

      Select any application that is provided in the list or provide 0 to select any other application that is not in the list.

      1) If you provide 0, enter `full` name of Main class of any other application. For example, `com.ibm.testApp.Main`

      2) If you select any option from the provided list, alias name is created based on the class name. If the alias name exceeds the character limit, then provide the alias name within the character limit.

         Important: The maximum character limit for alias name is calculated in such a way that `alias_name + host_name` does not exceed 24 characters.

   c. Select the option to enable or disable Transaction Tracking. The default value is *Yes*.

   d. Select the option to enable or disable Diagnostics data collection. The default value is *Yes*.

      1) If you select *Yes*, then select the option to Method Trace mode. The default value is *No*.

   e. If you select an option from the list that is provided in Step b, copy the startup script `<DCHOME>/j2se_dc/.gdc/`*`toolkit_version`*`/runtime/j2se<application_alias>.<hostname>.<application_alias>` to the location of your choice.

- For V8.1.4.0.6 and later, complete the following steps:

   a. When prompted, enter the home directory of Java application. For example, `/root/J2seApp/`

   b. Select the application type that you want to monitor.

      - Java Application
      - Jetty Server

   c. If you select the application type as *Java Application*, then follow the steps that are mentioned in section V8.1.4.0.3 to V8.1.4.0.5 of Step 5.

   d. If you select the application type as *Jetty Server*, then follow these steps:

      1) Enter the Jetty Home directory. For example, `/home/jetty/jetty-distribution-9.4.12.v20180830`

      2) Enter the Alias name. If the alias name exceeds the character limit, then provide the alias name within the character limit.

         Important: If you select the application type as *Java Application* and any option from the list in Step b of section V8.1.4.0.3 to V8.1.4.0.5, then copy the startup script `<DCHOME>/j2se_dc/.gdc/toolkit_version/runtime/j2se<application_alias>.<hostname>.<application_alias>` to the location of your choice.

   e. If the selected type is *Jetty Server*, then `dcstartup.bat/dcstartup.sh` is copied into the given Jetty Home directory.

- For V8.1.4.0.7, complete the following steps:

   - If you configure J2SE data collector by using the Open JDK version 9 or later and when you enter the path to Java Home, a warning displays with the content as follows:

     Warning:

     ```
     WARNING: An illegal reflective access operation has occurred
     WARNING: Illegal reflective access by jnr.posix.JavaLibCHelper
     (file:/root/testopen/preconf-13march/j2se_dc/.gdc/7.3.0.14.0/
     bin/lib/jython.jar)to method sun.nio.ch.SelChImpl.getFD()
     WARNING: Please consider reporting this to the maintainers of
     jnr.posix.JavaLibCHelper
     WARNING: Use --illegal-access=warn to enable warnings of next
     illegal reflective access operations
     WARNING: All illegal access operations will be denied in next
     ```

```
release Mar 15, 2019 11:35:06 AM
org.python.netty.util.internal.PlatformDependent <clinit>
INFO: Your platform does not provide complete lowlevel API for
accessing direct buffers reliably.
Unless explicitly requested, heap buffer is always preferred
to avoid potential system unstability.
```

However, the J2SE data collector works correctly and you can ignore the warning.

For V8.1.4.0.2 and earlier versions, follow step 6 to modify Windows `dcstartup.bat` or Linux AIX `dcstartup.sh` file.

6. To modify the Windows `dcstartup.bat` or Linux AIX `dcstartup.sh` file, complete the following steps:
   - If the application classes and JAR files are bundled in a single JAR file, complete the following steps:
     a. Open the following file:
        - Windows `dcstartup.bat`
        - Linux AIX `dcstartup.sh`

     b. Replace `-cp .:$classpath:$Classpath $ITCAM_JVM_OPTS` *`full name of the main class`* with `$ITCAM_JVM_OPTS -jar` *`Application jar file`* and save the file.
   - If the application is using multiple JAR files, complete the following steps:
     a. Open the following file:
        - Windows `dcstartup.bat`
        - Linux AIX `dcstartup.sh`

     b. Set the *CLASSPATH* variable to the JAR files.

     c. Replace `-cp .:$classpath:$Classpath $ITCAM_JVM_OPTS` *`full name of the main class`* with `-cp .:$classpath:$Classpath $ITCAM_JVM_OPTS -jar` *`Application jar file`* and save the file.

        The application JAR file must contain the main application class.

   Note: To modify Windows `dcstartup.bat` or Linux AIX `dcstartup.sh` file for V8.1.4.0.3 to V8.1.4.0.5 and V8.1.4.0.6 to later (If the Application type is selected as *Java Application*) versions, follow step 7.

7. Complete following steps when your application is using multiple JAR files.

   a) Open the following file:
   - Windows `dcstartup.bat`
   - Linux AIX `dcstartup.sh`

   b) Set the *CLASSPATH* variable to the JAR files.

   c) For V8.1.4.0.7, if you configure the J2SE data collector with Java 9 or 10 and if you use the SSL connection for APM connectivity, then the Transaction Tracking data is not displayed. To resolve the issue, you can add the flag **--add-modules java.xml.bind** to the last line of `dcstartup.bat` or `dcstartup.sh` file.

   For example,

   - If the Application is a jar file, then update the last line as follows:

     ```
     PathToJava --add-modules java.xml.bind --add-opens=
     
     jdk.management/com.sun.management.internal=
     
     ALL-UNNAMED -jar $Classpath $ITCAM_JVM_OPTS AppJarName.jar
     ```

   - If the Application is not bundled in a jar file, then update the last line as follows:

```
PathToJava --add-modules java.xml.bind --add-opens=

jdk.management/com.sun.management.internal=ALL-UNNAMED -cp
.:$classpath:$Classpath $ITCAM_JVM_OPTS FullyQualifiedClassName
```

8. To enable deep-dive diagnostics monitoring, edit the `custom_request.xml` with the J2SE specific classes and methods that you want to monitor. You can do so by two ways: manual and automated process.

   To automatically populate the `custom_request.xml` with the J2SE application-specific classes and methods:

   a) Go to the `<DCHOME>/j2se_dc/.gdc/7.3.0.14.0/runtime/j2se<application_alias>.<hostname>.<application_alias>/` directory and open the `dc.properties` file.

   b) Enable the `is.auto.update.custom_requests.xml` property by setting its value to *true* and save the file.

   c) Run commands from step 9.

   d) Stop the data collector after 10 to 15 minutes.

   e) Check whether the `DCHOME>/j2se_dc/.gdc/7.3.0.14.0/runtime/j2se<application_alias>.<hostname>.<application_alias>/custom/custom_requests.xml` is populated with the custom methods and classes.

   f) Remove the unwanted entries and open the `dc.properties` file again.

   g) Disable the `is.auto.update.custom_requests.xml` property by setting its value to *false* and save the file.

   h) Run commands from step 9.

   Note: If some of the custom methods from the application are not auto-discovered, then you need to add the custom methods manually.

   To manually populate the `custom_request.xml`:

   a) Go to `<DCHOME>/j2se_dc/.gdc/7.3.0.14.0/runtime/j2se<application_alias>.<hostname>.<application_alias>/custom/custom_requests.xml` and edit the `custom_request.xml`.

   For example,

   ```
   <edgeRequest>

   <requestName>truncateDb</requestName>

   <Matches>testApp.JDBC.DBManager</Matches>

   <type>application</type>

   <methodName>truncateDb</methodName>

   </edgeRequest>
   ```

   b) Add the application-specific classes and methods.

9. Run the following command:

   Windows `dcstartup.bat`

   Linux AIX `dcstartup.sh`

   Note: If the selected application type is *Jetty Server*, then run the `dcstartup.bat/dcstartup.sh` present in the Jetty Home directory.

   The J2SE application is started along with the configured data collector.

What to do next

Log in to the Cloud APM console to view the data that is collected by the data collector in the dashboards. For information about using the console, see "Starting the Cloud APM console" on page 957.

For help with troubleshooting, see the Troubleshooting section.

Checking the Status of Transaction Tracking and Diagnostics data collection

For V8.1.0.3 and later, on the agent configuration page, you can check the status of the transaction tracking and diagnostics data.

About this task

You can check the status of transaction tracing and diagnostic data collection with the help of two commands. Refer the procedure to know about these commands.

Procedure

1. Use **config status** command.
   a) Open the bin directory. Issue the command

      Linux AIX
      ```
      cd <DCHOME>/j2se_dc/.gdc/toolkit_version/bin/
      ```
      Windows
      ```
      cd <DCHOME>\j2se_dc\.gdc\toolkit_version\bin\
      ```

      Where *toolkit_version* is,

      - For V8.1.4.0 and earlier, *toolkit_verion* is 7.3.0.5.0.
      - For V8.1.4.0.1 and later, *toolkit_verion* is 7.3.0.14.0.

   b) Enter the following command to check status

      Linux AIX
      ```
      config.sh status
      ```
      Windows
      ```
      config.bat status
      ```

   c) Select the Java Applications with alias name that are identified, from the list to check their status, or select exit.

      1) ddperf
      2) Main
      3) Exit

2. Use **config status <application_alias_name>** command.
   a) Enter the following command to open a directory

      Linux AIX
      ```
      cd <DCHOME>/j2se_dc/.gdc/7.3.0.14.0/bin/
      ```
      Windows
      ```
      cd <DCHOME>\j2se_dc\.gdc\7.3.0.14.0\bin\
      ```

   b) Enter the following command to check status

      Linux AIX
      ```
      config.sh status <application_alias_name>
      ```
      Windows
      ```
      config.bat status <application_alias_name>
      ```

Changing the Status of Transaction Tracking and Diagnostics data collection

For V8.1.0.3 and later, on the agent configuration page, you can change the status of the transaction tracking and diagnostics data.

About this task

You can change the status of transaction tracing and diagnostic data collection with the help of command prompt. Refer the following procedure to know about these commands.

Procedure

1. Open the bin directory and run the following command:

   Linux AIX
   ```
   cd <DCHOME>/j2se_dc/.gdc/toolkit_version/bin/
   ```

Windows

```
cd <DCHOME>\j2se_dc\.gdc\toolkit_version\bin\
```

Where *toolkit_version* is,

- For V8.1.4.0 and earlier, *toolkit_verion* is 7.3.0.5.0.
- For V8.1.4.0.1 and later, *toolkit_verion* is 7.3.0.14.0.

2. To check the status, enter the following command:

   Linux AIX

   ```
   config.sh <application_alias_name>
   ```

   Windows

   ```
   config.bat <application_alias_name>
   ```

3. When prompted select the option to enable or disable Transaction Tracking. Default is `Yes`.
4. When prompted select the option to enable or disable Diagnostics data collection. Default is `Yes`.
   a) If Yes, then select the option to enable/disable Method Trace. Default is `NO`.

Configuring JBoss monitoring

The Monitoring Agent for JBoss monitors the resources of JBoss application servers and the JBoss Enterprise Application platform. Use the dashboards that are provided with the JBoss agent to identify the slowest applications, slowest requests, thread pool bottlenecks, JVM heap memory and garbage collection issues, busiest sessions, and other bottlenecks on the JBoss application server.

Before you begin

- The directions here are for the most current release of the agent, except as indicated.
- Make sure that the system requirements for the JBoss agent are met in your environment. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the JBoss agent.
- Before you configure the JBoss agent, the JBoss server first must be configured by completing the following tasks.
  1. “Enable JMX MBean server connections” on page 450.
  2. “Add a JBoss server management user” on page 451.
  3. “Enabling Web/HTTP Statistic Collection” on page 452. This procedure is for JBoss EAP version 7.x and WildFly versions 8.x, 9.x and 10.x.

About this task

The Managed System Name includes the instance name that you specify, for example, `instance_name:host_name:pc`, where *pc* is your two character product code. The Managed System Name is limited to 32 characters.

The instance name that you specify is limited to 28 characters minus the length of your host name. For example, if you specify `JBoss` as your instance name, your managed system name is `JBoss:hostname:JE`.

Note: If you specify a long instance name, the Managed System name is truncated and the agent code does not display correctly.

The JBoss agent is a multiple-instance agent. You must create an agent instance for each JBoss server you monitor, and start each agent instance manually.

Transaction tracking capability is available for the JBoss agent in the Cloud APM, Advanced offering.

- To enable transaction tracking for a new agent instance, complete step 1 or step 2 of this procedure, then follow the procedure to “Setup the JBoss agent transaction tracking or diagnostics data collector” on page 460.
- To enable transaction tracking for an agent instance that is already configured for basic monitoring, follow the procedure to “Setup the JBoss agent transaction tracking or diagnostics data collector” on page 460.

- To disable transaction tracking for an agent instance, follow the procedure to “Disable the JBoss agent transaction tracking data collector” on page 462.
- To uninstall transaction tracking for all agent instances and remove the transaction tracking toolkit, follow the procedure to “Uninstall all JBoss agent transaction tracking” on page 463.

Procedure

1. Configure the agent on Windows systems by using the **IBM Performance Management** window or by using the silent response file.
   - “Configuring the agent on Windows systems” on page 453.
   - “Configuring the agent by using the silent response file” on page 456.
2. Configure the agent on Linux systems by running command line script and responding to prompts, or by using the silent response file.
   - “Configuring the agent by responding to prompts” on page 455.
   - “Configuring the agent by using the silent response file” on page 456.
3. Optional: Configure transaction tracking by configuring individual agent instances to provide transaction tracking data and configuring your Application Performance Dashboard to display transaction tracking data.

   a) Follow the procedure to “Setup the JBoss agent transaction tracking or diagnostics data collector” on page 460.

   b) Enable the transaction tracking data in the Application Performance Dashboard for the JBoss agent.

      1) From the navigation bar, click **System Configuration** > **Agent Configuration**. The **Agent Configuration** page is displayed.

      2) Select the **JBoss** tab.

      3) Select the check boxes for the JBoss server agent instances that you want to monitor and take one of the following actions from the **Actions** list:

         - To enable transaction tracking, click **Set Transaction Tracking** > **Enabled**. The status in the **Transaction Tracking** column is updated to *Enabled*.
         - To disable transaction tracking, click **Set Transaction Tracking** > **Disabled**. The status in the **Transaction Tracking** column is updated to *Disabled*.

   c) View the JBoss agent transaction tracking data dashboards by adding the JBoss agent instance to an application in your Application Performance Dashboard.

      For more information about using the Applications editor, see Managing applications.

   d) Ensure that user accounts are assigned to a role that includes the Diagnostic Dashboard permission to have access to the following JBoss agent transaction tracking Application Dashboard buttons. Otherwise, these buttons are disabled for that user in the Application Dashboard.

      1) The **Diagnose** drill-down button on the **Slowest 5 Response Time** widget.

      2) The **Inflight Requests** button on the **Applications** widget.

   Note: Transaction tracking capability is available for the JBoss agent in the Cloud APM, Advanced offering. For the JBoss agent with basic resource monitoring capability, which is in the Cloud APM, Base offering, skip this step.

What to do next

In the Cloud APM console, go to your Application Performance Dashboard to view the data that was collected. For more information about using the Cloud APM console, see “Starting the Cloud APM console” on page 957.

If you are unable to view the data in the agent dashboards, first check the server connection logs and then the data provider logs. The default paths to these logs are as follows.

- Linux `/opt/ibm/apm/agent/logs`
- Windows `C:\IBM\APM\TMAITM6_x64\logs`

For help with troubleshooting, see the Cloud Application Performance Management Forum.

Enable JMX MBean server connections

Before the JBoss agent can gather data from the JBoss server, Java Management Extensions (JMX) MBean server connections must be enabled.

Procedure

Follow the steps for your JBoss server release and version.

- Configure EAP 5.2.

  Make a backup copy of the `run.conf` file, then add the following lines to it:

  ```
  JAVA_OPTS="$JAVA_OPTS -Djboss.platform.mbeanserver"
  JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote"
  JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.ssl=false"
  JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.authenticate=false"
  JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.port=1090"
  JAVA_OPTS="$JAVA_OPTS -Djavax.management.builder.initial=
                         org.jboss.system.server.jmx.MBeanServerBuilderImpl"
  ```

- Configure AS 6.x.

  Specify the bind address as a parameter when you start the JBoss server.

  - Linux *jboss_server_home*/bin/run.sh **-b** Ip_address
  - Windows *jboss_server_home*\bin\run.bat **-b** <Ip_address>

  where *jboss_server_home* is the JBoss server installation directory.

  For example, if the bind address is 10.77.9.250:

  /apps/wildfly-9.0.2.Final/bin/run.sh **-b** 10.77.9.250

- Configure all other supported versions.

  JBoss and WildFly servers are installed with their JMX ports disabled for remote management by default. You must change the configuration of the JBoss server to allow remote management. You must edit the *jboss_server_home*/standalone/configuration/standalone.xml to allow remote management.

  a) Make a backup copy of *jboss_server_home*/standalone/configuration/standalone.xml file.

     Where *jboss_server_home* is the JBoss server installation directory.

  b) Allow remote configuration.

     Search for `urn:jboss:domain:jmx` and within its subsystem section, make sure that the `remoting-connector` entry has `use-management-endpoint="true"`.

     Example result.

     ```
     <subsystem xmlns="urn:jboss:domain:jmx:1.3">
                 <expose-resolved-model/>
                 <expose-expression-model/>
                 <remoting-connector use-management-endpoint="true"/>
     </subsystem>
     ```

  c) Allow remote connections.

     Find where the interfaces are defined and replace 127.0.0.1 (loopback) with the external IP on the server to bind to. Do not bind to 0.0.0.0.

     Example before replacement.

```
<interfaces>
      <interface name="management">
          <inet-address value="${jboss.bind.address.management:127.0.0.1}"/>
      </interface>
      <interface name="public">
          <inet-address value="${jboss.bind.address:127.0.0.1}"/>
      </interface>
...
```

Example after the replacement if the external IP address is 192.168.101.1.

```
<interfaces>
      <interface name="management">
          <inet-address value="${jboss.bind.address.management:192.168.101.1}"/>
      </interface>
      <interface name="public">
          <inet-address value="${jboss.bind.address:192.168.101.1}"/>
      </interface>
...
```

Add a JBoss server management user

Before the JBoss agent can gather data from the JBoss server, a management user must be added if one does not exist.

Procedure

Use the JBoss **`add-user`** script to add a management user.

1. Go to the binary or `bin` directory under the JBoss server installation directory.
2. Run the **`add-user`** script.
   - Linux **`./add-user.sh`**
   - Windows **`add-user.bat`**
3. Follow the prompts to generate a management user.

Example

```
root@jboss-wf10-rh7:/apps/wildfly-10.0.0.Final/bin
] ./add-user.sh

What type of user do you wish to add?
 a) Management User (mgmt-users.properties)
 b) Application User (application-users.properties)
(a): a

Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : MyAdmin
Password recommendations are listed below. To modify these restrictions edit the add-
user.properties
configuration file.
 - The password should be different from the username
 - The password should not be one of the following restricted values {root, admin,
administrator}
 - The password should contain at least 8 characters, 1 alphabetic character(s), 1 digit(s),
1 non-alphanumeric symbol(s)
Password :
Re-enter Password :
What groups do you want this user to belong to? (Please enter a comma separated list, or leave
blank
for none)[  ]:
About to add user 'MyAdmin' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'MyAdmin' to file '/apps/wildfly-10.0.0.Final/standalone/configuration/mgmt-
users.properties'
Added user 'MyAdmin' to file '/apps/wildfly-10.0.0.Final/domain/configuration/mgmt-
users.properties'
Added user 'MyAdmin' with groups  to file
           '/apps/wildfly-10.0.0.Final/standalone/configuration/mgmt-groups.properties'
Added user 'MyAdmin' with groups  to file
           '/apps/wildfly-10.0.0.Final/domain/configuration/mgmt-groups.properties'
```

```
Is this new user going to be used for one AS process to connect to another AS process?
e.g. for a slave host controller connecting to the master or for a Remoting connection for
server to
server EJB calls.
yes/no? no
```

Enabling Web/HTTP Statistic Collection

Before the JBoss agent can gather JBoss server web metrics and other subsystem metrics, statistics collection must be enabled for each subsystem. This procedure is for JBoss EAP version 7.x and WildFly versions 8.x, 9.x and 10.x.

Procedure

The **statistics-enabled** attribute of various JBoss subsystems controls statistic collection. This setting can be viewed and updated by using the JBoss command line interface.

Note: This procedure is for JBoss EAP version 7.x and WildFly versions 8.x, 9.x and 10.x.

1. Go to the binary or bin directory under the JBoss server installation directory.
2. Start the JBoss command line interface.
   - Linux **./jboss-cli.sh --connect** [**--controller**=*IP*:*port*]
   - Windows **jboss-cli.bat --connect** [**--controller**=*IP*:*port*]

   where *IP* is the JBoss server's IP address and *port* is the JBoss server's port. For example, 192.168.10.20:9990.

   Tip: If the connection attempt results in the error, "Failed to connect to the controller: The controller is not available at localhost:9990: java.net.ConnectException: WFLYPRT0053: Could not connect to http-remoting://localhost:9990. The connection failed: WFLYPRT0053: Could not connect to http-remoting://localhost:9990. The connection failed: Connection refused", use the **--controller** parameter.

   This error indicates that the management server is not listening on the localhost IP address (127.0.0.1) and is configured to listen on the computer's IP address.
3. Run the following commands to view the current state of each subsystem's statistics-enabled attribute:

   Note: If JBoss is running in Domain Mode, each command must be prefixed with the associated profile and these commands must be run for each monitored profile. For example: /profile=full/subsystem=ejb3:read-attribute(name=statistics-enabled)

   /subsystem=**ejb3**:read-attribute(name=enable-statistics)

   /subsystem=**transactions**:read-attribute(name=statistics-enabled)

   /subsystem=**undertow**:read-attribute(name=statistics-enabled)

   /subsystem=**webservices**:read-attribute(name=statistics-enabled)

   /subsystem=**datasources/data-source**=*Data_Source_Name*:read-attribute(name=statistics-enabled)

   /subsystem=**datasources/data-source**=*Data_Source_Name*/**statistics=pool**:read-attribute(name=statistics-enabled)

   /subsystem=**datasources/data-source**=*Data_Source_Name*/**statistics=jdbc**:read-attribute(name=statistics-enabled)

   where *Data_Source_Name* is the name of a data source that is configured for use with JBoss.

   Note: Data sources can be listed by using the command /subsystem=datasources:read-resource.

   Example result when statistics are not enabled:

```
{
    "outcome" => "success",
    "result" => false
}
```

4. Run the following command to change the value of each subsystem's statistics-enabled attribute to *true*:

   ```
   /subsystem=ejb3:write-attribute(name=enable-statistics, value=true)
   /subsystem=transactions:write-attribute(name=statistics-enabled,value=true)
   /subsystem=undertow:write-attribute(name=statistics-enabled,value=true)
   /subsystem=webservices:write-attribute(name=statistics-enabled,value=true)
   /subsystem=datasources/data-source=Data_Source_Name:write-attribute(name=statistics-enabled,value=true)
   /subsystem=datasources/data-source=Data_Source_Name/statistics=pool:write-attribute(name=statistics-enabled,value=true)
   /subsystem=datasources/data-source=Data_Source_Name/statistics=jdbc:write-attribute(name=statistics-enabled,value=true)
   ```

   Example result when you enable statistics for a subsystem:

   ```
   {
       "outcome" => "success",
       "response-headers" => {
           "operation-requires-reload" => true,
           "process-state" => "reload-required"
       }
   }
   ```

5. Exit the JBoss command line interface.
6. Restart the JBoss server.

   Note: Any currently running JBoss agents with transaction tracking enabled must be restarted.

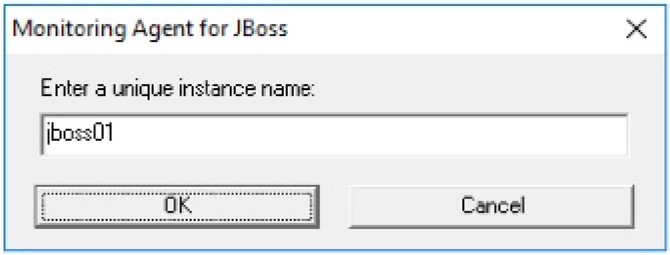
Configuring the agent on Windows systems

You can configure the JBoss agent on Windows operating systems by using the IBM Cloud Application Performance Management window. After you update the configuration values, you must start the agent to save the updated values.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Cloud Application Performance Management**.
2. In the **IBM Performance Management** window, right-click the **Monitoring Agent for JBoss** template, and then click **Configure agent**.

   Remember: After you configure an agent instance for the first time, the **Configure agent** option is disabled. To configure the agent instance again, right-click on it and then click **Reconfigure**.
3. Enter a unique instance name then click **OK**. Use only letters, Arabic numerals, the underline character, and the minus character in the instance name.
   For example, `jboss01`.

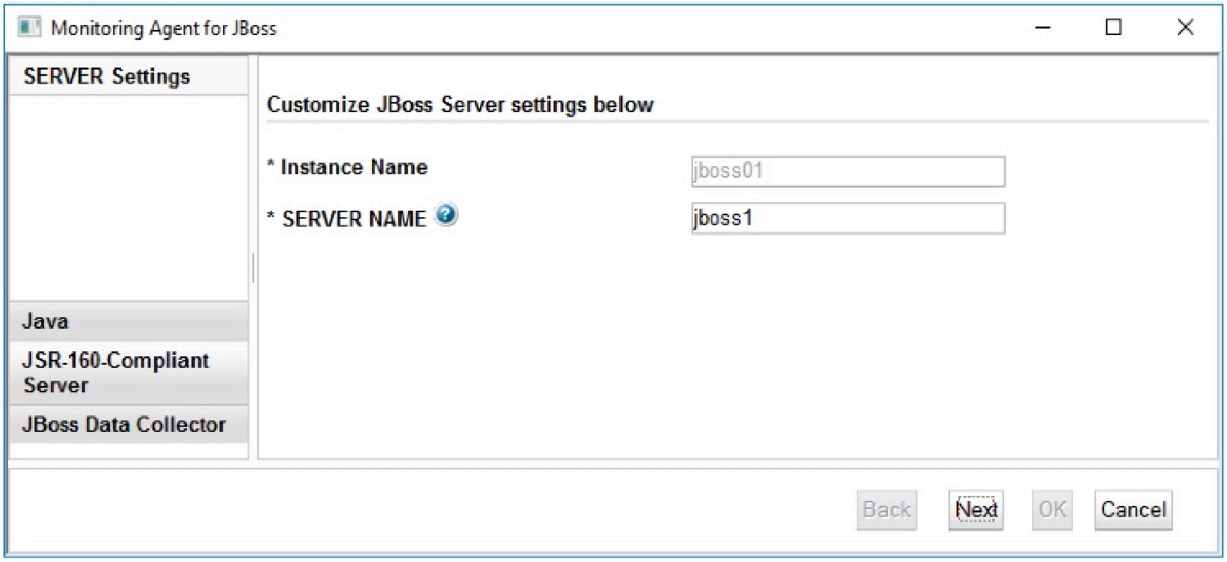


Enter a unique instance name: jboss01

Figure 17. The window to enter a unique instance name.

4. Enter the JBoss Server settings, then click **Next**.

   See Table 167 on page 458 for an explanation of each of the configuration parameters.



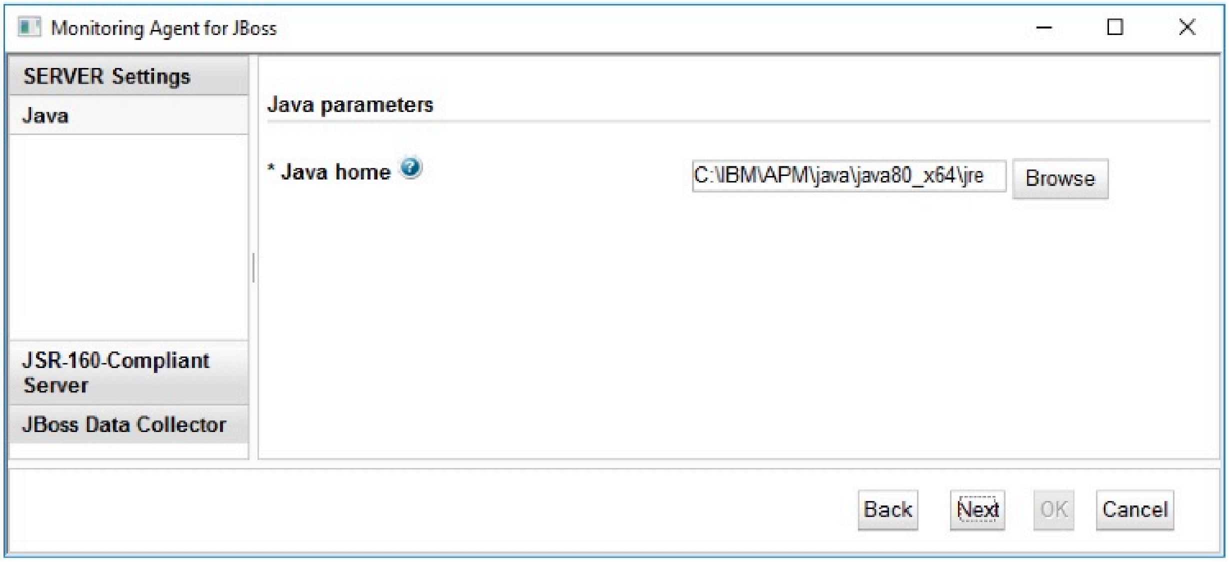
* Instance Name: jboss01

* SERVER NAME: jboss1

Figure 18. The window for configuration parameters for the JBoss server

5. Enter the Java settings, then click **Next**.

   See Table 167 on page 458 for an explanation of each of the configuration parameters.

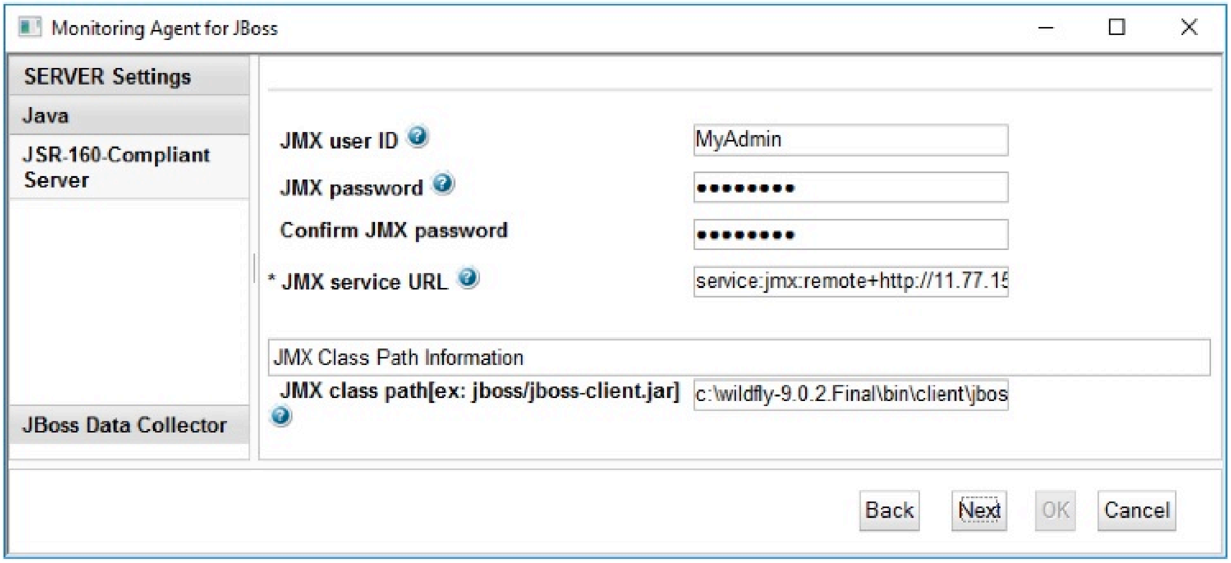


* Java home: C:\IBM\APM\java\java80_x64\jre

Figure 19. The window to specify Java settings.

6. Enter the JMX settings, then click **Next**.

   See Table 167 on page 458 for an explanation of each of the configuration parameters.



JMX user ID: MyAdmin

JMX password: ••••••••

Confirm JMX password: ••••••••

* JMX service URL: service:jmx:remote+http://11.77.15

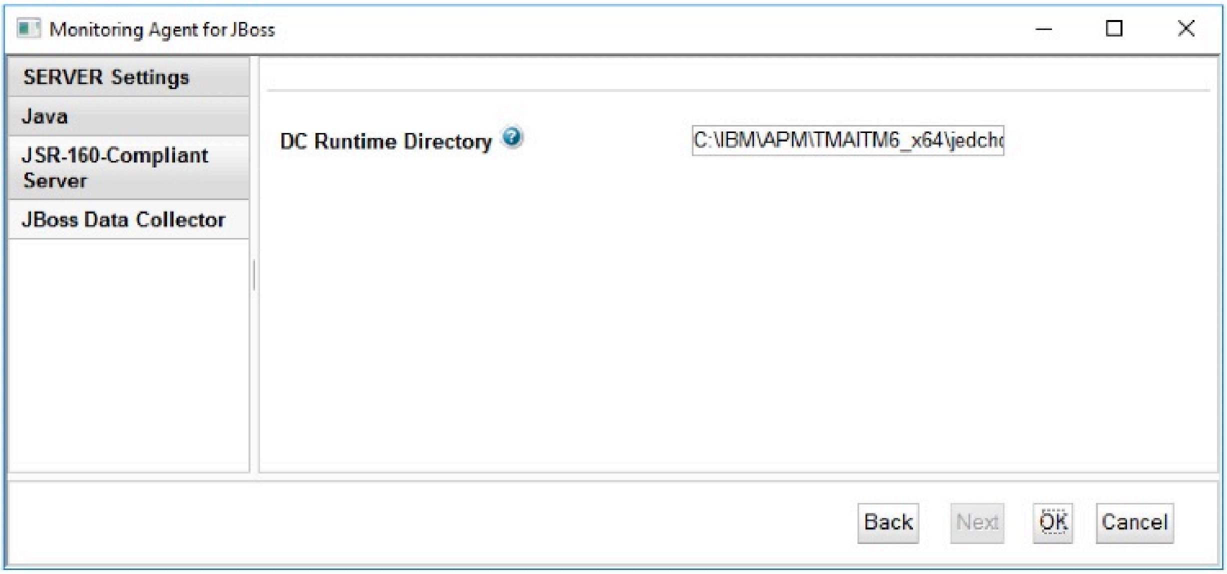
JMX Class Path Information

JMX class path[ex: jboss/jboss-client.jar]: c:\wildfly-9.0.2.Final\bin\client\jbos

Figure 20. The window to specify JMX settings.

7. View the JBoss agent data collector settings.

   Leave the **`DC Runtime Directory`** blank during initial configuration of the agent. See Table 167 on page 458 for an explanation of each of the configuration parameters.



DC Runtime Directory: C:\IBM\APM\TMAITM6_x64\jedch

Figure 21. The window to specify JBoss agent data collector settings

8. Click **OK** to complete the agent configuration.
9. In the IBM Cloud Application Performance Management window, right-click the instance that you configured, and then click **Start**.

Configuring the agent by responding to prompts

After installation of the JBoss agent, you must configure it before you start the agent. If the JBoss agent is installed on a local Linux or UNIX computer, you can follow these instructions to configure it interactively through command line prompts.

About this task

Remember: If you are reconfiguring a configured agent instance, the value that is set in the last configuration is displayed for each setting. If you want to clear an existing value, press the space key when the setting is displayed.

Procedure

1. On the command line, run the following command:

   ```
   install_dir/bin/jboss-agent.sh config instance_name
   ```

   where *install_dir* is the path where the agent is installed and *instance_name* is the name that you want to give to the agent instance.

   Example

   ```
   /opt/ibm/apm/agent/bin/jboss-agent.sh config example-inst01
   ```

2. Respond to the prompts to set configuration values for the agent.

   See "Configuration parameters for the JBoss agent" on page 457 for an explanation of each of the configuration parameters.

3. Run the following command to start the agent:

   ```
   install_dir/bin/jboss-agent.sh start instance_name
   ```

   where *install_dir* is the path where the agent is installed and *instance_name* is the name of the agent instance.

   Example

   ```
   /opt/ibm/apm/agent/bin/jboss-agent.sh start example-inst01
   ```

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

The silent response file contains the agent configuration parameters with default values that are defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

- To configure the JBoss agent in the silent mode, complete the following steps:

  a) In a text editor, open the `jboss_silent_config.txt` file that is available at the following path:

  – Linux AIX `install_dir/samples/jboss_silent_config.txt`

    Example, `/opt/ibm/apm/agent/samples/jboss_silent_config.txt`

  – Windows `install_dir\samples\jboss_silent_config.txt`

  where *install_dir* is the path where the agent is installed.

  The default *install_dir* paths are listed here:

  – Linux `/opt/ibm/apm/agent`

  – Windows `C:\IBM\APM\TMAITM6_x64`

  Example

  ```
  Linux AIX /opt/ibm/apm/agent/samples/jboss_silent_config.txt
  ```

```
Windows C:\IBM\APM\samples\jboss_silent_config.txt
```

b) In the `jboss_silent_config.txt` file, specify values for all mandatory parameters. You can also modify the default values of other parameters.

See "Configuration parameters for the JBoss agent" on page 457 for an explanation of each of the configuration parameters.

c) Save and close the `jboss_silent_config.txt` file, and run the following command:

- Linux AIX *install_dir*/bin/jboss-agent.sh config **instance_name** *install_dir*/samples/jboss_silent_config.txt
- Windows *install_dir*\bin\jboss-agent.bat config **instance_name** *install_dir*\samples\jboss_silent_config.txt

where *install_dir* is the path where the agent is installed and *instance_name* is the name of the agent instance.

The default *install_dir* paths are listed here:

- Linux /opt/ibm/apm/agent
- Windows C:\IBM\APM\TMAITM6_x64

Important: Ensure that you include the absolute path to the silent response file. Otherwise, the agent data is not shown in the dashboards.

Example

```
Linux AIX /opt/ibm/apm/agent/bin/jboss-agent.sh config example-inst01
/opt/ibm/apm/agent/samples/jboss_silent_config.txt
```

```
Windows C:\IBM\APM\bin\jboss-agent.bat config example-inst01 C:\IBM\APM\samples
\jboss_silent_config.txt
```

d) Run the following command to start the agent:

- Linux AIX *install_dir*/bin/jboss-agent.sh start **instance_name**
- Windows *install_dir*\bin\jboss-agent.bat start **instance_name**

where *install_dir* is the path where the agent is installed and *instance_name* is the name of the agent instance.

The default *install_dir* paths are listed here:

- Linux /opt/ibm/apm/agent
- Windows C:\IBM\APM\TMAITM6_x64

Example

```
Linux AIX /opt/ibm/apm/agent/bin/jboss-agent.sh start example-inst01
```

```
Windows C:\IBM\APM\bin\jboss-agent.bat start example-inst01
```

Configuration parameters for the JBoss agent

The configuration parameters for the JBoss agent are displayed in a table.

1. JBoss Agent Settings - JBoss agent environment settings.
2. Table 168 on page 458 - Example JMX service URLs.

Table 167. JBoss Agent Settings

Parameter name	Description	Silent configuration file parameter name
Server Name	Provide a name to identify the JBoss/ WildFly Server.	**KJE_SERVER**
Java home	The path to where Java is installed.	**JAVA_HOME**
JMX user ID	The user ID for connecting to the MBean server.	**KQZ_JMX_JSR160_JSR160_USER_ID**
JMX password	Password	**KQZ_JMX_JSR160_JSR160_PASSWORD**
JMX service URL	The service URL for connecting to the MBean server. See Table 168 on page 458 for examples.	**KQZ_JMX_JSR160_JSR160_SERVICE_URL**
JMX class path	The JAR files that are searched to locate a class or resource. Locate and enter the path to the `jboss-client.jar` file for your JBoss server. Example for a JBoss EAP 6 server, `/opt/EAP-6.3.0/jboss-eap-6.3/bin/client/jboss-client.jar`.	**KQZ_JMX_JSR160_JSR160_JAR_FILES**
DC Runtime Directory	**Note:** This parameter is only for the JBoss agent with transaction tracking capability, which is in the Cloud APM, Advanced offering. For the JBoss agent with basic resource monitoring capability, which is in the Cloud APM, Base offering, skip this parameter. The full path to the JBoss data collector runtime directory is set by the **simpleConfig** script. Leave this parameter blank during the initial configuration of the agent.	**KQZ_DC_RUNTIME_DIR**

Table 168. JMX service URLs

JBoss server version	JMX service URL with default port[1]
WildFly 8, 9 and 10 JBoss EAP 7	`service:jmx:remote+http://ip:9990` `service:jmx:remote+https://ip:9994`
JBoss EAP 6 JBoss AS 7	`service:jmx:remoting-jmx://ip:9999`
JBoss EAP 5.2 JBoss AS 6.1	`service:jmx:rmi:///jndi/rmi://ip:1090/jmxrmi`

[1] The port is based on the port in the JBoss configuration file entry `<socket-binding name="management-native" interface="management" port="${jboss.management.native.port:*NNNN*}"/>`. If the port was changed from the default value, adjust it according to the port number in your configuration file.

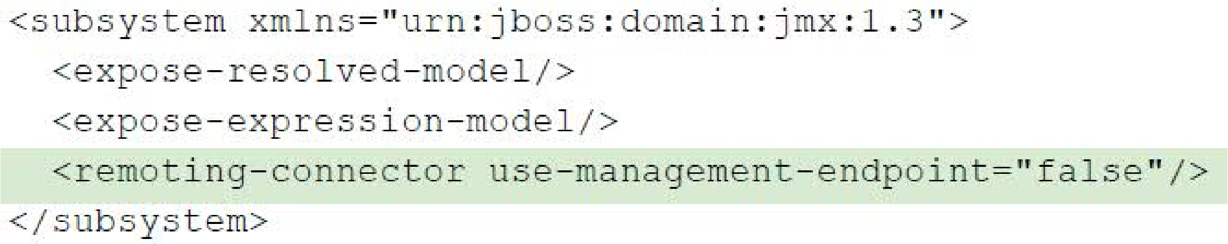
Configuring JBoss agent in domain mode

To monitor JBoss servers in domain mode with JBoss agent, follow the steps:

About this task

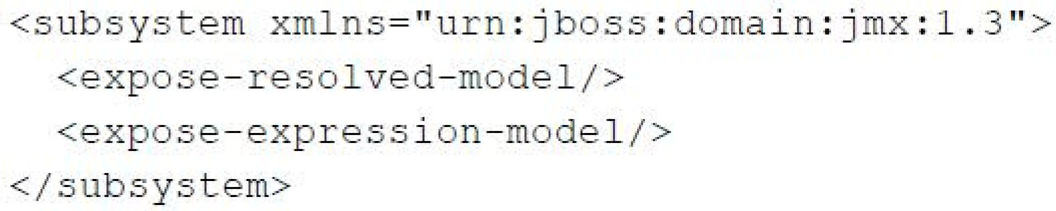
Procedure

1. Install JBoss agent on each computer that contains JBoss servers you want to monitor.
2. Add the following line to the `domain.xml` file under all socket profiles used by the JBoss servers that you want to monitor.



```
<subsystem xmlns="urn:jboss:domain:jmx:1.3">
  <expose-resolved-model/>
  <expose-expression-model/>
  <remoting-connector use-management-endpoint="false"/>
</subsystem>
```

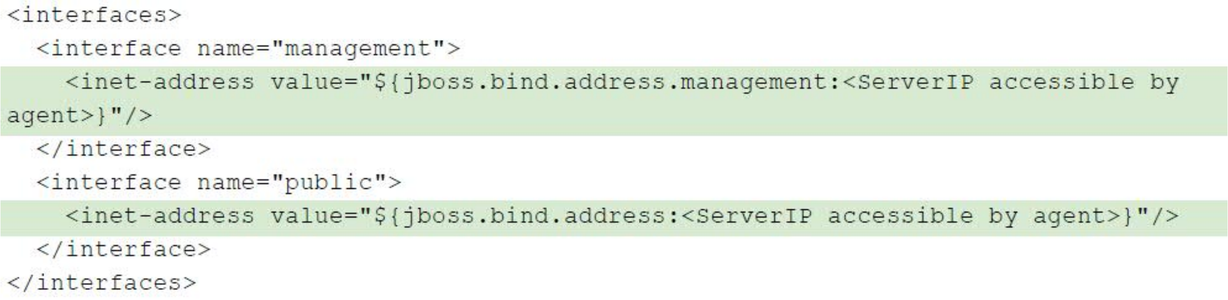
Note: The following lines might occur multiple times (once per socket profile).



```
<subsystem xmlns="urn:jboss:domain:jmx:1.3">
  <expose-resolved-model/>
  <expose-expression-model/>
</subsystem>
```

Note: Server socket profiles are set in the host XML files. The default host XML file is host.xml in the configuration directory, but can be overridden with the `--host-config` option when you run the JBoss domain executable file. Make sure to edit the correct files for your system.

3. Update the host XML file for each JBoss host to set its IP address. Do not use loopback address, such as 127.0.0.1 and local host.



```
<interfaces>
  <interface name="management">
    <inet-address value="${jboss.bind.address.management:<ServerIP accessible by agent>}"/>
  </interface>
  <interface name="public">
    <inet-address value="${jboss.bind.address:<ServerIP accessible by agent>}"/>
  </interface>
</interfaces>
```

Note: If you edit the IP address of your domain controller, ensure that the host configurations are modified to use the updated domain controller IP address.

4. Configure the JBoss agent on each computer containing JBoss servers you wish to monitor. For more information, see “Configuring the agent by responding to prompts” on page 455 and “Configuring the agent on Windows systems” on page 453.
5. Start the JBoss agent by running the command:

```
install_dir/bin/jboss-agent.sh start instance_name
```

6. Verify the monitored JBoss servers are displayed on Cloud APM console correctly.

Setup the JBoss agent transaction tracking or diagnostics data collector

If you want to use the transaction tracking or diagnostics capability of the JBoss agent, you need to make some changes to the agent instance environment settings file, the JBoss server startup file, and the DC Runtime Directory agent configuration parameter. A script is provided to help you make the changes.

Before you begin

Linux Ensure that the resource limit for open files is greater than 5,000 for the transaction tracking or diagnostics toolkit to work properly.

- Display the current open file limit setting. **`ulimit -n`**
- Example setting the open file limit to 5,056. **`ulimit -n 5056`**

Perform Configuring the JBoss agent Step "1" on page 449 or "2" on page 449 before you follow this procedure.

The JBoss agent must be installed locally to the JBoss server that is monitored with the transaction tracking or diagnostics capability.

The user account that runs this script must have write permission to the following directories and files:

1. The *JBOSS_HOME* directory.
2. The *JBOSS_HOME*`/bin` directory and files.
3. The *JBOSS_HOME*`/modules/system/layers/base/org/jboss/as/server/main/module.xml` file.
4. The *install_dir*`/config` directory.
5. The *install_dir*`/config/`*hostname*`_je_`*instance_name*`.cfg` file.

where

JBOSS_HOME
JBoss server installation directory.

install_dir
Path where the agent is installed. The default path is:

- Linux `/opt/ibm/apm/agent`
- Windows `C:\IBM\APM\TMAITM6_x64`

hostname
The name of the host computer where the agent is installed.

instance_name
Name of the agent instance that is assigned in the agent configuration method topic:

- Configuring the agent on Windows systems, step "3" on page 453
- Configuring the agent by responding to prompts, step "1" on page 456
- Configuring the agent by using the silent response file, step "3" on page 457

Procedure

Run the **`simpleConfig`** script.

1. Log in to the JBoss server with the JBoss agent installed.
2. Change directory to the agent installation directory.

   - Linux *install_dir*
   - Windows *install_dir*`\TMAITM6_x64`

   where *install_dir* is the path where the agent is installed.

   The default *install_dir* paths are listed here:

   - Linux `/opt/ibm/apm/agent`
   - Windows `C:\IBM\APM\TMAITM6_x64`
3. Change directory to `jedchome/7.3.0.13.0/bin`.
4. Run the setup script.
   - Linux `./simpleConfig.sh`
   - Windows `simpleConfig.bat`

   Note: The setup script is not working for JBoss servers in domain mode. To enable the transaction tracking or diagnostics capability for JBoss servers in domain mode, you must configure JBoss agents as in the "Configuring JBoss agent in domain mode" on page 459 topic and then enable the transaction tracking or diagnostics capability by following the steps in the JBoss TT and DD configuration in domain mode document. To get this document, contact with IBM support team.
5. Follow the prompts to enter parameters for your environment:
   a) Enter the JBoss agent *instance_name* chosen for the agent instance.

   b) Enter the JBoss server installation directory.

      If the *JBOSS_HOME* environment variable is set, its value will be offered as the default value.

   where

   JBOSS_HOME
      The JBoss server installation directory.

   instance_name
      The name of the agent instance that is assigned in the agent configuration method topic:
      - Configuring the agent on Windows systems, step "3" on page 453
      - Configuring the agent by responding to prompts, step "1" on page 456
      - Configuring the agent by using the silent response file, step "3" on page 457

   install_dir
      The path where the agent is installed. The default path is:
      - Linux `/opt/ibm/apm/agent`
      - Windows `C:\IBM\APM\TMAITM6_x64`
6. Restart the JBoss server and the agent if they are running.

Results

JBoss server files that are changed during transaction tracking or diagnostics configuration:

- *`JBOSS_HOME`*`/bin/standalone.conf`

  This file is updated with configuration settings required for the transaction tracking or diagnostics capability. Configuration markers are inserted into the file for use when you disable the transaction tracking or diagnostics capability. A backup file is saved in the *`JBOSS_HOME`*`/`bak directory prior to adding or removing the transaction tracking or diagnostics capability changes.
- *`JBOSS_HOME`*`/modules/system/layers/base/org/jboss/as/server/main/module.xml`

  This file is updated with a JAVA EE API module dependency. Configuration markers are inserted into the file for use when you disable the transaction tracking or diagnostics capability. A backup file is saved in the *`JBOSS_HOME`*`/`bak directory prior to adding or removing the transaction tracking or diagnostics capability changes.

Agent files that are changed during transaction tracking or diagnostics configuration:

- Agent instance configuration file
  - Linux *`install_dir`*`/config/`*`hostname`*`_je_`*`instance_name`*`.cfg`

  - Windows *install_dir*\TMAITM6_x64*hostname*_JE_*instance_name*.cfg
- Agent environment settings file
  - Linux *install_dir*/config/je_*instance_name*.environment
  - Windows *install_dir*\TMAITM6_x64\KJEENV_*instance_name*

where

JBOSS_HOME
JBoss server installation directory.

install_dir
Path where the agent is installed. The default path is:

- Linux `/opt/ibm/apm/agent`
- Windows `C:\IBM\APM\TMAITM6_x64`

hostname
The name of the host computer where the agent is installed.

instance_name
Name of the agent instance that is assigned in the agent configuration method topic:

- Configuring the agent on Windows systems, step "3" on page 453
- Configuring the agent by responding to prompts, step "1" on page 456
- Configuring the agent by using the silent response file, step "3" on page 457

Disable the JBoss agent transaction tracking data collector

The transaction tracking capability of the JBoss agent requires changes to the agent instance environment settings file, the JBoss server startup file, and the DC Runtime Directory agent configuration parameter. A script is provided to remove these changes for an agent instance with transaction tracking enabled.

Before you begin

Ensure that the JBoss server and the JBoss agent are shut down.

The user account that runs this script must have write permission to the following directories and files:

1. The *JBOSS_HOME* directory
2. The *JBOSS_HOME*/bin directory and files
3. The *JBOSS_HOME*/modules/system/layers/base/org/jboss/as/server/main/module.xml file
4. The *install_dir*/config directory
5. The *install_dir*/config/*hostname*_je_*instance_name*.cfg file

Procedure

Run the **simpleConfig** script with the **remove** option.

1. Log in to the JBoss server with the JBoss agent installed.
2. Change directory to the agent installation directory.
   - Linux *install_dir*
   - Windows *install_dir*\TMAITM6_x64
3. Change directory to `jedchome/7.3.0.13.0/bin`.
4. Run the **simpleConfig** with the **remove** option.
   - Linux `./simpleConfig.sh` **remove** *instance_name*

- Windows simpleConfig.bat **remove** *instance_name*

5. Start the JBoss server and the agent.

   Where:

   JBOSS_HOME
   The JBoss server installation directory

   hostname
   The name of the host computer where the agent is installed

   instance_name
   The name of the agent instance that is assigned in the agent configuration method topic:

   - Configuring the agent on Windows systems, step "3" on page 453
   - Configuring the agent by responding to prompts, step "1" on page 456
   - Configuring the agent by using the silent response file, step "3" on page 457

   install_dir
   The path where the agent is installed. The default path is:

   - Linux /opt/ibm/apm/agent
   - Windows C:\IBM\APM\TMAITM6_x64

   architecture
   The IBM Application Performance Management or Cloud APM system architecture identifier. For example, lx8266 represents Linux Intel v2.6 (64-bit). For a complete list of the architecture codes, see the *install_dir*/registry/archdsc.tbl file.

Uninstall all JBoss agent transaction tracking

The transaction tracking capability of the JBoss agent can be uninstalled. A script is provided to remove all agent instance with transaction tracking enabled and also remove the transaction tracking toolkit.

Before you begin

Ensure that the JBoss server and all JBoss agent instances are shut down.

The user account that runs this script must have write permission to the following directories and files:

1. The *JBOSS_HOME* directory.
2. The *JBOSS_HOME*/bin directory and files.
3. The *JBOSS_HOME*/modules/system/layers/base/org/jboss/as/server/main/module.xml file.
4. The *install_dir*/config directory.
5. The *install_dir*/config/*hostname*_je_*instance_name*.cfg file.

Procedure

Run the **simpleConfig** script with the **uninstall** option.

1. Log in to the JBoss server with the JBoss agent installed.
2. Change directory to the agent installation directory.
   - Linux *install_dir*/*architecture*/je/bin. For example: /opt/ibm/apm/agent/lx8266/je/bin or /opt/ibm/apm/agent/lx8266/je/bin
   - Windows *install_dir*\TMAITM6_x64
3. Change directory to jedchome/7.3.0.13.0/bin.
4. Run the **simpleConfig** with the **uninstall** option.
   - Linux ./simpleConfig.sh **uninstall**

- Windows `simpleConfig.bat` **`uninstall`**

5. Start the JBoss server and all agent instances.

   Where:

   JBOSS_HOME
   The JBoss server installation directory.

   hostname
   The name of the host computer where the agent is installed.

   instance_name
   The name of the agent instance that is assigned in the agent configuration method topic:

   - Configuring the agent on Windows systems, step "3" on page 453
   - Configuring the agent by responding to prompts, step "1" on page 456
   - Configuring the agent by using the silent response file, step "3" on page 457

   install_dir
   The path where the agent is installed. The default path is:

   - Linux `/opt/ibm/apm/agent`
   - Windows `C:\IBM\APM\TMAITM6_x64`

   architecture
   The IBM Application Performance Management or Cloud APM system architecture identifier. For example, lx8266 represents Linux Intel v2.6 (64-bit). For a complete list of the architecture codes, see the *`install_dir`*`/registry/archdsc.tbl` file.

Configuring Liberty monitoring

You can use the Liberty data collector to monitor your Liberty.

- To monitor Liberty V18.* and older versions, do the steps as instructed in "Configuring the Liberty data collector for Liberty V18.* and older versions" on page 464.

  Note: Liberty V19 and newer versions are not supported by Cloud APM in SaaS.

Configuring the Liberty data collector for Liberty V18.* and older versions

For Liberty V18.* and older versions, you can use both the new configuration process and old configuration process. See the following topics to learn how to use previous method to configure the Liberty data collector.

Configuring the Liberty data collector in on-premises environments (Liberty V18.* and older versions)

To monitor the Liberty profile on Linux for System x, you can directly deploy a stand-alone data collector to your local Liberty directory without installing WebSphere Applications agent.

Before you begin

1. Download the `IBM_Data_Collectors_Install.tgz` data collector package from the IBM Passport Advantage® website. For detailed instructions, see "Downloading your agents and data collectors" on page 105.
2. If your firewall rules do not allow transparent outbound HTTPS connections to external hosts, you can configure data collectors to send traffic to a forward proxy. For instructions, see "Configuring data collectors to communicate through a forward proxy" on page 163.
3. The monitor-1.0 feature is required by the data collector. You can download this feature from the Liberty feature repository with the **`installUtility`** command. For instructions, see the section about downloading assets in the WebSphere Application Server Network Deployment Knowledge Center.

4. For the Memory Analysis dashboard to contain data, you must enable memory allocation collection for the data collector during configuration. This diagnostic feature requires IBM Health Center 3.0.8 or later. If the IBM Health Center version is not eligible, upgrade the JRE that is used by the application server to a version that contains IBM Health Center 3.0.8 or later.

   Tip: To check the IBM Health Center version that is included in the JRE used by the application server, change to the `bin` directory within the JRE home directory and then issue `java -Xhealthcenter -version`.

About this task

You can choose to manually configure the data collector or to use the provided configuration script to configure the data collector.

Procedure

- To manually configure the data collector, get the data collector files from the data collector package and then modify some local files for the Liberty server.

  a) Run the following command to extract files from the data collector package.

  ```
  tar -xzf IBM_Data_Collectors_Install.tgz
  ```

  The `liberty_datacollector_8.1.4.0.tgz` package is included in the extracted directory.

  b) Extract files from the `liberty_datacollector_8.1.4.0.tgz` package to a local directory with the following command. The extracted directory will become the home directory of the data collector.

  ```
  tar -xzf liberty_datacollector_8.1.4.0.tgz
  ```

  For example, to extract the files to the `/opt/ibm/apm/` directory, issue the following commands:

  ```
  cd /opt/ibm/apm
  tar -xzf liberty_datacollector_8.1.4.0.tgz
  ```

  You can find the extracted files in the `/opt/ibm/apm/liberty_dc/.gdc/7.3.0.14.08` directory. This directory is referred to as the data collector home directory (*dc_home*) in the following steps.

  c) Navigate to the Liberty server home directory. For example, `/opt/ibm/wlp/usr/servers/defaultServer`.

  d) Edit the `jvm.options` file by adding the following parameters. If the `jvm.options` file does not exist, create it with a text editor.

  ```
  -agentlib:am_ibm_16=server_name
  -Xbootclasspath/p:dc_home/toolkit/lib/bcm-bootstrap.jar
  -Djava.security.policy=dc_home/itcamdc/etc/datacollector.policy
  -Dliberty.home=liberty_home
  -Xhealthcenter:level=inprocess
  -Xgc:allocationSamplingGranularity=10000
  -verbosegc
  -Xverbosegclog:absolute_path_to_log_file,1,10000
  ```

  When you add the entries, take note of the following things:

  – Each entry must be on a single line.

  – Replace *server_name* with the name of the Liberty server.

  – Replace *dc_home* with the home directory of the data collector. For example, `/opt/ibm/apm/liberty_dc/.gdc/7.3.0.14.08`.

  – Replace *liberty_home* with the root of Liberty installation directory. For example, `/opt/ibm/wlp`.

  – If the Liberty server is working with heavy workload, add the `-Xmx` parameter to allocate extra 512M heap size for the data collector. For example, `-Xmx1024M`.

- The `-Xhealthcenter:level=inprocess` and `-Xgc:allocationSamplingGranularity=10000` lines are optional. Add the two lines only if you want to enable memory allocation collection, which is disabled by default. Enabling memory allocation collection is required for the Memory Analysis dashboard to contain data.
- The `-Xverbosegclog:`*`absolute_path_to_log_file`*`,1,10000` line is optional, which specifies the path to the redirected garbage collection log file. If not specified, logs are written into one file and rotates every 10000 allocation failures. The original `stdout` or `stderr` file (`console.log`) might be very large as the server runs. Add this line if you want to save the garbage collection output log files to another directory and limit the log file number and size. If the specified path is invalid, this line takes no effect and the garbage collection log file remains the `stdout` or `stderr` file.

e) Open the `server.env` file in the same directory and add the following path to the environment entry. If the `server.env` file does not exist, create it with a text editor.

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/lib:dc_home/toolkit/lib/lx8266:dc_home/
toolkit/lib/lx8266/ttapi
```

When you add the entries, take note of the following things:

- Each entry must be on a single line.
- Replace *dc_home* with the home directory of the data collector. For example, `/opt/ibm/apm/liberty_dc/.gdc/7.3.0.14.08`.

f) Modify the `server.xml` in the same directory to enable the monitoring feature by adding the following line to the `<featureManager>` section:

```
<feature>monitor-1.0</feature>
```

g) Restart the Liberty server.

- To configure the data collector by responding to prompts, use the configuration script that is provided in the data collector packages.

a) Run the following command to extract files from the data collector package.

```
tar -xzf IBM_Data_Collectors_Install.tgz
```

The `liberty_datacollector_8.1.4.0.tgz` package is included in the extracted directory.

b) Extract files from the `liberty_datacollector_8.1.4.0.tgz` package with the following command.

```
tar -xzf liberty_datacollector_8.1.4.0.tgz
```

For example,

```
cd /opt/ibm
tar -xzf liberty_datacollector_8.1.4.0.tgz
```

The extracted data collector files are in the `liberty_dc` directory.

c) Change to the `liberty_dc/.gdc/7.3.0.14.08/bin` directory and start the configuration script by running the following command:

```
./config_liberty_dc.sh
```

d) When prompted, enter the root of your Liberty installation directory or accept the default. For example, `/opt/ibm/wlp`.

e) When prompted, enter the home of JVM that is used by the application server or accept the default. For example, `/opt/ibm/java`.

f) The configuration program can automatically discover and list the application servers that are not configured within the specified directory. Enter the number that corresponds to the Liberty server that you want to configure. To select more than one server, separate the numbers by space or enter * to select all.

g) After the configuration program finishes updating files for all Liberty servers, manually update the JVM heap size to allocate extra 512M heap for the data collector.

h) Restart the servers for configuration to take effect.

Results

The data collector is configured and is connected to the Cloud APM server. Resource monitoring, transaction tracking, and diagnostic data is enabled. However, heap collection and memory allocation collection are disabled. You can enable them with the data collector properties files if you need the data in the Heap Dump and Memory Analysis dashboards.

What to do next

- To view the monitoring data for your Liberty servers, start the Cloud APM console. For instructions, see Starting the Cloud APM console. For information about using the Applications editor, see Managing applications.

  Remember: When you add the Liberty data collector instance on the Application Dashboard, select **Liberty Runtime** instead of **WebSphere Application Server** from the component list.
- For the Heap Dump dashboard and/or Memory Analysis dashboard to contain data, you also need to enable the data collector for heap snapshot collector and/or memory allocation collection, which can be done in the data collector `.properties` files. See "Enabling or disabling transaction tracking and diagnostic data collection" on page 875.
- If the key file or the Cloud APM server changes, reconnect the data collector to the Cloud APM server. For instructions, see "Reconnecting the data collector to the Cloud APM server" on page 185.

Unconfiguring the data collector for on-premises applications

If you do not need to monitor your Liberty servers or if you want to upgrade the data collector to a new version, you must unconfigure the data collector that you deployed to the Liberty server.

About this task

To unconfigure the data collector that is deployed to the Liberty server, roll back the changes that are made when you configure the data collector. You can choose to configure the data collector manually or with the provided `unconfig_liberty_dc` script.

Procedure

- To manually unconfigure the data collector, complete the following steps:

  a) Navigate to the Liberty server home directory. For example, `/opt/ibm/wlp/usr/servers/defaultServer`.

  b) Edit the `jvm.options` file to remove the following parameters if any.

```
-agentlib:am_ibm_16=server_name
-Xbootclasspath/p:dc_home/toolkit/lib/bcm-bootstrap.jar
-Djava.security.policy=dc_home/itcamdc/etc/datacollector.policy
-Dliberty.home=liberty_home
-Xhealthcenter:level=inprocess
-Xgc:allocationSamplingGranularity=10000
-verbosegc
-Xverbosegclog:absolute_path_to_log_file,1,10000
```

  c) Edit the `server.env` file in the same directory to remove the following value for the **`LD_LIBRARY_PATH`**

```
/lib:dc_home/toolkit/lib/lx8266:dc_home/toolkit/lib/lx8266/ttapi
```

  where *dc_home* is the home directory of the data collector. For example, `/opt/ibm/apm/liberty_dc/.gdc/7.3.0.14.08`.

d) Edit the `server.xml` file in the same directory to remove `<feature>monitor-1.0</feature>` from the `<featureManager>` section.

e) Restart the Liberty server.

- To unconfigure the data collector with the `unconfig_liberty_dc.sh` script, complete the following steps:

  a) Change to the *dc_home*`/bin` directory. For example, `/opt/ibm/apm/liberty_dc/.gdc/7.3.0.14.08/bin`.

  b) Start the unconfiguration script by running the following command:

  ```
  ./unconfig_liberty_dc.sh
  ```

  c) When prompted, enter the root of your Liberty installation directory or accept the default. For example, `/opt/ibm/wlp`.

  d) The unconfiguration program can automatically discover and list the application servers that are configured within the specified directory. Enter the number that corresponds to the Liberty server that you want to unconfigure. To select more than one server, separate the numbers by space or enter `*` to select all.

  e) After the unconfiguration program finishes updating files for all Liberty servers, restart the servers for changes to take effect.

What to do next

After you unconfigure the data collector, the Cloud APM console continues to display the data collector in any applications that you added the data collector to. The Cloud APM console will show that no data is available for the application and will not indicate that the data collector is offline. For information about how to remove the data collector from applications and from resource groups, see "Removing data collectors from Cloud APM console" on page 188.

You can also delete the home directory of the data collector if you do not need it anymore.

Configuring the Liberty data collector in IBM Cloud environment (Liberty V18.* and older versions)

To monitor a Liberty profile running in the IBM Cloud environment, you must download the data collector package from IBM Marketplace, deploy the data collector to your local application files, and then push the updates to IBM Cloud.

Before you begin

It is assumed that the Liberty application is pushed to the IBM Cloud environment by using the Cloud Foundry commands. The `manifest.yml` file and the Liberty server home directory (which contains the `server.xml` file) already exist.

If your Liberty application is deployed as a WAR file, you must modify some local files to update your application by pushing a local directory that contains both the WAR file and the data collector files. An example is provided here to explain how to get a local Liberty server home directory if you only have a WAR file.

1. Issue the following command to run the Liberty application locally:

   ```
   mvn install liberty:run-server
   ```

   In the directory that contains the Liberty WAR file, a sub-directory, `/liberty/wlp/usr/servers/defaultServer`, is created. This directory can serve as the Liberty server home directory in the following procedure.
2. From the root directory that contains the Liberty WAR file, copy the entire *application_name*-SNAPSHOT folder to the `/liberty/wlp/usr/servers/defaultServer` directory.
3. In the `/liberty/wlp/usr/servers/defaultServer` directory, edit the `bootstrap.properties` file to modify the **appLocation** path. The **appLocation** path must be set to the relative path to the application directory in IBM Cloud.

4. Remove the `logs` and `workarea` folders. They do not need to be pushed to IBM Cloud.
5. Modify the **`path`** value in the `manifest.yml` file to point to the `defaultServer` directory.

   For example, `path: target/liberty/wlp/usr/servers/defaultServer`.

Procedure

Complete the following steps to configure the Liberty data collector:

1. Download the data collector package named `IBM_Data_Collectors_Install.tgz` from IBM Marketplace. For detailed instructions, see "Downloading your agents and data collectors" on page 105.
2. Run the following command to extract files from the data collector package.

   ```
   tar -xzf IBM_Data_Collectors_Install.tgz
   ```

   The `liberty_datacollector_8.1.4.0.tgz` package is included in the extracted directory.
3. Extract files from the `liberty_datacollector_8.1.4.0.tgz` package to a temporary directory.

   ```
   tar -xzf liberty_datacollector_8.1.4.0.tgz
   ```

   For example,

   ```
   cd /root/tmp
   tar -xzf liberty_datacollector_8.1.4.0.tgz
   ```

   You can find the extracted files in the `liberty_dc` directory within the temporary directory.
4. Copy the `.gdc` directory from the `liberty_dc` directory to the home directory of your Liberty server where the `server.xml` file is stored. The Liberty server home directory is referred to as *`liberty_server_home`* in the following steps.

   ```
   cp -rf temp_dir/liberty_dc/.gdc liberty_server_home
   ```

   For example,

   ```
   cp -rf /root/tmp/liberty_dc/.gdc /opt/liberty855/wlp/usr/servers/defaultServer/
   ```

5. Copy or merge the contents of the `jvm.options` and `server.env` files from the `liberty_dc/etc` directory to *`liberty_server_home`* directory.

   - If the `jvm.options` and `server.env` files do not exist in *`liberty_server_home`* directory, copy the two files from*`temp_dir`*`/liberty_dc/etc` to *`liberty_server_home`*.

     ```
     cp temp_dir/liberty_dc/etc/jvm.option liberty_server_home
     cp temp_dir/liberty_dc/etc/server.env liberty_server_home
     ```

   - If the `jvm.options` or `server.env` file exists in *`liberty_server_home`* directory, merge the contents with the ones from *`temp_dir`*`/liberty_dc/etc` directory.
6. If your IBM Cloud applications cannot directly connect to the Cloud APM server due to the network or firewall settings, configure the data collector to send traffic through a forward proxy. To do it, edit the `jvm.options` file in one of the following ways:

   - If authentication is not required, add the following lines to the file:

     ```
     -Dhttp.proxyHost=http_proxy_host
     -Dhttp.proxyPort=http_proxy_port
     -Dhttps.proxyHost=http_proxy_host
     -Dhttps.proxyPort=http_proxy_port
     -Djava.net.useSystemProxies=true
     ```

   - If a user name and password are required to access the forward proxy server, add the following lines to the file:

     ```
     -Dhttp.proxyHost=http_proxy_host
     -Dhttp.proxyPort=http_proxy_port
     ```

```
-Dhttp.proxyUser=http_proxy_user
-Dhttp.proxyPassword=http_proxy_password
-Dhttps.proxyHost=http_proxy_host
-Dhttps.proxyPort=http_proxy_port
-Dhttps.proxyUser=http_proxy_user
-Dhttps.proxyPassword=http_proxy_password
-Djava.net.useSystemProxies=true
```

7. Modify the `server.xml` file within the Liberty server home directory to enable the monitoring feature by adding the following line to the `<featureManager>` section:

   ```
   <featureManager>
                <feature>monitor-1.0</feature>
   </featureManager>
   ```

8. Modify the `manifest.yml` file of your Liberty application to allocate additional 512M memory.
9. Open a command prompt, change to the local directory that contains the `manifest.yml` file for the Liberty server. For example, `/opt/liberty855/`.
10. Log in to IBM Cloud and update the Liberty profile with the **`cf push`** command.

Results

The data collector is configured and is connected to the Cloud APM server. Resource monitoring, transaction tracking, and diagnostic data is enabled. However, heap collection and memory allocation collection are disabled. You can enable them with the data collector properties files if you need the data in the Heap Dump and Memory Analysis dashboards.

What to do next

- To view the monitoring data for your IBM Cloud application, start the Cloud APM console. For instructions, see Starting the Cloud APM console. For information about using the Applications editor, see Managing applications.

  Remember: When you want to add the Liberty data collector instance on the Application Dashboard, select **Liberty Runtime** instead of **WebSphere Application Server** from the component list.
- For the Heap Dump dashboard and/or Memory Analysis dashboard to contain data, you also need to enable the data collector for heap snapshot collector and/or memory allocation collection, which can be done in the data collector `.properties` files. See "Customizing data collector with properties files" on page 472.
- If the key file or the Cloud APM server changes, reconnect the data collector to the Cloud APM server. For instructions, see "Reconnecting the data collector to the Cloud APM server" on page 185.

Environment variables for customizing the Liberty data collector

To customize the Liberty data collector for the IBM Cloud applications, use the IBM Cloud UI to add the environment variables that are supported by the data collector.

Tip: To add environment variables on the IBM Cloud UI, first log in to the IBM Cloud UI and click your application, and then click **Runtime** > **Environment variable**. In the **user-defined** section, add the environment variables.

- Use the variables that are listed in Table 169 on page 471 to configure connection between the Liberty data collector and the Cloud APM server.
- Use the variable that is listed in Table 170 on page 471 to enable or disable method trace for your IBM Cloud applications.
- After method tracing is enabled, use the variables that are listed in Table 171 on page 472 to specify thresholds for different types of requests, so that different levels of monitoring data can be collected.

Remember: After you add or modify the environment variable, restart your application for the changes to take effect.

Table 169. Environment variables for server connections		
Variable name	**Possible values**	**Description**
APM_BM_GATEWAY_URL	- `https://` *`server_ip_or_hostname`*`:443` - `http://` *`server_ip_or_hostname`*`:80`	The URL of the target Cloud APM server gateway.
APM_KEYFILE_PSWD	Encrypted password of the key file	The encrypted key file password that is paired with the key file. If you are a Linux user, you can use the `echo -n` *`<keyfile password>`* `| base64` command to encrypt your password. **Remember:** Set this variable only when you configured the *APM_BM_GATEWAY_URL* variable to use HTTPS.
APM_KEYFILE_URL	`http://` *`hosted_http_server`*`:`*`port`*`/` `keyfile.jks`	The URL to download the key file. **Remember:** Set this variable only when you configured the *APM_BM_GATEWAY_URL* variable to use HTTPS.

Table 170. Environment variable for method trace		
Variable name	**Possible values**	**Description**
METHOD_TRACE_ENABLE	- `true` - `false`	Use this variable to enable or disable method tracing. The value of `true` enables method tracing. The default value is `false`.

After method trace is enabled, you can configure thresholds for different types of requests to customize method trace. The following thresholds, which trigger the collection of different levels of monitoring data, can be configured for each request type:

Primary thresholds
If you configure the primary threshold for a request type, the timing information of this type of requests is captured, such as CPU time and response time for that request type. As a result, when a request takes more time to complete than the time that is specified for the primary threshold, the timing of the request is captured.

Secondary thresholds
If you configure the secondary threshold for a request type, deep context data is captured, such as stack traces and SQL for database requests. The context data that is captured differs based on the request type. When a request takes more time to complete than the time that is specified for the secondary threshold, its context data is captured.

The environment variable for different request thresholds is named as *`<request_type>_<threshold level>`*. For example, to configure a primary threshold for the JMS request, add the `JMS_PRIMARY` variable and set its value.

Table 171 on page 472 lists the corresponding environment variables that you can add for different request types. The values are in milliseconds.

Table 171. Environment variables for different request thresholds

Variable name	Default value (in milliseconds)
SERVLET_PRIMARY	20
SERVLET_SECONDARY	50
JDBC_PRIMARY	20
JDBC_SECONDARY	50
JNDI_PRIMARY	20
JNDI_SECONDARY	50
EJB_PRIMARY	20
EJB_SECONDARY	50
WEBSERVICES_PRIMARY	20
WEBSERVICES_SECONDARY	50
APP_METHODS_PRIMARY (application methods – non J2EE)	50
APP_METHODS _SECONDARY	1000
JCA_PRIMARY	50
JCA_SECONDARY	80
JMS_PRIMARY	40
JMS_SECONDARY	70

Customizing data collector with properties files

By default, transaction tracking and method tracing are enabled for the data collector. Heap snapshot collection and memory allocation collection are disabled. You can customize the data collection or the intervals at which the diagnostic data is collected by editing the `.properties` files of the data collector.

About this task

The properties files of the data collector are in the *dc_home* directory, for example, `/opt/liberty855/wlp/usr/servers/defaultServer/.gdc/7.3.0.14.08`. Use different properties to customize the data collector for the following purposes:

- Enable or disable transaction tracking.
- Enable or disable heap snapshot collection.
- Specify the interval at which the data collector takes snapshot of heap dump.
- Enable or disable memory allocation monitoring.
- Specify the interval at which the data collector collects memory allocation information.
- Enable or disable method tracing.

Remember: After you modify the `.properties` files, use the **`cf push`** command to push the updates to the IBM Cloud environment.

Procedure

- To enable or disable transaction tracking, set the **`com.ibm.tivoli.itcam.dc.bluemix.transaction.enabled`** property in the following file to `true` or `false`:

dc_home/ldc/etc/ldc.properties

If transaction tracking is enabled, you can monitor IBM Java application stack in the topologies.

- To enable or disable heap snapshot collection, set the **com.ibm.tivoli.itcam.hc.send.heap.enable** and **com.ibm.tivoli.itcam.hc.snapshot.automatic.enable** properties in the following file to true or false.

  dc_home/healthcenter/etc/hc.properties

  If heap snapshot collection is enabled, the data collector can take heap snapshot at specified intervals. Heap dump information can be displayed in the Heap Dump dashboard.

- To change the interval at which heap snapshot is taken by the data collector, set the **com.ibm.tivoli.itcam.hc.snapshot.automatic.interval** property in the same file to a positive integer. The unit of the interval is minute and the default is 360.

  dc_home/healthcenter/etc/hc.properties

- To enable or disable memory allocation collection, set the **com.ibm.tivoli.itcam.hc.events.collection.automatic.enable** property in the following file to true or false.

  dc_home/healthcenter/etc/hc.properties

  Remember: To enable memory allocation collection, you also need to ensure the following two lines are added to the jvm.options file of the Liberty server.

  ```
  -Xhealthcenter:level=inprocess
  -Xgc:allocationSamplingGranularity=10000
  ```

  After memory allocation collection is enabled, data is available in the Memory Analysis dashboard.

- To specify the interval at which memory allocation information is collected, set the **com.ibm.tivoli.itcam.hc.events.collection.automatic.interval** property in the same file to a positive integer. The unit of the interval is minute and the default is 15.

  dc_home/healthcenter/etc/hc.properties

- To enable or disable method tracing, set the **dfe.enable.methoddata** property in the following file to true or false:

  dc_home/gdc/etc/gdc_dfe.properties

What to do next

- After method tracing is enabled, you can set thresholds for different types of requests by using the environment variables, so that different levels of monitoring data can be collected for different requests. For applicable environment variables, see Table 171 on page 472.
- If you disabled memory allocation collection, remember to remove the following lines from the jvm.options file of the Liberty server:

  ```
  -Xhealthcenter:level=inprocess
  -Xgc:allocationSamplingGranularity=10000
  ```

Unconfiguring the data collector for IBM Cloud applications

If you do not need to monitor your Liberty profiles in IBM Cloud environment or if you want to upgrade the data collector to a new version, you must unconfigure the data collector that you previously deployed.

About this task

To unconfigure the data collector for your Liberty profile in IBM Cloud environment, roll back the changes that are made in jvm.options, server.env and server.xml files, and then update the Liberty profile in IBM Cloud with the **cf push** command.

Procedure

1. In your local directory of the Liberty profile, modify the `jvm.options` file to remove the following parameters. You can delete the file if it is empty after the change.

   ```
   -agentlib:am_ibm_16=defaultServer
   -Xbootclasspath/p:../../../../.gdc/7.3.0.14.08/toolkit/lib/bcm-bootstrap.jar
   -Djava.security.policy=../../../../.gdc/7.3.0.14.08/itcamdc/etc/datacollector
   .policy
   -Dliberty.home=/home/vcap/app/.liberty
   -Xhealthcenter:level=inprocess
   -Xgc:allocationSamplingGranularity=10000
   -verbosegc
   -Xverbosegclog:/home/vcap/app/wlp/usr/servers/defaultServer/logs/gc.log,1,10000
   ```

2. In the `server.env` file, remove the following value for the **`LD_LIBRARY_PATH`** environment variable. You can delete the file if it is empty after the change.

   ```
   LD_LIBRARY_PATH=:/lib:../../../../.gdc/7.3.0.14.08/toolkit/lib/lx8266
   :../../../../.gdc/7.3.0.14.08/toolkit/lib/lx8266/ttapi
   ```

3. Modify the `server.xml` file to remove the monitor-1.0 feature by removing the following line to the `<featureManager>` section

   ```
   <feature>monitor-1.0</feature>
   ```

4. Delete the `.gdc` directory within the Liberty home directory.
5. Open a command prompt, change to the directory that contains the `manifest.yml` file of the Liberty server.
6. Log in to IBM Cloud and update the Liberty profile with the **`cf push`** command.

What to do next

After you unconfigure the data collector, the Cloud APM console continues to display the data collector in any applications that you added the data collector to. The Cloud APM console will show that no data is available for the application and will not indicate that the data collector is offline. For information about how to remove the data collector from applications and from resource groups, see "Removing data collectors from Cloud APM console" on page 188.

Configuring Linux KVM monitoring

You must configure the Monitoring Agent for Linux KVM to collect data of the Red Hat Enterprise Virtualization Hypervisor (RHEVH) and Red Hat Enterprise Virtualization Manager (RHEVM) servers. After you install the agent on a server or a virtual machine, you must create the first instance, and start the agent manually.

Before you begin

Review hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Linux KVM agent.

About this task

The Linux KVM agent is a multi-instance and multi-connection agent. Multi-instance means that you can create multiple instances and each instance can make multiple connections to one or more RHEVM or RHEVH servers.

Remember: Use different instances to monitor RHEVM or the RHEVH servers.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the "Change history" on page 54.

You can use the same configuration script to configure instances for the RHEVH and the RHEVM servers:

- To configure a connection to the RHEVM server, complete the steps that are mentioned in the "Configuring a connection to the RHEVM server" topic.
- To configure a connection to the RHEVH server, complete the steps that are mentioned in the "Configuring a connection to the RHEVH server" topic.

Creating a user and granting required permissions

Before you configure the Linux KVM agent, you must create a user and grant required permissions to the user to monitor the RHEVM and RHEVH servers.

Procedure

1. Open the **Red Hat Enterprise Virtualization Manager Web Administration** portal.
2. Click **Configure**.
3. In the **Configuration** window, select **Roles**.
   a) To create a role, click **New**.
   b) In the **New Role** window, add the name of the role and select **Admin** as the account type.
   c) Ensure that the check boxes in the **Check boxes to Allow Action** pane are not selected, and click **OK**.
4. In the **Configuration** window, select **System Permission**.
   a) To grant a user permission, click **Add**.
   b) In the **Add System Permission to User** window, select the user to whom you want to grant the permission.
   c) From the **Assign role to user** list, select the role that you created and click **OK**.

What to do next

Complete the agent configuration:

- “Configuring a connection to the RHEVH server” on page 481
- “Configuring a connection to the RHEVM server” on page 479

Configuring protocols

The agent uses different protocols to connect to the RHEVH server. You can configure any of these protocols: SSH, TLS, or TCP.

About this task

The Linux KVM agent remotely connects to each hypervisor by using the **virsh** tool that manages your QEMU-KVM virtual machines, and collects metrics. The `libvirt` API in the agent environment uses several different remote transport protocols. For the list of supported protocols, see the Remote support page.

Configuring the SSH protocol

You can configure the SSH protocol to remotely monitor a host.

About this task

Assumption: The Linux KVM agent is installed on host A. You want to remotely monitor the hypervisor on host B.

Procedure

1. Log in to host A with the same user ID that runs the Linux KVM agent process, for example, the root user ID.

   Tip: Ensure that you know the ID on host B that accepts the SSH connection and the root user ID on host A.

2. Generate the **id_rsa** and **id_rsa.pub** keys on host A by using the *ssh-keygen* utility.

   The keys are saved at the following location: `~/.ssh: $ ssh-keygen -t rsa`.
3. Copy the authorized keys from host B:

   ```
   $ scp Id on host B@name or IP address of host B:~/.ssh/authorized_keys
   ~/.ssh/authorized_keys_from_B
   ```
4. Append the public key for host A to the end of the authorized keys for host B:

   ```
   cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys_from_B
   ```
5. Copy the authorized keys back to host B:

   ```
   $ scp ~/.ssh/authorized_keys_from_B Id on host B@name or IP address of host
   B:~/.ssh/authorizede_keys
   ```

   Remember: If you are monitoring multiple hosts, repeat steps "3" on page 476, "4" on page 476, and "5" on page 476 for each host.
6. Remove the authorized keys that you copied on host B:

   ```
   ~/.ssh/authorized_keys_from_B
   ```
7. Add the following command to the **~/.bash_** profile of the current ID on host A:

   ```
   $ eval `ssh-agent`
   ```

   Remember: Ensure that you use the single back quotation mark (`) that is located under the tilde (~) on US keyboards, rather than the single quotation mark (').
8. Add the identity to host A and enter the password that you used when you created the ID:

   ```
   $ ssh-add ~/.ssh/id_rsa
   ```
9. Run the following command if you receive the `Could not open a connection to your authentication agent` message:

   ```
   exec ssh-agent bash
   ```

   Tip: You can replace the bash with the shell that you are using and then run the following command again:

   ```
   $ ssh-add ~/.ssh/id_rsa
   ```
10. Test the SSH protocol to ensure that it connects from host A to host B without entering the SSH password:

    Tip: If you are monitoring multiple hosts, use the following command to test the connection for each host:

    ```
    $ ssh Id on host B@name or IP address of host B
    ```
11. To verify the connection, run the following command:

    ```
    virsh -c qemu+ssh://Id on host B@name or IP address of host B:port/system
    ```

    If you did not change the default SSH port, omit the **:port** section of the command.

    Important: If the **virsh** command succeeds, the Linux KVM agent connects to the hypervisor.
12. You must restart host A before you restart the Linux KVM agent on host A. To restart, run the **ssh-add** command again and specify the password each time.

    Tip: You can use SSH keychains to avoid reentering the password.

Configuring the TLS protocol

You can configure the TLS protocol to remotely monitor a host.

About this task

Assumption: The Linux KVM agent is installed on host A. You want to remotely monitor the hypervisor on host B.

Procedure

1. To create a certificate authority (CA) key and a certificate in your hypervisor, complete the following steps:
   a) Log in to host B.
   b) Create a temporary directory and change the path to this temporary directory:

      ```
      mkdir cert_files
      ```

      ```
      cd cert_files
      ```

   c) Create a 2048-bit RSA key:

      ```
      openssl genrsa -out cakey.pem 2048
      ```

   d) Create a self-signed certificate to your local CA:

      ```
      openssl req -new -x509 -days 1095 -key cakey.pem -out \
      cacert.pem -sha256 -subj "/C=US/L=Austin/O=IBM/CN=my CA"
      ```

   e) Check your CA certificate:

      ```
      openssl x509 -noout -text -in cacert.pem
      ```

2. To create the client and server keys and certificates in your hypervisor, complete the following steps:
   a) Create the keys:

      ```
      openssl genrsa -out serverkey.pem 2048
      ```

      ```
      openssl genrsa -out clientkey.pem 2048
      ```

   b) Create a certificate signing request for the server:

      Remember: Change the kvmhost.company.org address, which is used in the server certificate request, to the fully qualified domain name of your hypervisor host.

      ```
      openssl req -new -key serverkey.pem -out serverkey.csr \
      -subj "/C=US/O=IBM/CN=kvmhost.company.org"
      ```

   c) Create a certificate signing request for the client:

      ```
      openssl req -new -key clientkey.pem -out clientkey.csr \
      -subj "/C=US/O=IBM/OU=virtualization/CN=root"
      ```

   d) Create client and server certificates:

      ```
      openssl x509 -req -days 365 -in clientkey.csr -CA cacert.pem \
      -CAkey cakey.pem -set_serial 1 -out clientcert.pem
      ```

      ```
      openssl x509 -req -days 365 -in serverkey.csr -CA cacert.pem \
      -CAkey cakey.pem -set_serial 94345 -out servercert.pem
      ```

   e) Check the keys:

      ```
      openssl rsa -noout -text -in clientkey.pem
      ```

      ```
      openssl rsa -noout -text -in serverkey.pem
      ```

   f) Check the certificates:

      ```
      openssl x509 -noout -text -in clientcert.pem
      ```

      ```
      openssl x509 -noout -text -in servercert.pem
      ```

3. To distribute the keys and certificates to the host server, complete the following steps:
   a) Copy the CA certificate `cacert.pem` file to this directory: `/etc/pki/CA`

      ```
      cp cacert.pem /etc/pki/CA/cacert.pem
      ```

   b) Create the `/etc/pki/libvirt` directory, and copy the `servercert.pem` server certificate file to the `/etc/pki/libvirt` directory. Ensure that only the root user can access the private key.

      ```
      mkdir /etc/pki/libvirt
      ```

```
cp servercert.pem /etc/pki/libvirt/.
```

```
chmod -R o-rwx /etc/pki/libvirt
```

Remember: If the keys or certificates are named incorrectly or copied to the wrong directories, the authorization fails.

c) Create the `/etc/pki/libvirt/private` directory and copy the `serverkey.pem` server key file to the `/etc/pki/libvirt/private` directory. Ensure that only the root user can access the private key.

```
mkdir /etc/pki/libvirt/private
```

```
cp serverkey.pem /etc/pki/libvirt/private/.
```

```
chmod -R o-rwx /etc/pki/libvirt/private
```

Remember: If the keys or certificates are named incorrectly or copied to the wrong directories, the authorization fails.

d) Verify that the files are correctly placed:

```
find /etc/pki/CA/*|xargs ls -l
```

```
ls -lR /etc/pki/libvirt
```

```
ls -lR /etc/pki/libvirt/private
```

Remember: If the keys or certificates are named incorrectly or copied to the wrong directories, the authorization fails.

4. To distribute keys and certificates to clients or management stations, complete the following steps:

a) Log in to host A.

b) Copy the CA certificate `cacert.pem` from the host to the `/etc/pki/CA` directory in host A without changing the file name.

```
scp kvmhost.company.org:/tmp/cacert.pem /etc/pki/CA/
```

c) Copy the client certificate `clientcert.pem` file to the `/etc/pki/libvirt` directory from host B. Use the default file names and make sure that only the root user is able to access the private key.

```
mkdir /etc/pki/libvirt/
```

```
scp kvmhost.company.org:/tmp/clientcert.pem /etc/pki/libvirt/.
```

```
chmod -R o-rwx /etc/pki/libvirt
```

Remember: If the keys or certificates are named incorrectly or copied to the wrong directories, the authorization fails.

d) Copy the client key `clientkey.pem` to the `/etc/pki/libvirt/private` directory from the host. Use the default file names and ensure that only the root user can access the private key.

```
mkdir /etc/pki/libvirt/private
```

```
scp kvmhost.company.org:/tmp/clientkey.pem /etc/pki/libvirt/private/.
```

```
chmod -R o-rwx /etc/pki/libvirt/private
```

Remember: If the keys or certificates are named incorrectly or copied to the wrong directories, the authorization fails.

e) Verify that the files are correctly placed:

```
ls -lR /etc/pki/libvirt
```

```
ls -lR /etc/pki/libvirt/private
```

5. To edit the `libvirtd` daemon configuration, complete the following steps:

a) Log in to host B.

b) Make a copy of the `/etc/sysconfig/libvirtd` file and the `/etc/libvirt/libvirtd.conf` file.

c) Edit the /etc/sysconfig/libvirtd file and ensure that the **--listen** parameter is passed to the libvirtd daemon. This step ensures that the libvirtd daemon is listening to network connections.

d) Edit the /etc/libvirt/libvirtd.conf file and configure a set of allowed subjects with the **tls_allowed_dn_list** directive in the libvirtd.conf file.

Important: The fields in the subject must be in the same order that you used to create the certificate.

e) Restart the libvirtd daemon service for changes to take effect:

```
/etc/init.d/libvirtd restart
```

6. To change the firewall configuration, access the security level configuration and add TCP port 16514 as a trusted port.
7. To verify that the remote management is working, run the following command on host A:

```
virsh -c qemu+tls://kvmhost.company.org/system list --all
```

Configuring the TCP protocol

Use the TCP protocol only for testing.

About this task

Assumption: The Linux KVM agent is installed on host A. You want to remotely monitor the hypervisor on host B.

Procedure

1. Log in to host B.
2. Edit the /etc/libvirt/libvirtd.conf file and ensure that the **listen_tcp** parameter is enabled, and the value of the **tcp_port** parameter is set to the default value of 16509.
3. Edit the /etc/libvirt/libvirtd.conf file to set the **auth_tcp** parameter to "none". This step instructs TCP not to authenticate the connection.
4. Restart the **libvirt** daemon on host B in listening mode by running it with the **--listen** flag or by editing the /etc/sysconfig/libvirtd file and uncommenting the LIBVIRTD_ARGS="--listen" line.
5. To verify the connection, run the following command:

```
virsh -c qemu+tcp://kvmhost.company.org:port/system
```

If you did not change the default TCP port, omit the **:port** section of the command.

Important: If the **virsh** command succeeds, the Linux KVM agent connects to the hypervisor.

What to do next

Configure the agent by completing the steps that are described in "Configuring a connection to the RHEVH server" on page 481.

Configuring a connection to the RHEVM server

To configure a connection to the RHEVM server, you must run the script and respond to prompts.

Before you begin

1. Download the security certificate that is available at the following path: http://[RHEVM-IP]/ovirt-engine/services/pki-resource?resource=cacertificate&format=X509-PEM-CA

   Depending on the browser, the certificate is either downloaded or imported into the browser's Keystore.

   - If the browser downloads the certificate: Save the file as rhvm.cer

- If the browser imports the certificate: Export it from the browser's certification options and save it as `rhvm.cer`

2. Use the *keytool* utility to import the security certificate file to generate a local keystore file:

   `keytool -import -alias ALIAS -file CERTIFICATE_FILE -keystore KEYSTORE_FILE`

   Example **`keytool -import -alias RHEVM36vmwt9 -file hjs495-vmw-t-9.cer -keystore RHEVM36KeyStore`**

   Where

   `ALIAS`
   A unique reference for each certificate that is added to the certificate truststore of the agent, for example, an appropriate alias for the certificate from *datasource.example.com* is *datasource*.

   `CERTIFICATE_FILE`
   The complete path and file name to the data source certificate that is being added to the truststore.

   `KEYSTORE_FILE`
   The name of the keystore file that you want to specify.

   Tip: The *keytool* utility is available with Java Runtime Environment (JRE). The keystore file is stored at the same location from where you run the command.

3. Ensure that the user, who connects to the RHEVM, is an administrator with the SuperUser role. Use can use an existing user ID with this role, or you can create a new user ID by completing the steps that are mentioned in "Creating a user and granting required permissions" on page 475.

Procedure

1. On the command line, run the following command:

   `install_dir/bin/linux_kvm-agent.sh config instance_name`

   Example **`/opt/ibm/apm/agent/bin/linux_kvm-agent.sh config instance_name`**

   Where

   instance_name
   The name that you want to give to the instance.

   install_dir
   The path where the agent is installed.

2. Respond to the prompts and specify values for the configuration parameters.

   For information about the configuration parameters, see "Configuration parameters to connect to the RHEVM server" on page 481.

3. Run the following command to start the agent:

   `install_dir/bin/linux_kvm-agent.sh start instance_name`

   Example **`/opt/ibm/apm/agent/bin/linux_kvm-agent.sh start instance_name`**

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you need help with troubleshooting, see the Troubleshooting section.

Configuring a connection to the RHEVH server

To configure a connection to the RHEVH server, you must run the script and respond to prompts.

Before you begin

- Ensure that the user, who connects to the RHEVM, is a root user. You can use an existing user ID or create a new user ID by completing the steps that are mentioned in "Creating a user and granting required permissions" on page 475.
- Configure the protocol that you want to use to connect to the RHEVH server by completing the steps that are described in "Configuring protocols" on page 475.

Procedure

1. On the command line, run the following command:

   `install_dir/bin/linux_kvm-agent.sh config instance_name`

   Example **`/opt/ibm/apm/agent/bin/linux_kvm-agent.sh config instance_name`**

   Where

   instance_name
   The name that you want to give to the instance.

   install_dir
   The path where the agent is installed.
2. Respond to the prompts and specify values for the configuration parameters.

   For information about the configuration parameters, see "Configuration parameters to connect to the RHEVH server" on page 483.
3. Run the following command to start the agent:

   `install_dir/bin/linux_kvm-agent.sh start instance_name`

   Example **`/opt/ibm/apm/agent/bin/linux_kvm-agent.sh start instance_name`**

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you need help with troubleshooting, see the Troubleshooting section.

Configuration parameters to connect to the RHEVM server

You can modify the default values of configuration parameters that are used for connecting the agent with the RHEVM server.

The following table contains detailed descriptions of the configuration parameters.

Table 172. Names and descriptions of the configuration parameters for connecting to the RHEVM server

Parameter name	Description	Mandatory field
Edit Monitoring Agent for Linux KVM settings	Indicates that you can begin editing the default values of the configuration parameters. Enter `1 (Yes)`, which is also the default value, to continue.	Yes
Maximum number of Data Provider Log Files	The maximum number of log files that the data provider creates before it overwrites the previous log files. The default value is 10.	Yes
Maximum Size in KB of Each Data Provider Log	The maximum size in KB that a data provider log file must reach before the data provider creates a new log file. The default value is 5190 KB.	Yes

Table 172. Names and descriptions of the configuration parameters for connecting to the RHEVM server (continued)

Parameter name	**Description**	**Mandatory field**
Level of Detail in Data Provider Log	The level of detail that can be included in the log file that the data provider creates. The default value is 4 (Info). The following values are valid: • 1 = Off: No messages are logged. • 2 = Severe: Only errors are logged. • 3 = Warning: All errors and messages that are logged at the Severe level and potential errors that might result in undesirable behavior. • 4 = Info: All errors and messages that are logged at the Warning level and high-level informational messages that describe the state of the data provider when it is processed. • 5 = Fine: All errors and messages that are logged at the Info level and low-level informational messages that describe the state of the data provider when it is processed. • 6 = Finer: All errors and messages that are logged at the Fine level plus highly-detailed informational messages, such as performance profiling information and debug data. Selecting this option can adversely affect the performance of the monitoring agent. This setting is intended only as a tool for problem determination along with IBM support staff. • 7 = Finest: All errors and messages that are logged at the Fine level and the most detailed informational messages that include low-level programming messages and data. Choosing this option might adversely affect the performance of the monitoring agent. This setting is intended only as a tool for problem determination along with IBM support staff. • 8 = All: All errors and messages are logged.	Yes
Edit Hypervisor settings	Indicates whether you want to edit the parameters for a connection to the RHEVH server. Enter `5 (Next)` because you are configuring a connection to the RHEVM server. The default value is `5 (Next)`.	Yes
Edit RHEVM Connection Details settings	Indicates whether you want to edit the parameters for a connection to the RHEVM server. Enter `1 (Add)` to continue. The default value is `5 (Next)`. **Important:** After you specify values for all the configuration parameters, you are again prompted to indicate whether you want to continue to edit the parameters. Enter `5 (Exit)`.	Yes
RHEVM ID	The unique user name, which you specify for the RHEVM that you connect to.	Yes
Host	The host name or IP address of the data source that is used to connect to the RHEVM server.	Yes
User	The user name of the data source with sufficient privileges to connect to the RHEVM server.	Yes

Table 172. Names and descriptions of the configuration parameters for connecting to the RHEVM server (continued)

Parameter name	**Description**	**Mandatory field**
Password	The password of the user name that you use to connect to the RHEVM server.	Yes
Re-type password	The same password that you specified in the **Password** field.	Yes
Port	The port number that is used to connect to the RHEVM server.	Yes
Domain	The domain to which the user belongs.	Yes
KeyStorePath	The file path and name of the local keystore file that you created by using the **`keytool`** command.	Yes

Configuration parameters to connect to the RHEVH server

You can modify the default values of configuration parameters that are used for connecting the agent with the RHEVH server.

The following table contains detailed descriptions of the configuration parameters.

Table 173. Names and descriptions of the configuration parameters for connecting to the hypervisor

Parameter name	**Description**	**Mandatory field**
Edit Monitoring Agent for Linux KVM settings	Indicates that you can begin editing the default values of the configuration parameters. Enter `1 (Yes)`, which is also the default value, to continue.	Yes
Maximum number of Data Provider Log Files	The maximum number of log files that the data provider creates before it overwrites the previous log files. The default value is 10.	Yes
Maximum Size in KB of Each Data Provider Log	The maximum size in KB that a data provider log file must reach before the data provider creates a new log file. The default value is 5190 KB.	Yes

Table 173. Names and descriptions of the configuration parameters for connecting to the hypervisor (continued)

Parameter name	Description	Mandatory field
Level of Detail in Data Provider Log	The level of detail that can be included in the log file that the data provider creates. The default value is 4 (Info). The following values are valid: • 1 = Off: No messages are logged. • 2 = Severe: Only errors are logged. • 3 = Warning: All errors and messages that are logged at the Severe level and potential errors that might result in undesirable behavior. • 4 = Info: All errors and messages that are logged at the Warning level and high-level informational messages that describe the state of the data provider when it is processed. • 5 = Fine: All errors and messages that are logged at the Info level and low-level informational messages that describe the state of the data provider when it is processed. • 6 = Finer: All errors and messages that are logged at the Fine level plus highly-detailed informational messages, such as performance profiling information and debug data. Selecting this option can adversely affect the performance of the monitoring agent. This setting is intended only as a tool for problem determination along with IBM support staff. • 7 = Finest: All errors and messages that are logged at the Fine level and the most detailed informational messages that include low-level programming messages and data. Selecting this option might adversely affect the performance of the monitoring agent. This setting is intended only as a tool for problem determination along with IBM support staff. • 8 = All: All errors and messages are logged.	Yes
Edit Hypervisor settings	Indicates whether you want to edit the parameters for a Hypervisor connection. Enter `1 (Add)`. The default value is 5 `(Next)`.	Yes
Hypervisor ID	The unique user name, which you specify for the RHEVH that you connect to.	Yes
Host	The host name or IP address of the data source that is used to connect to the RHEVH server.	Yes
User	A user name of the data source with sufficient privileges to connect to the RHEVM server.	Yes
Remote Transport	The protocol that is used by the local `libvirt` API to connect to remote `libvirt` APIs. The default value is 1. The following values are valid: • 1 = SSH • 2 = TLS • 3 = TCP (Unencrypted - not recommended for production use)	Yes

Table 173. Names and descriptions of the configuration parameters for connecting to the hypervisor (continued)

Parameter name	Description	Mandatory field
Port	The port that is used by the transport protocol to connect to the `libvirt` API. The default value is 22. **Important:** This port is only needed if the standard ports were changed (22 for SSH, 16514 for TLS, 16509 for TCP).	Yes
Domain	The domain to which the user belongs.	Yes
Connection Instance Type	Indicates whether the local `libvirt` API connects to the privileged system driver or the per-user unprivileged session driver. The default value is 1. The following values are valid: • 1 = system • 2 = session	Yes
Edit RHEVM Connection Details settings	Indicates whether you want to edit the parameters for a connection to the RHEVM server. Enter `1 (Add)` to continue. The default value is `5 (Next)`. **Important:** After you specify values for all the configuration parameters, you are again prompted to indicate whether you want to continue to edit the parameters. Enter `5 (Next)`.	Yes

Configuring MariaDB monitoring

You must configure the MariaDB agent so that the agent can collect data to monitor the availability and performance of MariaDB server resources. Refer the following prerequisites to configure the MariaDB agent for both remote and local monitoring.

Before you begin

Ensure that the system requirements for the MariaDB agent are met in your environment. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the MariaDB agent.

About this task

The MariaDB agent is a single instance agent. You must configure the agent manually after it is installed. You can configure the agent on Windows and Linux operating systems. The agent requires an instance name and the MariaDB server user credentials to configure it. The managed system name includes the instance name that you specify, for example, `*instance_name*:*host_name*:*pc*`, where *pc* is your two character product code. The managed system name can contain up to 32 characters. The instance name that you specify can contain up to 28 characters, excluding the length of your hostname. For example, if you specify `MariaDB` as your instance name, your managed system name is `MariaDB:hostname:MJ`.

Important: If you specify a long instance name, the managed system name is truncated and the agent code is not displayed.

Configuring the agent on Windows systems

You can configure the agent on Windows operating systems by using the IBM Cloud Application Performance Management window. After you update the configuration values, start the agent to apply the updated values.

Procedure

To configure the agent on Windows operating systems, complete the following steps:

1. Click **Start > All Programs > IBM Monitoring agents > IBM Performance Management**.
2. In the **IBM Performance Management** window, complete these steps:
   a) Double-click the **Monitoring Agent for MariaDB** template.
   b) In the **Monitoring Agent for MariaDB** window, specify an instance name and click **OK**.
3. In the **Monitoring Agent for MariaDB** window, complete these steps:
   a) In the **IP Address** field, enter the IP address of MariaDB server that you want to monitor remotely. If the agent is installed on a server to be monitored, retain the default value.
   b) In the **JDBC user name** field, enter the name of a MariaDB server user. The default value is root.
   c) In the **JDBC password** field, type the password of a JDBC user.
   d) In the **Confirm JDBC password** field, type the password again.
   e) In the **JDBC Jar File** field, click **Browse** and locate the directory that contains the MariaDB connector Java file and select it.
   f) Click **Next**.
   g) In the **JDBC port number** field, specify the port number of the JDBC server.

      The default port number is 3306.
   h) From the **Java trace level** list, select a trace level for Java.

      The default value is `Error`.
   i) Click **OK**.

      The instance is displayed in the **IBM Performance Management** window.
4. Right-click the **Monitoring Agent for MariaDB** instance, and click **Start**.

   Remember: To configure the agent again, complete these steps in the **IBM Performance Management** window:

   a. Stop the agent instance that you want to configure.
   b. Right-click the **Monitoring Agent for MariaDB** instance, and click **Reconfigure**.
   c. Repeat steps 3 and 4.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For more information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring the agent on Linux systems

You can run the configuration script and respond to prompts to configure the agent on Linux operating systems.

Procedure

To configure the agent on Linux operating systems, complete the following steps:

1. On command line, run the following command:

   ```
   install_dir/bin/mariadb-agent.sh config instance_name
   ```

   Where *instance_name* is the name you want to give to the instance, and *install_dir* is the installation directory for the MariaDB agent.
2. When you are prompted to enter a value for the following parameters, press **Enter** to accept the default value, or specify a different value and press **Enter**.

   - IP address
   - JDBC username
   - JDBC password
   - Retype JDBC password

- JDBC JAR file
- JDBC port number (Default port number is 3306.)
- Java trace level (Default value is `Error`.)

For more information about the configuration parameters, see "Configuring the agent by using the silent response file" on page 576.

3. Run the following command to start the agent:

```
install_dir/bin/mariadb-agent.sh start instance_name
```

What to do next
Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For more information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

You can use the silent response file to configure the MariaDB agent on Linux and Windows system. After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

To configure the agent by using the silent response file, complete the following steps:

Remember: This procedure assumes the following default path where the agent is installed:
Windows `C:\IBM\APM`
Linux `opt/ibm/apm/agent`
If the agent is installed at a different path, substitute the path in the instructions. Also, edit the **`AGENT_HOME`** parameter in the silent response file to specify the path where the agent is installed.

1. In a text editor, open the response file that is available at the following path:

   Linux *install_dir*`/samples/mariadb_silent_config.txt`

   Windows *install_dir*`\samples\mariadb_silent_config.txt`

   Where *install_dir* is the installation directory of the MariaDB agent
2. In the response file, specify a value for the following parameters:
   - For the **`Server Name`** parameter, specify the IP address of a MariaDB server that you want to monitor remotely. Otherwise, retain the default value as `localhost`.
   - For the **`JDBC user name`** parameter, retain the default username value of `root` or specify the name of a user with privileges to view the INFORMATION_SCHEMA tables.
   - For the **`JDBC password`** parameter, enter the JDBC user password.
   - For the **`JDBC Jar File`** parameter, retain the default path if this path to the MariaDB connector for the Java JAR file is correct. Otherwise, enter the correct path. The connector is available at the following default path:

     Linux `/usr/share/java/mariadb-connector-java.jar`
     Windows `C:\Program Files (x86)\MariaDB\mariadb-connector-java.jar`
   - For the **`JDBC port number`** parameter, retain the default port number of 3306 or specify a different port number.

- For the **Java trace level** parameter, retain the default value of `Error` or specify a different level according to the IBM support instructions.

3. Save and close the response file, and run the following command to update the agent configuration settings:

   Linux

   ```
   install_dir/bin/mariadb-agent.sh config instance_name install_dir/
   samples/mariadb_silent_config.txt
   ```

   Windows

   ```
   install_dir\BIN\mariadb-agent.bat config instance_name install_dir
   \samples\mariadb_silent_config.txt
   ```

   Where *instance_name* is the name that you want to give to the instance, and *install_dir* is the installation directory of MariaDB agent.

   Important: Ensure to include the absolute path to the silent response file. Otherwise, dashboards do not display agent data.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For more information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring Microsoft Active Directory monitoring

The Monitoring Agent for Microsoft Active Directory is automatically configured and started after installation.

Before you begin

Review the hardware and software prerequisites, see Software Product Compatibility Reports for Microsoft Active Directory agent

To view data for all attributes in the dashboard, complete the following tasks:

- "Running the Microsoft Active Directory agent as an administrator user" on page 488
- "Configuring local environment variables" on page 489

About this task

The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see "Change history" on page 54.

Running the Microsoft Active Directory agent as an administrator user

You must have administrative rights to run the Microsoft Active Directory agent.

About this task

All data sets are available to the users who are members of the Administrators group. In this task, you create a user, assign administrator rights to the user, and change the user account for the agent to this user.

Procedure

1. Click **Start** > **All Programs** > **Administrative Tools** > **Active Directory Users and Computers**.
2. To expand the domain where you want to create the user, click the plus sign (+) next to the name of a domain.
3. Right-click **Users**, and then click **New** > **User**.
4. To create a new user, open the **New Object - User** wizard.

   By default, a new user is a member of the Domain Users group.

5. Right-click the new user that is created in the Domain Users group, and click **Properties**. The **Username Properties** window is displayed. The username is the name of the new user.
6. In the **Username Properties** window, complete the following steps:
   a) Click the **Member of** tab. In the **Member of** area, add the Administrators group.
   b) Click **Apply**, and then click **OK**.
7. Click **Start** > **Run**, and then type `services.msc`.
8. In the **Services** window, complete the following steps:
   a) Right-click the **Monitoring Agent for Active Directory service**, and click **Properties**.
   b) In the **Monitoring Agent for Active Directory Properties** window, on the **Log On** tab, click **This Account**. Enter the user credentials.
   c) Click **Apply**, and then click **OK**.
9. Restart the agent service.

Configuring local environment variables

You must specify values for the environment variables to view the Sysvol replication data in the dashboard. Optionally, you can also update the cache interval value to enable or disable caching.

Procedure

1. In the **IBM Performance Management** window, from the **Actions** menu, click **Advanced** > **Edit ENV File**.
2. In the K3ZENV file, change the values of the following environment variables.

   `ADO_CACHE_INTERVAL`
   Determines whether to start or stop the caching and is used to set a value for the cache interval. Cache interval is the duration in seconds between two consecutive data collections. You can specify any positive integer value for the cache interval to start the caching. You can specify the zero value for the cache interval to stop the caching. By default, the caching is started, and the cache interval value is set to `1200`.

   `ADO_SYSVOL_FORCE_REPLICATION_FLAG`
   Determines whether the force replication that is initiated by the agent is enabled or disabled. The default value of this variable is TRUE. To disable force replication, change the value of this variable to `FALSE`.

   `ADO_SYSVOL_REPLICATION_TEST_INTERVAL`
   Determines the time interval in minutes between two Sysvol replication tests. The default value of this variable is `0` minutes. To complete the Sysvol replication test, ensure that the value of this variable is greater than zero.

   `ADO_SYSVOL_REPLICATION_TEST_VERIFICATION_INTERVAL`
   Determines the amount of time in minutes that the agent waits to verify the results of Sysvol replication after it completes the Sysvol replication test.

   The value of the **`ADO_SYSVOL_REPLICATION_TEST_INTERVAL`** variable must be greater than the value of the **`ADO_SYSVOL_REPLICATION_TEST_VERIFICATION_INTERVAL`** variable. You can use the following values for these variables:

   `ADO_SYSVOL_REPLICATION_TEST_INTERVAL`: `1440`
   `ADO_SYSVOL_REPLICATION_TEST_VERIFICATION_INTERVAL`: `30`

   After you assign valid values to the two environment variables, the Active Directory agent creates one file in the Sysvol shared folder of the managed system and initializes forced Sysvol replication. This forced replication is initialized from the managed system to the Sysvol shared folders of the Sysvol replication partners. After you verify the results of the replication test, the agent removes the files that are created and replicated from the managed system and Sysvol replication partners.
3. Optional: In the K3ZENV file, add the **`APM_ATTRIBUTES_ENABLE_COLLECTION`** environmental variable and set its value to `Yes` to view data for the following data sets in the **Attribute details** tab.

- Services
- Replication
- File Replication Service
- Moved or Deleted Org Unit
- LDAP
- Security Accounts Manager
- DFS
- Address Book
- Event Log
- Password Setting Objects

Remember: If you want to disable data collection for these data sets, set the value for the **`APM_ATTRIBUTES_ENABLE_COLLECTION`** environment variable to `No`.

4. Restart the Microsoft Active Directory agent.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Running Microsoft Active Directory agent as a non-administrator user

You can run the Log File agent as a non-administrator user.

About this task

You can run the monitoring agent for Active Directory as a non-administrator user; however, Trust Topology attributes and Sysvol Replication attributes might not be available. These attributes are available only to domain users.

To view the Trust Topology attributes, a non-administrator user must have the following registry permissions:

- Grant full access to the `HKEY_LOCAL_MACHINE\SOFTWARE\Candle directory`.
- Grant read access to the `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Perflib` directory.

To view the Sysvol Replication attributes, a non-administrator user must have full access to the `Sysvol` folder on all domain controllers in a domain.

Important: When Microsoft Active Directory agent is running as a non-administrator user, some services from the Services attribute group show values for Current State and Start Type attributes as Unknown on the APM User Interface.

The following table contains the attribute groups for the Active Directory agent that display data for domain users and performance monitoring users.

Table 174. Attribute groups for domain users and performance monitoring users	
User right	**Attribute group**
Domain users	• RID Pool Information • Services • Event Logs • DNS • DNS ADIntegrated Details • DNS ADIntegrated • DHCP • Trust • Group Policy Objects • Lost and Found Objects • Exchange Directory Service • Replication Conflict Objects • LDAP Attribute • Root Directory Server • Containers • Replication Partner • Domain Controller Availability • Replication Partner Latency • Forest Topology
Domain users and performance monitoring users	All attribute groups that are mentioned for the domain users and the following extra attribute groups: • Address Book • Replication • Directory Services • Knowledge Consistency Checker • Kerberos Key Distribution Center • Lightweight Directory Access Protocol • Local Security Authority • Name Service Provider • Security Accounts Manager • File Replication Service • Distributed File System Replication • DFS Replication Connections • DFS Replicated Folders • DFS Service Volume • Domain Controller Performance • Remote Access Server • Direct-Access Server • Netlogon Attributes

Note: Additionally, the following attribute groups display data for users who are members of the *Administrators* group:

- Active Directory Database Information
- Moved or Deleted Organizational Unit
- Password Setting Objects

For information, refer "Configuring Microsoft Active Directory monitoring" on page 488

Procedure

1. Click **Start > Programs > Administrative Tools > Active Directory Users and Computers**.
2. Expand the domain in which you want to create the user by clicking the plus sign (+) next to the name of a domain.
3. Right-click **Users**, and then click **New > User**.
4. Create a new user by using the **New Object - User** wizard. By default, a new user is a member of the **Domain Users** group.
5. Right-click the new user that is created in the *Domain Users* group, and click **Properties**. The **Username Properties** window opens, where *username* is the name of the new user. Complete the following steps in the **Username Properties** window:
   a) Click **Member of** tab. In the **Member of** area, add the **Performance Monitor Users** group.
   b) Click **Apply**, and then click **OK**.
6. Go to the `Candle_Home` directory. The default path is `C:\IBM\APM`.
7. Right-click the APM folder and click **Properties**. The **APM Properties** window opens. Complete the following steps in the **APM Properties** window:
   a) On the **Security** tab, click **Edit**.
   b) Click **Add** to add the new user and grant full access to this user.
   c) Click **Apply**, and then click **OK**.
8. Click **Start > Run**, and then type `services.msc`. The **Services** window opens. Complete the following steps in the **Services** window:
   a) Right-click the **Monitoring Agent** for Active Directory service, and click **Properties**.
   b) In the **Active Directory Properties** window, on the **Log On** tab, click **This Account**. Enter the user credentials.
   c) Click **Apply**, and then click **OK**.
9. Restart the agent service.

Configuring domain services for attribute group AD_Services_Status

You can configure the MS Active Directory Domain Services in `Services.properties` to be used or to be excluded
when determining `Server Status`. The attribute group `AD_Services_Status` and its situation are applicable for
Windows Server 2012 and later.

About this task

The `Services.properties` file contains the following default MS Active Directory Domain Services and its configuration.
True indicates that the service will be considered when determining Server Status value.
False indicates that the service will not be considered when determining Server Status value.

Table 175. MS Active Directory Domain Services and the configuration setting.

MS Active Directory Domain Services	Default Setting
DFS Replication	true
Remote Procedure Call (RPC)	false
DNS Client	true
DNS Server	true
Group Policy Client	false
Intersite Messaging	true
Kerberos Key Distribution Center	true
NetLogon	true
Windows Time	true
DHCP Client	false
Active Directory Web Services	false
Active Directory Federation Services	false

Note: Agent restart is required to enable data collection for the attribute group `AD_Services_Status` on Windows Server 2012 and later.

Procedure

1. Stop the agent.
2. Locate the `Services.properties` file for modification, if any.
   For 32-bit agent, the `Services.properties` file is located at *CANDLE_HOME*`\TMAITM6\`.
   For 64-bit agent, the `Services.properties` file is located at *CANDLE_HOME*`\TMAITM6_x64\`.

   The *CANDLE_HOME* is the agent installation directory.
3. If you want the Domain Services to be considered when determining the Server Status, set its value to true.
   If you want the Domain Services to be excluded when determining the Server Status, set its value to false.
   Save and close the file.
4. Start the agent.

Upgrading Microsoft Active Directory agent

You can upgrade the MS Active Directory agent to the latest version.

Before you begin

Ensure the `installAPMAgents.bat` file provided in the installer of the latest release is available in the machine that the agent is installed.

About this task

To upgrade the agent to the latest version, complete the following procedure.

Procedure

1. Logon to the machine that the agent is installed.
2. Launch a command prompt, run the `installAPMAgents.bat` file that is sourced from the installer of the latest release.

3. Enter the installation directory that the existing agent resides, and press enter.
4. The command prompt shows the base agent version and the target agent version to be upgraded. Press enter to proceed.
5. When agent upgrade is successful, the upgraded agent version is showed on the **IBM Performance Management** window.
6. On the **IBM Performance Management** window, right-click the agent and select **Reconfigure** from the drop-down menu.
7. To reflect the upgraded agent version on the **Application Performance Dashboard**, logon to the APM server and restart the APM server components by using the following commands.
   a. **`apm stop_all`**
   b. **`apm start_all`**
8. On the **IBM Performance Management** window, right-click the agent and select **Recycle** from the drop-down menu.

Results

The upgraded agent is reflected on the **Application Performance Dashboard**.

Note: It could take up to 30 minutes or more to show the upgraded agent on the **Application Performance Dashboard**.

Configuring Microsoft Cluster Server monitoring

You must configure the Monitoring Agent for Microsoft Cluster Server so that the agent can collect the cluster server data. Use the silent response file to configure the agent.

Before you begin

Ensure that you complete the following tasks:

- Create an empty resource group for the agent.
- Create a generic service cluster resource in the resource group of the agent on Windows Server 2008, 2012, 2016, and 2019 systems.
- Ensure that the user, who connects to the Microsoft Cluster Server environment or application, has administrator privileges. Use an existing user with administrator privileges, or create a new user. Assign administrator privileges to the new user by adding the new user to the Administrators group.

  Remember: To configure the Microsoft Cluster Server agent, you can use a local or a domain user provided that the user has administrator privileges.

Review hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Microsoft Cluster Server agent.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the “Change history” on page 54.

About this task

The Microsoft Cluster Server agent is a single instance agent. You must install and configure the agent manually in the same way on each node in the cluster. To configure the agent, see “Configuring the agent by using the silent response file” on page 495.

Creating a generic service cluster resource on Windows Server 2008, 2012, 2016, and 2019 systems

You must add the cluster agent service as a resource so that the agent can monitor the cluster server.

Before you begin

Ensure that the agent is stopped on each node in the cluster.

Procedure

To create a generic service cluster resource, complete the following steps:

1. Open the **Failover Cluster Manager** on any one of the cluster nodes.
2. Complete one of the following steps:
   - For Windows Server 2008:

     In the navigation pane, right-click **Services And Applications**, and then click **More Actions** > **Create Empty Service or Application**. The new service displays in the services and applications list. Rename the newly created service.
   - For Windows Server 2012:

     In the navigation pane, right-click **Roles**, and then click **More Actions** > **Create Roles**. The new service displays in the roles list.
   - For Windows Server 2016 and 2019:

     In the navigation pane, right-click **Roles**, and then click **Configure Roles**. The new service displays.
3. Right-click the new service and click **Add resource** > **Generic Service**.
4. In the **New Resource Wizard** window, select **Monitoring Agent for Microsoft Cluster Server** and click **Next**.
5. Click **Next** in the subsequent windows until you see the **Finish** button.
6. Click **Finish**.

   The agent service is added as a resource.
7. Right-click **Monitoring Agent for Microsoft Cluster Server** resource and click **Bring Resource Online**.

Results

The agent is started on the preferred node.

Configuring the agent by using the silent response file

The silent response file contains the Microsoft Cluster Server agent configuration parameters with default values defined for some parameters. You can edit the silent response file to configure the agent with different values for the configuration parameters.

Before you begin

Create a response file that contains the configuration parameters that you want to modify. If you want to modify the default configuration parameters, edit the response file.

About this task

You can configure the agent using the silent response file.

Procedure

1. Open the silent response file that is available at this path: `*install_dir*\samples \microsoft_cluster_server_silent_config.txt`
2. For the **`CTIRA_HOSTNAME`** environment variable, specify the cluster name as a value.

3. On each cluster node, run the following command: **`install_dir\BIN\microsoft_cluster_server-agent.bat config install_dir\samples\microsoft_cluster_server_silent_config.txt`**

What to do next

Change the user account from the local user to the domain user.

Changing the user account

After you configure the Microsoft Cluster Server agent, you can change the user account from the local user to the domain user.

About this task

By default, the agent runs under the local user account. The agent must be run under the domain user so that the agent can monitor all nodes in the cluster from a single node.

Procedure

To change the user account, complete the following steps:

1. Open the **IBM Performance Management** window.
2. Right-click the agent and click **Change Startup**.
3. Enter the domain login credentials.
4. Open the **Failover Cluster Manager** on one of the nodes, and start the cluster service.

Results

The agent is started on the node.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Performance Management console, see "Starting the Cloud APM console" on page 957.

Configuring Microsoft Exchange monitoring

You must configure the Monitoring Agent for Microsoft Exchange Server to monitor the availability and performance of Exchange Servers.

Before you begin

Before you configure the agent, ensure that you complete the following tasks:

- "Creating users" on page 497
- "Assigning administrator rights to the Exchange Server user" on page 499
- "Making the Exchange Server user a local administrator" on page 501
- "Configuring the Exchange Server for reachability" on page 502
- "Configuring the agent to run under the domain user" on page 503
- Review the hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Microsoft Exchange Server agent.

About this task

The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see "Change history" on page 54.

You can start the Microsoft Exchange Server agent after the agent is installed. However, manual configuration is required to view data for all the agent attributes.

- To configure the agent locally, see “Configuring the agent locally” on page 504.
- To configure the agent by using the silent response file, see “Configuring the agent by using the silent response file” on page 508.

Creating users

You can create a user for the agent on the Exchange Server manually or by running the *New User* utility. You must create the user on each Exchange Server that you want to monitor.

Before you begin

Install the Microsoft Exchange Server agent. To create a user, you must be a domain administrator with full administrator rights on the Microsoft Exchange Server.

About this task

Use one of the following procedures to create users:

- “Creating users on Exchange Server 2007 and 2010” on page 497
- “Creating users on Exchange Server 2013” on page 498
- “Creating users by running the New User utility” on page 498

Creating users on Exchange Server 2007 and 2010

You must create a user for the agent on Exchange Server 2007 and 2010 so that the agent can communicate and authenticate with the Exchange Server that you want to monitor.

Procedure

To create a user, complete the following steps:

1. Click **Start > Programs > Microsoft Exchange Server 2007 > Exchange Management Console**. The **Exchange Management Console** window opens.
2. In the Console tree, click **Mailbox in Recipient Configuration**.
3. In the Action pane, click **New Mailbox**. The New Mailbox wizard opens.
4. On the **Introduction** page, click **User Mailbox**.
5. On the **User Type** page, click **New User**.
6. On the **User Information** page, specify the following information:

   Organizational unit
   By default, the users container in the Active Directory is displayed. Click **Browse** to change the default organizational unit.

   First name
   Type the first name of the user.

   Initials
   Type the initials of the user.

   Last name
   Type the last name of the user.

   Name
   By default, the user's first name, initials, and last name are displayed in this field. You can modify the name.

   User log on name (User Principal Name)
   Type the name that the user must use to log on to the mailbox.

   User log on name (pre-Windows 2000, or earlier)
   Type the user name that is compatible with Microsoft Windows 2000 Server, or earlier.

   Password
   Type the password that the user must use to log on to the mailbox.

Confirm password
Retype the password that you entered in the **Password** field.

User must change password at next logon
Select this check box if you want the user to reset the password.

7. On the **Mailbox Settings** page, specify the following information:

   Alias
   By default, the value for this field is identical to the value that you specified in the **User logon name (User Principal Name)** field.

   Mailbox database
   Click **Browse** to open the **Select Mailbox Database** window. Select the mailbox database that you want to use and click **OK**.

   Managed folder mailbox policy
   Select this check box to specify a messaging records management (MRM) policy. Click **Browse** to select the MRM mailbox policy that you want to associate with this mailbox.

   Exchange ActiveSync mailbox policy
   Select this check box to specify an Exchange ActiveSync mailbox policy. Click **Browse** to select the Exchange ActiveSync mailbox policy that you want to associate with this mailbox.
8. On the **New Mailbox** page, review the configuration summary. Click **New** to create a mailbox. On the **Completion** page, the Summary section shows whether the mailbox was created.
9. Click **Finish**.

What to do next

Assign administrator rights to the Exchange user that you created.

Creating users on Exchange Server 2013

You must create a user for the agent on Exchange Server 2013 so that the agent can communicate and authenticate with the Exchange Server that you want to monitor.

Procedure

To create a user on Exchange Server 2013, complete the following steps:

1. Log in to the Exchange Admin Center with administrator credentials.
2. On the **Exchange admin center** page, click on the **recipients**, and then click **mailboxes**.
3. Click the down arrow next to the plus sign (+) that is located under the **mailboxes** option, and then click **User mailbox**.
4. On the "new user mailbox" page, click **New user**, and specify values for the other fields.
5. Click **Save**.

What to do next

Assign administrator rights to the Exchange user that you created.

Creating users by running the New User utility

You can run the New User utility to create users on Exchange Server 2007, or later. The user that is created by running this utility has all the required permissions to run the agent. This utility is installed when you install the agent.

Before you begin

Ensure that the agent is installed. To run the New User utility, you must be a domain administrator with full administrator rights on the Exchange Server.

About this task

When you run this utility, the user is created in the Users group of the Active Directory, and has the following permissions:

- On Exchange Server 2007:
  - Local administrator
  - Remote desktop user
  - Exchange recipient administrator
- On Exchange Server 2010, or later:
  - Local administrator
  - Remote desktop user
  - Exchange Servers or Public Folder Management.

Procedure

To run the New User utility, complete the following steps:

1. Double-click the `kexnewuser.exe` file that is available at the following location: *`install_dir`*`\TMAITM6_x64` Where *install_dir* is the path where the agent is installed.
2. In the **New User** window, complete the following steps:
   a) Enter the **first name** and the **last name** of the user.

      Restriction: The length of the first and the last name must not exceed 28 characters.

   b) In the **User Logon Name** field, enter the name that the user must type whenever the user logs in.

      Restriction: The length of the user logon name must not exceed 256 characters.

   c) In the **Password** field, enter your password.

   d) In the **Confirm Password** field, enter the password again.

   e) Select **User Must Change Password at Next Logon** if you want the specified password to be reset the next time when the user logs on.

   f) Click **Next**.

   The configuration values that you specify are validated, and error messages are displayed for incorrect values.
3. From the list of mailbox databases, select the required mailbox database, and click **Next**.

   A summary of configuration values is displayed.
4. Click **Finish**.

Results

The settings are saved, and the user is created.

Assigning administrator rights to the Exchange Server user

The user that you created for the Microsoft Exchange Server agent must be a domain administrator with full administrator rights on Microsoft Exchange Server. The administrator rights are necessary to access the Microsoft Exchange Server agent components.

Before you begin

Create an Exchange Server user who has the mailbox on the Exchange Server that is being monitored.

About this task

Use one of the following procedures to assign administrator rights to the user:

- "Assigning administrator rights on Exchange Server 2007" on page 500
- "Assigning administrator rights on Exchange Server 2010" on page 500

- "Assigning administrator rights on Exchange Server 2013" on page 500
- "Assigning administrator rights on Exchange Server 2016" on page 501

Assigning administrator rights on Exchange Server 2007

You must assign Exchange Recipient Administrator rights to the user on Exchange Server 2007.

Procedure

1. Click **Start > Programs > Microsoft Exchange Server 2007 > Exchange Management Console**. The **Exchange Management Console** window opens.
2. In the Console tree, click **Organization Configuration**.
3. In the Action pane, click **Add Exchange Administrator**.
4. On the **Add Exchange Administrator** page, click **Browse**. Select the new user that you created, and then select **Exchange Recipient Administrator** role.
5. Click **Add**.
6. On the **Completion** page, click **Finish**.

Assigning administrator rights on Exchange Server 2010

You must assign Exchange Servers or Public Folder Management rights to the user on Exchange Server 2010.

Procedure

1. Log on to Exchange server with Administrator privileges.
2. Click **Start > Administrative Tools > Server Manager.**
3. Expand the **Tools**.
4. Click **Active Directory Users and Computers**.
5. Expand **Domain**, click **Microsoft Exchange Security Groups**.
6. Right-click **Exchange Servers or Public Folder Management**, then click **Properties**.
7. In **Exchange Servers Properties or Public Folder Management Properties** window, go to **Members** and click **Add**.
8. From the list of users, select the user that you want to add to the group, and click **OK**.
9. Click **OK**.

Assigning administrator rights on Exchange Server 2013

You must assign Exchange Servers or Public Folder Management rights to the user on Exchange Server 2013.

Procedure

1. Log on to Exchange server with Administrator privileges.
2. Click **Start > Administrative Tools > Server Manager.**
3. Expand the **Tools**.
4. Click **Active Directory Users and Computers**.
5. Expand **Domain**, click **Microsoft Exchange Security Groups**.
6. Right-click **Exchange Servers or Public Folder Management**, then click **Properties**.
7. In **Exchange Servers Properties or Public Folder Management Properties** window, go to **Members** and click **Add**.
8. From the list of users, select the user that you want to add to the group, and click **OK.**
9. Click **OK**.

Assigning administrator rights on Exchange Server 2016

You must assign Exchange Servers or Public Folder Management rights to the user on Exchange Server 2016.

Procedure

1. Log on to Exchange server with Administrator privileges.
2. Click **Start > Administrative Tools > Server Manager.**
3. Expand the **Tools**.
4. Click **Active Directory Users and Computers**.
5. Expand **Domain**, click **Microsoft Exchange Security Groups**.
6. Right-click **Exchange Servers or Public Folder Management**, then click **Properties**.
7. In **Exchange Servers Properties or Public Folder Management Properties** window, go to **Members** and click **Add**.
8. From the list of users, select the user that you want to add to the group, and click **OK**.
9. Click **OK**.

What to do next

Make the user a local administrator of the computer where the Exchange Server is installed.

Making the Exchange Server user a local administrator

To access the Exchange Server data, the user that you created for the Microsoft Exchange Server agent must be a local administrator of the computer where the Exchange Server is installed.

Before you begin

Create an Exchange Server user.

About this task

Use one of the following procedures to make the user a local administrator:

- "Making the user a local administrator on Windows 2003 computer" on page 501
- "Making the user a local administrator on Windows 2008 computer" on page 502
- "Making the user a local administrator on Windows 2012 computer" on page 502
- "Making the user a local administrator on Windows 2016 computer" on page 502

Making the user a local administrator on Windows 2003 computer

You must make the user that you created for the Exchange Server a local administrator of the computer that runs on the Windows 2003 operating system, and where the Exchange Server is installed.

Procedure

1. Right-click **My Computer** on the computer desktop and click **Manage**.
2. Expand **Local Users and Groups**.
3. Click **Groups**.
4. Double-click **Administrators** to display the **Administrators Properties** window.
5. Click **Add**.
6. Select **Entire Directory** from the **Look in** list.
7. Select the name of the user that you created and click **Add**.
8. Click **OK**.
9. Click **OK**.

Making the user a local administrator on Windows 2008 computer

You must make the user that you created for the Exchange Server a local administrator of the computer that runs on the Windows Server 2008 operating system, and where the Exchange Server is installed.

Procedure

1. Click **Start > Administrative Tools > Server Manager**.
2. In the navigation pane, expand **Configuration**.
3. Double-click **Local Users and Groups**.
4. Click **Groups**.
5. Right-click the group to which you want to add the user account, and then click **Add to Group**.
6. Click **Add** and type the name of the user account.
7. Click **Check Names** and then click **OK**.

Making the user a local administrator on Windows 2012 computer

You must make the user that you created for the Exchange Server a local administrator of the computer that runs on the Windows Server 2012 operating system and where the Exchange Server is installed.

Procedure

1. Click **Start> Server Manager**.
2. On the **Server Manager dashboard** page, click **Tools > Computer Management**.
3. In the navigation pane of the **Computer Management** page, expand **Local Users and Groups**, and then click **Users**.
4. From the users list, right-click the user to which you want to assign administrator rights, and click **Properties**.
5. Click the **Member Of** tab, and click **Add**.
6. On the **Select Group** page, type `Administrators`, and then click **OK**.
7. Click **Apply** and **OK**.

Making the user a local administrator on Windows 2016 computer

You must make the user that you created for the Exchange Server a local administrator of the computer that runs on the Windows Server 2016 operating system and where the Exchange Server is installed.

Procedure

1. Click **Start> Server Manager** .
2. On the **Server Manager dashboard** page, click **Tools > Computer Management** .
3. In the navigation pane of the **Computer Management** page, expand **Local Users and Groups**, and then click **Users**.
4. From the users list, right-click the user to which you want to assign administrator rights, and click **Properties**.
5. Click the **Member Of** tab, and click **Add**.
6. On the **Select Group** page, type `Administrators`, and then click **OK**.
7. Click **Apply** and **OK**.

Configuring the Exchange Server for reachability

To verify reachability, the Microsoft Exchange Server agent sends an email message to the server, and measures the amount of time to receive an automated response. Before you start the agent, you must configure the Exchange Server to automatically respond to email messages.

Before you begin

Before you configure the Exchange Server, ensure that the following tasks are completed:

- A mailbox is created for the user on the Exchange Server that you want to monitor.
- The user that you created for the agent is a domain user.
- The servers in your Microsoft Exchange organization are configured for mail flow between servers.

Procedure

Complete the following steps for each Exchange Server for which you want to verify reachability:

1. Log in to Microsoft Outlook by specifying credentials of the user that you created.
2. Click **Next** on the **Startup** window.
3. Select **Yes** and click **Next**.
4. In the **Microsoft Exchange Server** field, type the name of the Exchange Server.
5. In the **Mailbox** field, type the name of the user that you created.
6. Click **Finish**.
7. Click **OK**.
8. Click **Tools > Rules and Alerts > New Rule**.
9. Select **Start from a blank rule**.
10. Select **Check messages when they arrive** and click **Next**.
11. Select the following options:
    - **Where my name is in the To: box**
    - **With specific words in the subject or body**
12. Under **Step 2** in the window, click **Specific words**.
13. In the **Specify words or phrases to search for in the subject or body** field, type `AVAILABILITY CHECK`.
14. Click **Add**.
15. Click **OK** and then click **Next**.
16. Select **Have the server reply using a specific message** and click **a specific message**.
17. In the email message editor, type the following text in the subject field of the message:
    `CHECK RECEIVED: MAILBOX AVAILABLE.`
18. Close the email message editor and click **Yes** to save these changes.
19. Click **Next**.
20. When you are asked about exceptions, do not specify any restrictions.
21. Click **Next**.
22. Click **Finish** and then click **OK**.

What to do next

Configure the Microsoft Exchange Server agent.

Configuring the agent to run under the domain user

By default, the Microsoft Exchange Server agent is configured to run under the local user. The agent must be run under the domain user that you created.

Before you begin

Ensure that:

- The user that you created is a domain user with local administrator rights.
- The user has administrator rights on the server where the agent is installed.

About this task

When the agent is run under the domain user, the agent can monitor all the components of the Exchange Server.

Procedure

To change the user under which the agent runs, complete the following steps:

1. Run the following command to verify which user ID is being used for starting the agent.

   ```
   install_dir\InstallITM\KinCinfo.exe -r
   ```
2. If the monitoring agent was started with a user ID that does not belong to the Administrator group, stop the agent.
3. Open the **Manage Monitoring Services** window.
4. Right-click the agent instance, and click **Change Startup**.
5. Specify the fully qualified user ID as `<Domain\Userid>`, and then specify the password.
6. Start the monitoring agent.

Configuring the agent locally

You can configure the agent locally by using the IBM Cloud Application Performance Management window.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Monitoring Agent for Microsoft Exchange Server**, and then click **Configure agent**.

   ⚠ **Attention:** Click **Reconfigure** if **Configure agent** is disabled.
3. In the **Monitoring Agent for Microsoft Exchange Server: Agent Advanced Configuration** window, click **OK**.
4. In the **Agent Configuration** window, complete the following steps:

   a) Click the **Exchange Server Properties** tab, and specify values for the configuration parameters. When you click **OK**, the specified values are validated.

   b) Click the **Exchange Services Monitoring** tab, and specify values for the configuration parameters. When you click **OK**, the specified values are validated.

   c) Click the **Advanced Configuration Properties** tab, and specify values for the configuration parameters. When you click **OK**, the specified values are validated.

   For information about the configuration parameters in each tab of the **Agent Configuration** window, see the following topics:

   - "Configuration parameters for the Exchange Server properties" on page 505
   - "Configuration parameters for Exchange services" on page 506
   - "Configuration parameters for reachability" on page 506

   For information about the validation of configuration values, see "Validation of configuration values" on page 507.
5. Recycle the agent.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Restriction: On the Cloud APM dashboard, instances of only one of the Exchange component types (Microsoft Exchange Server or Microsoft Exchange Server 2013) are displayed under My Components.

Configuration parameters for the Exchange Server properties
In the **Exchange Server Properties** tab of the **Agent Configuration** window, you can configure the Exchange Server properties, such as server name, domain name, and user name.

The following table contains detailed descriptions of the configuration settings in the **Exchange Server Properties** tab.

Table 176. Names and descriptions of configuration settings in the Exchange Server Properties tab

Parameter name	Description	Mandatory field	Examples
Exchange Server Name	The name of the Exchange Server. During installation of the Exchange Server, the default Exchange Server name is the Windows Server host name. If you change the default Exchange Server name, you must use the changed name when you configure the Exchange Server agent. **Remember:** In clustered and distributed environments, specify the Mailbox Server name for Exchange Server 2007.	Yes **Important:** Do not specify a value if the agent is installed on a server that has a single copy cluster with more than two nodes.	If the Exchange Server name is `popcorn`, enter `popcorn` in the **Exchange Server Name** field.
Exchange Domain Name	The name of the domain where the Exchange Server is installed.	Yes	If the Exchange Server is in the `LAB.XYZ.com` domain, enter the name that precedes the first dot, for example, LAB.
Exchange User Name	The name of the user who is configured to access the Exchange Server. **Remember:** The user must have a mailbox on the same Exchange Server.	Yes	
Exchange User Password	The password of the user who is configured to access the Exchange Server.	Yes	
Confirm Password	The same password that you specified for the Exchange Server user.	Yes	
Exchange MAPI Profile Name	MAPI profiles are the primary configuration settings that are required for accessing the Exchange Server. This field is disabled if you are using a 64-bit Microsoft Exchange Server agent to monitor Exchange Server 2007, or later.	No	
Configuration in cluster	Select this check box if you want to configure the Microsoft Exchange Server agent in a cluster environment.	Not applicable	
Cluster Server Name	The name of the Cluster Server. This field is enabled when you select the **Configuration in cluster** check box.	Yes, if the field is enabled.	`SCCCluster`

Table 176. Names and descriptions of configuration settings in the Exchange Server Properties tab (continued)

Parameter name	Description	Mandatory field	Examples
Exchange Subsystem ID	The name of the Cluster Server node. This field is enabled when you select the **Configuration in cluster** check box.	Yes, if the field is enabled.	`node1`
Exchange Agent Historical Data Directory	The location on the disk where the historical data is stored. This field is enabled when you select the **Configuration in cluster** check box.	Yes, if the field is enabled.	`c:\history`

Configuration parameters for Exchange services

In the **Exchange Services Monitoring** tab of the **Agent Configuration** window, you can select the Exchange services to know the Exchange Server status.

The following table contains detailed descriptions of the configuration settings in the **Exchange Services Monitoring** tab.

Table 177. Names and descriptions of configuration settings in the Exchange Services Monitoring tab

Parameter name	Description	Mandatory field
Exchange Services	Select the Exchange services from the available list of services, and click the arrow to move the selected services to the **Services Configured for Server Status** list so that the Microsoft Exchange Server agent can monitor them. **Remember:** The list of available services changes according to the Exchange Server version and the roles that are installed.	Not applicable
Services Configured for Server Status	The services that are already available in this list determine the status of the Exchange Server. These services are mandatory and cannot be moved from the **Services Configured for Server Status** list to the **Exchange Services** list. You can add more services to the **Services Configured for Server Status** list by moving the services from the **Exchange Services** list. You can move these additional services back to the **Exchange Services** list.	Not applicable

Configuration parameters for reachability

In the **Advanced Configuration Properties** tab of the **Agent Configuration** window, you can configure the parameters that are related to reachability, such as target email address and reachability interval.

The following table contains detailed descriptions of the configuration settings in the **Advanced Configuration Properties** tab.

Table 178. Names and descriptions of configuration settings in the Advanced Configuration Properties tab

Parameter name	Description	Mandatory field
Enable Mailbox Reachability Monitoring	Select this check box if you want the agent to capture the reachability metrics data.	Not applicable

Table 178. Names and descriptions of configuration settings in the Advanced Configuration Properties tab (continued)

Parameter name	Description	Mandatory field
Target Email Address	An email address to verify reachability. Separate multiple email addresses with a semicolon (;). **Restriction:** The total number of characters in this field must not exceed 1023.	Yes, if this field is enabled.
Email Transmission Interval (seconds)	The waiting time (in seconds) of the Exchange Server agent between sending emails.	Yes, if this field is enabled.
Email Transmission Timeout (seconds)	The interval (in seconds) for which the agent waits for a response to the email that was sent to test whether the Mailbox Server is reachable.	No
Enable Mailbox Detail Monitoring	Select this check box to collect data for the mailbox detail metrics.	Not applicable
Mailbox Detail Collection Start time	The time (in hh:mm:ss format) when mailbox detail metrics are collected.	No
Mailbox Detail Collection Interval (seconds)	The interval (in seconds) between collections of mailbox detail metrics.	No
Event Logs Collection Time (minutes)	The duration (in minutes) for which the agent collects event records.	No
Maximum Number of Events	The maximum count up to which event records are collected. The collection of event records stops when the number of collected event records exceeds the maximum count.	No
Collection Interval (seconds)	The interval (in seconds) between the agent cycles.	No
Exchange Topology Interval (seconds)	The interval (in seconds) between collections of topology detail information.	No
Message Tracking Collection Interval (hours)	The interval (in hours) for which the message tracking logs are collected. **Restriction:** The interval value must be in the range 1 - 12. If you specify the interval value that is greater than 12, the value is saved as 12. If you enter an invalid value that contains alphabets or special characters, the value is saved as 0, which indicates that the message tracking collection is disabled. This field is disabled if any of the following conditions is true: • The Mailbox Server role or the Hub Transport role is not installed on the Exchange Server. • The message tracking feature is disabled on the Exchange Server.	No

Validation of configuration values
The values that you specify while configuring the agent are validated. The validation ensures that the values are specified for all mandatory parameters and certain conditions are met, such as local administrator rights for the user.

The following table shows the validation tests that are performed on the specified configuration values.

Table 179. Validation tests

Validation test	Verifies whether
Exchange Server Name	The Mailbox Server name of the user matches the specified Exchange Server name.
Exchange Server Rights	The user has the required Exchange Server rights. On Exchange Server 2007, the user must have recipient administrator rights, and on Exchange Server 2010, or later, the user must have recipient management rights.
Local Admin	The user has local administrator rights.
Agent Service Logon	The agent service is configured to run with the specified user account.

If one or more validation tests fail, an error message is generated. You must specify values for all mandatory parameters. Otherwise, you cannot save the configured values.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters with default values defined for some parameters. You can edit the silent response file to configure the agent with different values for the configuration parameters.

About this task

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

1. Open the `msex_silent_config.txt` file that is located at *`install_dir`*`\samples`, and specify values for all mandatory parameters.

   You can also modify the default values of other parameters.
2. Run the following command:

   `install_dir\BIN\msexch-agent.bat config install_dir\samples \msex_silent_config.txt`

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Restriction: On the Cloud APM dashboard, instances of only one of the Exchange component types (Microsoft Exchange Server or Microsoft Exchange Server 2013) are displayed under My Components.

Configuring local environment variables for the agent

You can configure the local environment variables for the Microsoft Exchange Server agent to enable or disable event throttling for duplicate events.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Cloud Application Performance Management**.
2. In the **IBM Performance Management** window, from the **Actions** menu, click **Advanced** > **Edit ENV File**.
3. In the KEXENV file, change the values of the following environment variables:

EX_EVENT_THROTTLE_ENABLE
This variable enables you to throttle duplicate events. The default value is `False`. To enable event throttling to prevent triggering of situations for duplicate events, set the value of this variable to True.

EX_EVENT_THROTTLE_DURATION
This variable provides the duration (in minutes) for throttling of events. The default value is `0` minutes.

Configuring Microsoft Hyper-V monitoring

When you install the Monitoring Agent for Microsoft Hyper-V Server, the agent is automatically configured and started with the default configuration settings. Use the silent response file to modify the default configuration settings.

Before you begin

- Review hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Microsoft Hyper-V Server agent.
- Create a response file that contains the configuration parameters that you want to modify.
- To view the virtual machine data in the Virtual Machine page, ensure that you install the integration component and the OS agent on each virtual machine. For virtual machines that run on the Linux system, ensure that you complete the following tasks:
  - Upgrade the Linux system.
  - Install the updated `hypervkvpd` or `hyperv-daemons rpm` package on the virtual machine.

About this task

You can configure the agent when the agent is running or stopped. The agent remains in the same state after configuration. For example, if the agent is running, it remains in the running state after configuration.

Important: For the 8.1.3 release of Performance Management, the agent configuration window is removed because it is not required. The agent configuration window is available for 8.1.2, or earlier versions of Performance Management.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the "Change history" on page 54.

Procedure

To configure the agent, complete the following steps:

1. Open the `microsoft_hyper-v_server_silent_config.txt` file that is at *`install_dir`*`\samples`, and specify values for all mandatory parameters.

   You can also modify the default values of other parameters.

2. Open the command prompt, and enter the following command:

   `install_dir\BIN\microsoft_hyper-v_server-agent.bat config install_dir \samples\microsoft_hyper-v_server_silent_config.txt`

   The response file contains the following parameters:

   - KHV_DIRECTOR_PORT
   - KHV_DIRECTOR_SERVER

   Remember: The agent configuration is organized into the following groups:

   IBM Systems Director configuration (IBM_DIRECTOR_CONFIGURATION)
   The configuration elements that are defined in this group are always present in the agent's configuration. This group defines information that applies to the entire agent.

IBM Systems Director Server Port Number (KHV_DIRECTOR_PORT)
The port number for the IBM Systems Director Server. The default value is none.

IBM Systems Director Server Host Name (KHV_DIRECTOR_SERVER)
The host name or IP address of the IBM Systems Director Server that is managing the environment. The default value is none.

3. Start the agent if it is in the stopped state.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Providing Local Security Policy for running Monitoring Agent for Microsoft Hyper-V Server on Windows by a Non-Administrator user

Local security policies are available to run the Monitoring Agent for Microsoft Hyper-V Server on Windows by a non-administrator user.

About this task

A combination of following two local security policies works to run the Microsoft Hyper-V Server agent on Windows by a non-administrator user. For the Microsoft Hyper-V Server agent to start or stop, configure, and verify data, use these two policies.

- Debug Programs
- Log on as Service

Also, following attribute groups need administrator rights to get data on the APM portal:

- Availability
- Migration
- VM Mig WO Cluster
- VM Storage Migration

Follow the procedure that is given to avail the Local Security permissions for a non-administrator user.

Procedure

1. Install the Microsoft Hyper-V Server agent as a local administrator.
2. Add the non-administrator user under the `install_dir` directory and provide the following permissions to it:
   a) Provide full access to the `HKEY_LOCAL_MACHINE\SOFTWARE\IBMMonitoring` registry.
   b) Provide read access to the non-administrator user in the `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib` registry.
   c) Provide full access to the non-administrator user in the `install_dir` directory.
3. Go to the **Start** menu and run the **`secpol.msc`** command to open the Local Security policies.
4. To add a non-administrator user in the policies, refer "Granting Local Security Policy permissions" on page 511.
5. To add a non-administrator user in the Hyper-V Administrator Users group, refer "Adding a non-administrator user in the Hyper-V administrator users group" on page 512.
6. To add a non-administrator user in the Performance Monitor Users group, refer "Adding a non-administrator user in the Performance Business Monitor users group" on page 512.
7. To modify the DCOM security permission for a non-administrator user, refer "Modifying DCOM permissions" on page 511.
8. Restart the Microsoft Hyper-V Server agent and verify data on the APM portal.

Granting Local Security Policy permissions

To start or stop, configure, and verify data for the Microsoft Hyper-V Server agent, you need to grant permissions to these two local security policies: Debug Programs and Log on as Service.

Granting Debug Programs permission

About this task

To grant the Debug Programs permission, complete the following procedure.

Procedure

1. Click **Start > Control Panel > Administrative Tools > Local Security Policy**. The **Local Security Settings** window opens.
2. Expand **Local Policies** and click **User Rights Assignment**. The list of policies opens.
3. Double-click the **Debug Programs** policy. The **Debug Programs Properties** window opens.
4. Click **Add User or Group**. The **Select Users or Groups** window opens.
5. In the **Enter the object names to select** field, enter the user account name to whom you want to assign permissions, and then click **OK**.
6. Click **Apply**, and then click **OK**.

Granting Log on as Service permission

About this task

To grant the Log-on as Service permission, complete the following procedure.

Procedure

1. Click **Start > Administrative Tools > Local Security Policy**. The **Local Security Settings** window opens.
2. Expand **Local Policies** and click **User Rights Assignment**. The list of policies opens.
3. Double-click the **Log-on as service** policy. The **Log-on as service Properties** window opens.
4. Click **Add User or Group**. The **Select Users or Groups** window opens.
5. In the **Enter the object names to select** field, enter the user account name to whom you want to assign permissions, and then click **OK**.
6. Click **Apply**, and then click **OK**.

Modifying DCOM permissions

You need to modify DCOM permissions to run the Microsoft Hyper-V Server agent with the non-administrator user access.

About this task

To modify DCOM permissions, verify that the user has appropriate permissions to start the DCOM server. To modify permissions, complete the following procedure.

Procedure

1. Using the **Regedit** command, go to the `HKCR\Clsid\*clsid value` registry value.

   Note: When you configure the agent with a non-administrator user, the CLSID value is displayed in the event viewer with the event ID 10016.
2. In the Registry Editor pane, double-click **Default**.
3. In the **Edit string** dialog box, copy the value data string.
4. Click **Start > Control Panel > Administration Tools > Component Services**.

5. In the **Component Services** window, expand **Component Services > Computers > My Computer**, and double-click **DCOM**.
6. In the DCOM config pane, locate the copied string (program name), right-click the program name, and then click **Properties**.
7. In the **Properties** window, select the **Security** tab.
8. Under the **Launch and Activation Permissions** group box, select **Customize**, and then click **Edit**. The **Launch and Activation Permissions** window opens.
9. Click **Add**, enter a non-administrator user to the permission list, and click **OK**.
10. Select the **Allow** check box for Local Launch and Local Activation, and then click **OK**.

Adding a non-administrator user in the Hyper-V administrator users group

You need to add a non-administrator user in the Hyper-V administrator users group to get data on the APM portal.

About this task

To add a non-administrator user in the Hyper-V administrator users group, complete the following procedure.

Procedure

1. Click **Start > Control Panel > Administration Tools > Computer Management**. The **Computer Management** window opens.
2. Click **System Tools > Local Users and Groups > Groups**. The list of groups opens.
3. Double-click the **Hyper-V Administrators** group. The **Hyper-V Administrators Properties** window opens.
4. Click **Add**. The **Select Users or Groups** window opens.
5. In the **Enter the object names to select** field, enter the user account name to whom you want to assign permissions, and then click **OK**.
6. Click **Apply**, and then click **OK**.

Adding a non-administrator user in the Performance Business Monitor users group

You need to add a non-administrator user in the Performance Monitor users group to get data on the APM portal.

About this task

To add a non-administrator user in the Performance Business Monitor users group, complete the following procedure.

Procedure

1. Click **Start > Control Panel > Administration Tools > Computer Management**. The **Computer Management** window opens.
2. Click **System Tools > Local Users and Groups > Groups**. The list of groups opens.
3. Double-click the **Performance Monitor Users** group. The **Performance Business Monitor Users Properties** window opens.
4. Click **Add**. The **Select Users or Groups** window opens.
5. In the **Enter the object names to select** field, enter the user account name to whom you want to assign permissions, and then click **OK**.
6. Click **Apply**, and then click **OK**.

Configuring Microsoft IIS monitoring

When you install the Monitoring Agent for Microsoft Internet Information Services, the agent is automatically configured and starts with the default configuration settings.

Before you begin

- Review hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Microsoft IIS agent.
- Ensure that the user, who connects to the Microsoft Internet Information Server environment or application, has administrator privileges. Use an existing user with administrator privileges, or create a new user. Assign administrator privileges to the new user by adding the new user to the Administrators group.

  Remember: To configure the Microsoft IIS agent, you can use a local or a domain user provided that the user has administrator privileges.

About this task

You can configure the agent when the agent is running or stopped. The agent remains in the same state after configuration. For example, if the agent is running, it remains in the running state after configuration.

To configure the agent, you can either use the **IBM Performance Management** window or the silent response file.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the “Change history” on page 54.

What to do next

After you configure the agent, you can change the user account from the local user to the domain user. For steps to change the user account, see “Changing the user account” on page 515.

Configuring the agent on Windows systems

You can configure the Microsoft IIS agent on Windows operating systems by using the **IBM Performance Management** window. After you update the configuration values, you must start the agent to save the updated values.

About this task

You can configure the agent when the agent is running or stopped. The agent remains in the same state after configuration. For example, if the agent is running, it remains in the running state after configuration.

The Microsoft IIS agent provides default values for some parameters. You can specify different values for these parameters.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Monitoring Agent for Microsoft Internet Information Services**, and click **Reconfigure**.
3. In the Monitoring Agent for Microsoft Internet Information Services window, complete the following steps:
   a) On the **HTTP Error Log Configuration** tab, specify a location to save the log file, and click **Next**.

      Note: By default, this log file is saved at the following location: `C:\WINDOWS\system32\LogFiles\HTTPERR`. The administrator can change the location of the log file.

   b) On the **Site Log Configuration** tab, specify a location to save the log file, and click **OK**.

Note: By default, this log file is saved at the following location: `C:\inetpub\logs\LogFiles`. The administrator can change the location of the log file.

4. In the **Restart of Monitoring Agent for Microsoft IIS** window, click **Yes**.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you need help with troubleshooting, see the Troubleshooting section.

Configuring the agent by using the silent response file

When you install the Microsoft IIS agent, the agent is automatically configured and starts with the default configuration settings. Use the silent response file to modify the default configuration settings.

Before you begin

Create a response file that contains the configuration parameters that you want to modify. If you want to modify the default configuration parameters, edit the response file.

About this task

You can configure the agent when the agent is running or stopped. The agent remains in the same state after configuration. For example, if the agent is running, it remains in the running state after configuration.

Procedure

To configure the Microsoft IIS agent, complete the following steps:

1. On the command line, change the path to the directory that contains the `msiis-agent.bat` file.
2. Enter the following command: **`msiis-agent.bat config`** *absolute path to the response file*.

   The response file contains the following parameters:

   KQ7_SITE_LOG_FILE
   C:\inetpub\logs\LogFiles

   KQ7_HTTP_ERROR_LOG_FILE
   C:\WINDOWS\system32\LogFiles\HTTPERR

   Remember: The agent configuration is organized into the following groups:

   Site Log Configuration (SITE_LOG)
   This group contains the configuration parameters that are related to the site log file (KQ7_SITE_LOG_FILE). An administrator can specify a location to save the log file. By default, this log file is saved at the following location: `C:\inetpub\logs\LogFiles`

   HTTP Error Log Configuration (HTTP_ERROR_LOG)
   This group contains the configuration parameters that are related to the HTTP error log file (KQ7_HTTP_ERROR_LOG_FILE). An administrator can specify a location to save the log file. By default, this log file is saved at the following location: `C:\WINDOWS\system32\LogFiles\HTTPERR`
3. If the agent is in the stopped state, start the agent.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you need help with troubleshooting, see the Troubleshooting section.

Changing the user account

After you configure the Microsoft IIS agent, you can change the user account from the local user to the domain user.

About this task

By default, the Microsoft IIS agent runs under the local user account.

Procedure

1. Run the following command to verify which user ID is being used for starting the agent:

   ```
   install_dir\InstallITM\KinCinfo.exe -r
   ```
2. If the monitoring agent was started with a user ID that does not belong to the Administrator group, stop the agent.
3. Open the **Manage Monitoring Services** window.
4. Right-click the agent instance, and click **Change Startup**.
5. Specify the fully qualified user ID as <Domain\User ID>, and then specify the password.
6. Start the Microsoft IIS agent.

Prerequisite to install Web Application Attribute Group

You need to check the following set of prerequisites before you install the Web Application attribute group:

- Run the `host` command in Windows Power Shell to check the version of PowerShell. If the version is not 5.1, then download the required version from https://www.microsoft.com/en-us/download/details.aspx?id=54616.
- Check the version of the .NET framework. To check the version, in the registry, go to location `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NET Framework Setup\NDP\v4\Full`. If the .NET framework is 4.6 or less, get the required version of installation for the .NET framework from https://www.microsoft.com/en-in/download/details.aspx?id=49982.
- Then install the PowerShell module.
  - If the operating system is Windows Server 2008 R2, install the Carbon module using the command, `Install-Module -Name Carbon`.
  - If the operating system is Windows Server 2012 or Windows Server 2012 R2, install the IISAdministration module using the command `Install-Module -Name IISAdministration`.
  - The Windows Server 2016 or Windows Server 2019 has all the required features. Installing IIS server from the server manager installs the IISAdministration module.

Configuring Skype for Business Server monitoring

When you install the Monitoring Agent for Skype for Business Server (formerly known as MS Lync Server), the agent will be in the unconfigured state. To start the agent, you need to configure it.

Before you begin

- Review the hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Skype for Business Server agent.
- Ensure that the user that you use to run the Skype for Business Server agent, is a domain user with administrator privileges and has access to all the remote servers that are listed in the Lync or Skype for Business Server topology. Use an existing domain user with administrator privileges, or create a new domain user and assign administrator privileges to the new domain user.

About this task

You can configure the agent when the agent is running or stopped. The agent remains in the same state after configuration. For example, if the agent is running, it remains in the running state after configuration.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the "Change history" on page 54.

To configure the agent, you can either use the **IBM Performance Management** window or the silent response file.

What to do next

After you configure the agent, you can change the user account from the local user to the domain user. For steps to change the user account, see "Changing the user account" on page 518.

Permissions and access rights for a non-administrator user

You can run the monitoring agent for Skype for Business Server agent as a non-administrator user; however, some functions are inaccessible.

Registry Permissions

To create a non-administrator user, create a new user (non-administrator) and set up registry permissions for the new user as follows.

- Full access to the KEY_LOCAL_MACHINE\SOFTWARE\IBMMonitoring
- Full access to the CANDLE_HOME directory

The non-administrator user must be a member of the Performance Monitor Users and Performance Log Users. If you define these permissions for a non-administrator user, data is displayed for all the Perfmon-based attribute groups.

To view attribute groups' data collected from Database

If you want to view data for attribute groups that is collected from database, you must set up the following permissions for the non-administrator user.

- The non-administrator user account that you use to run the Skype for Business Server agent must have the Debug Program permission to add a debugger to any process.

  By default, the Debug Program permission is assigned only to the administrator and Local System accounts. To grant the Debug Program permission, you must complete the following steps on the Lync or Skype for Business Server:

  1. Click **Start > Administrative Tools > Local Security Policy**. The **Local Security Settings** window opens.
  2. Expand **Local Policies** and then click **User Rights Assignment**. The list of user rights opens.
  3. Double-click the **Debug Programs policy**. The **Debug programs Properties** window opens.
  4. Click **Add User or Group**. The **Select Users or Groups** window opens.
  5. In the Enter the object names to select field, enter the user account name to whom you want to assign permissions, and then click **OK**.
  6. Click **OK**.
- Grant Log on as Service permission

  To grant the Log-on as service permission, you must complete the following steps on the Lync or Skype for Business Server:

  1. Click **Start > Administrative Tools > Local Security Policy**. The **Local Security Settings** window opens.

2. Expand **Local Policies** and then click **User Rights Assignment**. The list of user rights opens.
3. Double-click the **Log-on** as service policy. The **Log-on as service Properties** window opens.
4. Click **Add User or Group**. The **Select Users or Groups** window opens.
5. In the Enter the object names to select field, enter the user account name to whom you want to assign permissions, and then click **OK**.
6. Click **OK**.

The Availability attribute group show data for users who are members of the Administrators group.

Configuring the agent on Windows systems

You can configure the Skype for Business Server agent (formerly known as MS Lync Server agent) on Windows operating systems by using the **IBM Performance Management** window. After you update the configuration values, you must start the agent to save the updated values.

About this task

You can configure the agent when the agent is running or stopped. The agent remains in the same state after configuration. For example, if the agent is running, it remains in the running state after configuration.

The Skype for Business Server agent provides default values for some parameters. You can specify different values for these parameters.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Monitoring Agent for Skype for Business Server**, and then click **Configure agent**.
3. In the Monitoring Agent for Skype for Business Server window, complete the following steps:
   a) On the **SQL Configuration for Skype for Business Topology** tab, to connect to the Microsoft Lync Server or Skype for Business Server Central Management Store, specify values for the configuration parameters, and then click **Next**.

      Note: You can skip this tab, as SQL Configuration for Skype for Business Topology is not applicable for IBM Cloud Application Performance Management.

      Important: Synthetic transaction configuration is optional. If you require the synthetic transaction data, specify the configuration parameters on the **Setup Information** and **Scheduler Configuration** tabs.

   b) On the **Administrator Login Credentials** tab, specify the administrator credentials, and then click **Next**.

   c) On the **Setup Information** tab, to run commands for the synthetic transactions, specify values for the configuration parameters, and then click **Next**.

   d) On the **Scheduler Configuration** tab, to schedule the synthetic transactions, specify values for the configuration parameters, and then click **Next**.

   e) On the **SQL Server Configuration for Skype for Business Monitoring Role** tab, to connect to the Microsoft Lync Server or Skype for Business Server monitoring role, specify values for the configuration parameters, and then click **Next**.

   For information about the configuration parameters, see “Configuration parameters for the agent” on page 519.
4. In the **IBM Performance Management** window, right-click **Monitoring Agent for Skype for Business Server**, and then click **Start**.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see “Starting the Cloud APM console” on page 957.

If you need help with troubleshooting, see the Troubleshooting section.

Configuring the agent by using the silent response file

When you install the Skype for Business Server agent (formerly known as MS Lync Server agent), the agent will be in the unconfigured state. To start the agent, you need to configure it.
The silent response file contains the agent configuration parameters with default values defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

Before you begin

Create a response file that contains the configuration parameters that you want to modify. If you want to modify the default configuration parameters, edit the response file.

About this task

You can configure the agent when the agent is running or stopped. The agent remains in the same state after configuration. For example, if the agent is running, it remains in the running state after configuration.

Procedure

To configure the Skype for Business Server agent, complete the following steps:

1. Open the command prompt.
2. Change the path to the directory that contains the `skype_for_business_server-agent.bat` file.
3. Enter the **`skype_for_business_server-agent.bat config`** *absolute path to the response file* command.

   For information about the configuration parameters, see "Configuration parameters for the agent" on page 519.
4. Start the agent if it is in the stopped state.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you need help with troubleshooting, see the Troubleshooting section.

Changing the user account

After you configure the Skype for Business Server agent, you can change the user account from the local user to the domain user.

About this task

By default, the Skype for Business Server agent runs under the local user account. When the agent runs under the domain user, the agent can collect data from the remote servers.

Procedure

1. Run the following command to verify which user ID is being used for starting the agent:

   ```
   install_dir\InstallITM\KinCinfo.exe -r
   ```
2. If the monitoring agent was started with a user ID that does not belong to the Administrator group, stop the agent.
3. Open the **Manage Monitoring Services** window.
4. Right-click the agent instance, and click **Change Startup**.
5. Specify the fully qualified user ID as <Domain\User ID>, and then specify the password.
6. Start the Skype for Business Server agent.

Configuration parameters for the agent

When you configure the Skype for Business Server agent (formerly known as MS Lync Server agent), you can change the default values of the configuration parameters, such as the database server name, database instance name, database name, and other parameters.

The following table contains descriptions of the configuration parameters for the Skype for Business Server agent.

Note: Out of all the fields, the Pool FQDN field is mandatory in following table.

Table 180. Names and descriptions of the configuration parameters for the agent

Parameter name	Description
Database Server Name (for example, PS6877)	• **SQL Configuration for Skype for Business Topology** tab: The name of the database server where the Lync or Skype for Business Server Central Management Store is installed. • **SQL Server Configuration for Skype for Business Monitoring Role** tab: The name of the database server where the monitoring role is installed.
Database Instance Name	• **SQL Configuration for Skype for Business Topology** tab: The default instance. • **SQL Server Configuration for Skype for Business Monitoring Role** tab: The name of the database instance where the monitoring role is installed.
Database Name	The name of the database.
Database User ID	The user ID of the database. This user must have access to the required Microsoft SQL Server instance. This user need not be an Active Directory user.
Database Password	The password of the database where the monitoring role is installed.
Username (Example: skype\administrator)	The user ID of the administrator. This user must be a domain user with administrator privileges and access to all the remote servers that are listed in the Lync or Skype for Business Server topology. The credentials of this user are also used in Synthetic Transaction feature. So, this user should be authorized to create Windows Schedule in Task Scheduler and execute Synthetic Transaction Commands.
Password	The login password of administrator.
Confirm Domain Password	Enter the same password that you specified in the Domain Password field.
Pool FQDN	The fully qualified domain name (FQDN) of Skype Pool for which synthetic commands are executed.
Geographic Location	The geographic location of the production system.
Test Users1 (for example, user1@skype.com)	First Username which can be used while executing Synthetic Transaction cmdlets. Format for username is SAMAccountName@domain.com. Do not provide Sip Address.
Test User1 PWD	The password of Test User1.

Table 180. Names and descriptions of the configuration parameters for the agent (continued)

Parameter name	**Description**
Confirm Test User1 PWD	Enter the same password that you specified in the **`Test User1 PWD`** field.
Test User2 (for example, user2@skype.com)	Second Username which can be used while executing Synthetic Transaction cmdlets. Format for username is SAMAccountName@domain.com. Do not provide Sip Address.
Test User2 PWD	The password of Test User2.
Confirm Test User2 PWD	Enter the same password that you specified in the **`Test User2 PWD`** field.
Use Agent Configuration Values	Keep this field enabled if you want to run synthetic commands using all fields provided in configuration panel. Disable to use values set by New-CsHealthMonitoringConfiguration. If disabled, the value of **`Pool FQDN`** will be used as identity for Get-CsHealthMonitoringConfiguration. Make sure to provide valid test user credentials to execute **`Test-CsMcxP2PIM`** command.
Frequency	The frequency of the scheduled utility that fetches the data of synthetic transactions. The frequency can have the following values: • Daily (DAY_FREQUENCY) • Weekly (WEEK_FREQUENCY) • Monthly (MONTHLY_FREQUENCY)
Collection Hour	The hour part of the time-stamp, in the 24-hour clock format that you select to schedule the utility.
Collection Minute	The minutes part of the time-stamp that you select to schedule the utility.
Start Date (YYYY-MM-DD)	The time when the scheduler is activated.
End Date (YYYY-MM-DD)	The time when the scheduler is deactivated.

Configuring Microsoft .NET monitoring

The Monitoring Agent for Microsoft .NET monitors .NET applications. The agent starts automatically after installation to collect resource monitoring data. However, to collect transaction tracking and diagnostics data, you must complete some configuration tasks.

Before you begin

Review the hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Microsoft .NET agent.

About this task

After the agent is installed, complete the following configuration tasks so that the agent can collect the transaction tracking and diagnostics data:

1. Registering the data collector

The data collector is a component of the Microsoft .NET agent. It collects the transaction tracking and diagnostics data and passes the data to the Microsoft .NET agent. You must register the data collector for collecting this data. For details, see "Registering the data collector" on page 522.

2. Configuring the collection of transaction tracking and diagnostics data
   After you register the data collector, enable the collection of transaction tracking and diagnostics data on the Cloud APM console. You can also enable the diagnostics data collection by using the **`configdc`** command. For details, see "Enabling collection of transaction tracking and diagnostics data" on page 525 and "Enabling the collection of diagnostics data by using the configdc command" on page 525.
3. Activating the configuration updates
   If you enable the diagnostics data collection by using the **`configdc`** command, you must activate the configuration so that the updates are saved to the configuration file. For more information about activating the configuration changes, see "Activating the configuration updates" on page 527.
4. Tuning the performance of data collector
   You might need to complete some tasks to fine tune the performance of data collector. For details, see "Performance tuning of data collector" on page 527.

Agent coexistence
In an agent coexistence environment, you can view the transaction tracking data from either the Cloud APM console or Tivoli® Enterprise Portal. For information about enabling data collection for transaction tracking in the agent coexistence environment, see "Enabling transaction tracking in agent coexistence environment" on page 526.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version. To access the documentation for earlier agent releases, see the following table:

Table 181. Agent versions and documentation

Microsoft .NET agent version	Documentation
8.1.3.2	IBM Cloud Application Performance Management
8.1.3 and 8.1.2	IBM Performance Management 8.1.3

The link opens an on-premises Knowledge Center topic.

Permissions to run an agent by using a local or domain account

Only a local or domain user who is a member of Administrators group has permissions to run the Microsoft .NET agent. This topic provides conditions that must be met if the local or domain user is not a member of Administrators group.

User must have the following permissions to the system drive and agent installation drive

1. Read
2. Write
3. Execute
4. Modify

User must have the following permission to the HKEY_LOCAL_MACHINE registry key

- Read

User must be a member of following groups on monitored server

1. Users
2. IIS_IUSRS
3. Performance Monitor Users

4. Performance Log Users

Note: However, it is advisable to run Microsoft .NET agent with a local or domain user that is a member of local Administrators group.

Registering the data collector

You must register the data collector to collect the transaction tracking and diagnostics data. To collect the resource monitoring data, no specific configuration is required.

About this task

Register the following components of data collector depending on the transaction, diagnostics, or both types of data that you want the data collector to collect:

Table 182. Data collector components and their functions

Component name	Monitors
httpmodule	ASP.NET transactions and collects request response time and CPU time
profiler	ADO.NET transactions and collects method, stack trace, and request context data for diagnostics
isapi	ASP.NET transactions and collects request response time and CPU time
soap	ASMX or WCF service transactions and WCF services response time

Remember:

- Use `isapi32` to filter the 32-bit applications on a 64-bit Microsoft IIS Server.
- Register all the components to track all transactions and view the complete transaction topology.

Procedure

1. On the server where the agent is installed, run the following command as an administrator:

   ```
   cd install_dir\qe\bin
   configdc.exe registerdc [all|isapi|isapi32|profiler|httpmodule|soap]
   ```

   Remember:

   - When you run the **`configdc.exe registerdc`** command without specifying any components to register, only httpmodule is registered.
   - To register all the components, run the **`configdc.exe registerdc all`** command.
   - To register any of the components together, run this command: **`configdc.exe registerdc`** *component_name component_name*. For example, **`configdc.exe registerdc`** httpmodule profiler
2. Restart the .NET applications.

What to do next

After you register the data collector, you must enable the data collection for transaction tracking and diagnostics. For information about enabling data collection, see “Enabling collection of transaction tracking and diagnostics data” on page 525.

If you want to stop monitoring .NET applications, unregister the data collector. Repeat the specified steps by using the **`configdc.exe unregisterdc`** command to unregister all the components of data collector.

Using the IIS Response Time module of the .NET agent

From 8.1.4.0.2 release onwards, the .NET agent includes the "IIS Response Time module", which works with the Response Time agent to show end user transactions data for the IIS server.

Enabling Response Time module

You need to enable Response Time module before using it.

Procedure

Complete the following steps to enable Response Time module:

1. Open the command prompt in administrator mode.
2. To stop IIS, run the following command:

   ```
   iisreset /stop
   ```
3. Go to `install_dir\qe\bin` directory on the command prompt.
4. To register the Response Time module for IIS, run the following command:

   ```
   configdc registerdc rtmodule
   ```
5. To start IIS, run the following command:

   ```
   iisreset /start
   ```

Results

The Response Time module is enabled.

Configuring the Response Time agent to work with the .NET agent’s IIS Response Time module

You need to configure the Response Time agent to work with the .NET agent's IIS Response Time module.

Before you begin

Install the Response Time agent (8.1.4 version), for more information see Chapter 6, “Installing your agents,” on page 121.

Procedure

Complete the following steps, to configuring the Response Time agent to work with the .NET Agent’s IIS Response Time module:

1. Open a text editor in administrator mode.
2. Open the following file in the text editor:

   config_dir\TMAITM6_x64\host_name_T5.config

   where *config_dir* is the APM home and *host_name* is the name of the server.
3. Update the following property:

   ```
   { KT5DISABLEANALYZER=YES } { KT5ENABLEWEBPLUGIN=YES }
   ```
4. Add the following property in the `SECTION=analyzerconfig [ ]` section:

   ```
   {KT5WEBPLUGINIPCNAME=KFC1}
   ```
5. Restart the Response Time agent.
6. Log in to the Performance Management console to verify the data that is collected by the agent in the dashboards. For information about using the Performance Management console, see “Starting the Cloud APM console” on page 957.

Configuring JavaScript Injection for IIS Response Time Module

You must configure JavaScript (JS) Injection to work with the .NET agent's Internet Information Services (IIS) Response Time module.

Procedure

To configure JS Injection to work with the .NET agent's IIS Response Time module, follow these steps:

1. Open a text editor in administrator mode.
2. Open the following file in the text editor:

   ```
   <APM_HOME>\qe\config\dotNetDcConfig.properties.inactive
   ```
3. To enable JS Injection for the Response Time module, set the **`RTModule.JSInjection.Enabled`** property to **`true`**.
4. To disable JS Injection for the Response Time module, set the **`RTModule.JSInjection.Enabled`** property to **`false`**.
5. Open the command prompt in administrator mode and go to the `<APM_HOME>\qe\bin` directory.
6. Run the following commands:
   - `configdc activateconfig`
   - `iisreset`

Disabling the IIS Response Time Module

You can disable the IIS Response Time module when you don't want to see end user transactions data for the IIS server.

Procedure

Complete the following steps to disable the IIS response time module:

1. Open the command prompt in administrator mode.
2. To stop IIS, run the following command:

   ```
   iisreset /stop
   ```
3. Go to `*install_dir*\qe\bin` directory on the command prompt.
4. To unregister Response Time module for IIS, complete the following steps:
   - To unregister response time module for IIS, run the following command:

     ```
     configdc unregisterdc rtmodule
     ```
   - To unregister all the components of data collector including the Response Time module, run the following command:

     ```
     configdc unregisterdc all
     ```
5. To start IIS, run the following command:

   ```
   iisreset /start
   ```

Results

The IIS response time module is disabled.

Limitations of IIS Response Time module

Limitations of IIS Response Time module are listed here.

- The user information is not tracked by the IIS Response Time module and user name currently appears as "Unknown".

Enabling collection of transaction tracking and diagnostics data

On the **Agent Configuration** page, you can enable or disable the collection of transaction tracking and diagnostics data.

Before you begin

Ensure that you have registered the data collector. For details, see “Registering the data collector” on page 522.

Procedure

Complete the following steps to configure the data collection for each managed system.

1. Log in to the Cloud APM console.
2. From the navigation bar, click **System Configuration** > **Agent Configuration**.

   The **Agent Configuration** page is displayed.
3. Click the **MS .NET** tab.
4. Select the check boxes of the managed systems for which you want to configure the data collection and complete any of the following actions from the **Actions** list.
   - To enable transaction tracking, click **Set Transaction Tracking** > **Enabled**. The status in the **Transaction Tracking** column is updated to Enabled for each selected managed system.
   - To enable the diagnostic data collection, select **Set Diagnostic Mode** and click the level that you want to set. The status in the **Diagnostic Mode** column is updated to display the specified level for each selected managed system.
     - Level 1: The HTTP module collects the request summary and request instance data.
     - Level 2: The HTTP module collects the request summary and request instance data. The profiler collects the method data and stack trace data.
   - To disable transaction tracking, click **Set Transaction Tracking** > **Disabled**. The status in the **Transaction Tracking** column is updated to Disabled for each selected managed system.
   - To disable diagnostic data collection, click **Set Diagnostic Mode** > **Disabled**. The status in the **Diagnostic Mode** column is updated to Disabled for each selected managed system.

Results

The data collection is configured for each managed system.

What to do next

Log in to the Cloud APM console to view the transaction tracking and diagnostics data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see “Starting the Cloud APM console” on page 957.

Enabling the collection of diagnostics data by using the configdc command

You can also enable or disable the collection of diagnostics data by using the **`configdc`** command. This process is optional.

Before you begin

- Ensure that you have registered the data collector. For details, see “Registering the data collector” on page 522.
- Ensure that you have completed the process at “Enabling collection of transaction tracking and diagnostics data” on page 525.
- Ensure that the `qe_custom.properties` file is processed by the APM server at `<APM_Home>\localconfig\qe` and has following properties:
  - transaction_tracking=ENABLED

- diagnostic_mode=LEVEL2

Procedure

1. Run the following command:

   ```
   cd install_dir\qe\bin configdc deepdivedc -tracelevel trace_level
   ```

   Where,

   install_dir
   The installation directory of the Microsoft .NET agent.

   trace_level
   The trace level that indicates the amount of diagnostics data that the .NET Data Collector collects. Specify one of the following values:

   0
   The collection of the diagnostics data is disabled.

   1
   The collection of the diagnostics data is enabled. The HTTP module collects the request summary and request instance data.

   2
   The collection of the diagnostics data is enabled. The HTTP module collects the request summary and request instance data. The profiler collects method data and stack trace data.

   Tip: When you set the trace level by using the **`configdc.exe deepdivedc -tracelevel`** command, the value of the `bci_dc.diagnose.level` parameter is set in the `dotNetDcConfig.properties` file.

2. Activate the configuration changes.

   For information about activation of changes, see "Activating the configuration updates" on page 527.

What to do next

Log in to the Cloud APM console to view the diagnostics data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Enabling transaction tracking in agent coexistence environment

In an agent coexistence environment, you can configure the data collector to collect and pass the transaction tracking data to the Tivoli Enterprise Portal, which is a component of IBM Tivoli Monitoring.

Before you begin

You must install the Microsoft .NET agent, which is delivered as part of Cloud APM and you must remove or unregister the .NET Data Collector component, which is delivered with ITCAM for Microsoft Applications. Use the **`configdc.exe unregisterdc`** command to unregister all modules of the data collector.

Procedure

To configure the data collector to collect and pass the transaction tracking data to the Tivoli Enterprise Portal, complete the following steps:

1. Go to the *`install_dir`*`\localconfig\qe` directory, where *install_dir* is the installation directory of the Microsoft .NET agent. The default path is `C:\IBM\APM`.
2. Open the `qe_default.properties` file and set the value of the **`transaction_tracking`** parameter to `ENABLED`.
3. Save and close the `qe_default.properties` file.
4. Go to the *`install_dir`*`\qe\config` directory.
5. Open the `dotNetDcConfig.properties.inactive` file in a text editor.

6. Set the **TTDC.enabled** and **TTAS.enabled** parameters as follows:

   ```
   TTDC.enabled=true
   TTAS.enabled=true
   ```

7. To configure the connection to the Transaction Collector, set the values of the **TTAS.Host** and **TTAS.Port** parameters to the IP address and port number of the Transaction Collector.
8. Run the following command to activate the changes:

   ```
   install_dir\qe\bin\configdc.exe activateconfig
   ```

9. Restart the .NET application for the changes to take effect.

Results

Now, the transaction tracking data can be collected and displayed on the Tivoli Enterprise Portal.

What to do next

To disable the transaction tracking for a .NET Data Collector, repeat the procedure and use the following configuration values:

- In the `qe_default.properties` file, set `transaction_tracking=DISABLED`.
- In the `dotNetDcConfig.properties.inactive` file, set `TTDC.enabled=false` and `TTAS.enabled=false`.

Activating the configuration updates

You must activate the updates that you make in the configuration settings by using the `configdc` command. The activation ensures that your updates are saved to the `dotNetConfig.properties` file.

About this task

When you update the configuration settings by using the **configdc** command, the parameter values are updated in the `dotNetConfig.properties` file. However, if this file is being used and cannot be modified, your configuration setting updates are saved in the `dotNetDcConfig.properties.inactive` file. You must activate the configuration so that the updates are saved in the `dotNetConfig.properties` file.

Procedure

1. Go to the following path:
   install_dir`\qe\bin`
   Where *install_dir* is the installation directory of the Microsoft .NET agent.
2. Run the following command:
   `configdc activateConfig`

What to do next

If the Internet Information Service (IIS) transactions are monitored and the data collector configuration is updated, restart the IIS to activate the configuration.

If ASMX or WCF web services are monitored and the data collector configuration is updated, restart the process that hosts the web service.

Performance tuning of data collector

When you configure the data collector to collect the transaction tracking and diagnostics data, the performance of data collector is affected. To improve the performance, you can complete some performance-tuning tasks.

You might need to complete the following tasks to improve the performance of data collector:

- Filter the ADO.NET interfaces that you want to monitor.

- Sample the transaction tracking and diagnostics data.
- Configure trace logging.

Specifying ADO.NET interfaces for monitoring

You can specify the ADO.NET client interfaces that you want to enable for transaction tracking.

Before you begin

If you want to view the ADO.NET interfaces that are supported by the .NET Data Collector, see Functions of namespaces supported by the data collector.

To view the configuration values of the .NET Data Collector, see the `dotNetDcConfig.properties` file in the *install_dir*`\qe\config` directory, where *install_dir* is the installation directory of the Microsoft .NET agent.

About this task

By default, all the supported ADO.NET interfaces are enabled for transaction tracking during the installation of the agent. To specify the interfaces that the data collector must monitor, enable or disable monitoring for specific interfaces.

If you disable the monitoring of an interface, the settings of any associated application domain filter remain in the data collector configuration file. The filter is retained when the interface is enabled again.

Procedure

- To enable the monitoring of an ADO.NET interface, complete these steps:
  a) From the *install_dir*`\qe\bin` directory, run the following command:

  ```
  configdc enableMonitor all | adsi | db2 | ldap | odbc | oledb | oracle | sql
  | http | web
  [-appdomain appdomain filter list]
  ```

  b) Activate the configuration changes.

  For information about activation of changes, see "Activating the configuration updates" on page 527.
- To disable the monitoring of an ADO.NET interface, complete these steps:
  a) From the *install_dir*`\qe\bin` directory, run the following command:

  ```
  configdc disableMonitor all | adsi | db2 | ldap | odbc | oledb | oracle | sql
  | http | web
  ```

  b) Activate the configuration changes.

  For information about activation of changes, see "Activating the configuration updates" on page 527.

Sampling transaction tracking and diagnostic data

If the system performance is affected due to transaction tracking or diagnostic data collection, you can enable sampling of the collected data to improve the performance.

About this task

When the system performance suffers due to the transaction tracking and diagnostic data collection, you can configure the data collector to periodically collect data through sampling. When the sampling is enabled, the data collector does not collect data for every request, but at intervals of several requests. You can change the sampling rate dynamically according to the CPU usage of the DotNetProfilerService process.

⚠ **CAUTION:** However, sampling can save system resources sampled data might not be efficient for diagnosing problems. After data sampling is enabled, the transaction tracking topology might be broken or lost. Therefore, enable data sampling only when the performance is seriously affected.

Procedure

To enable sampling on the transaction tracking and diagnostic data collection, complete the following steps:

1. Go to the following directory:
   `install_dir\qe\config`
   Where *install_dir* is the installation directory of the Microsoft .NET agent.
2. In a text editor, open the `dotNetDcConfig.properties.inactive` file.
3. Set the following parameters in the file:

   bci_dc.sampling.Enabled
   Specifies whether the data collector periodically collects the transaction tracking and diagnostic data. Valid values are `true` and `false`.

   bci_dc.sampling.base
   Specifies the base for data sampling. A valid value is a positive number. For example, if you set the value of the **`bci_dc.sampling.base`** parameter to 10, the data collector collects the transaction tracking and diagnostic data every 10 requests. The sampling rate is 1 out of 10 requests. The data collector collects data for 1st, 11th, 21st, 31st, and other requests.

   bci_dc.dynamic.sampling
   Specifies whether the sampling rate is constant or dynamic. Valid values are on and `off`. When you set the value of the **`bci_dc.dynamic.sampling`** parameter to on, the sampling rate is dynamically adjusted according to the value of the **`bci_dc.dynamic.max_cpu_usage`** parameter.

   bci_dc.dynamic.max_cpu_usage
   Specifies the CPU usage threshold for the DotNetProfilerService process. If the CPU usage of the DotNetProfilerService process is greater than 110% of the specified value, the sampling rate is decreased. If the CPU usage is less than 90% of the specified value, the sampling rate is increased. A valid value is in the range 1 - 100.
4. Save and close the `dotNetDcConfig.properties.inactive` file.
5. Run the following command to activate the changes:

   ```
   install_dir\qe\bin\configdc.exe activateconfig
   ```
6. Restart the .NET application for the change to take effect.

Enabling trace logging for the data collector

You can enable the generation of trace logs for the data collector. You can use these trace logs to troubleshoot problems that might occur with the collection of transaction tracking and diagnostics data.

About this task

To collect logs for ASP.NET transactions, ADO.NET transactions, and diagnostics data, enable trace logs for the httpmodule, profiler, and isapi components of data collector. To collect logs for ASMX and WCF transactions, enable trace logs for the soap component of data collector.

Important: The performance of data collector might be affected when the trace logging is enabled. Therefore, disable the trace logging after the required trace logs are collected.

Procedure

1. On the server where the agent is installed, navigate to the following path:
   `install_dir\qe\bin`

   Where, *install_dir* is the installation directory of the Microsoft .NET agent.
2. Complete both or any of the following procedures depending on the trace logs that you want to enable:

   - To enable trace logs for the httpmodule, profiler, soap components, and response time module, complete the following steps:

     a. Run the following command:

`configdc logging -tracing on`

b. Restart IIS and .NET applications.

- To enable trace logs for the BCI engine, complete the following steps:

  a. Navigate to the following path:
     `<APM_HOME>\qe\config`

  b. In a text editor, open the `dotNetDcConfig.properties.inactive` file.

  c. For the property **`bci_dc.trace.logging`**, specify the value as on.

  d. Run the following command:
     `configdc activateconfig`

  e. Restart IIS.

What to do next

To disable trace logs, do the following:

- To disable trace logs for the httpmodule, profiler, soap components, and response time module:
  - Run the following command:
    `configdc logging -tracing off`
  - Restart IIS and .NET applications.
- To disable trace logs for the BCI engine:
  - Go to the following path:
    `<APM_HOME>\qe\config`
  - In a text editor, open the `dotNetDcConfig.properties.inactive` file.
  - For the **`bci_dc.trace.logging`** property, specify the value as `off`.
  - Run the following command:
    `configdc activateconfig`
  - Restart IIS and .NET applications.

Configuring Microsoft Office 365 monitoring

You must configure the Microsoft Office 365 agent to monitor the availability and performance of Microsoft Office 365 subscriptions of the organization.

Before you begin

- Review the hardware and software prerequisites.
- To collect data for Office 365 users, the following modules must be installed on the Windows client where the agent is installed:
  - PowerShell 3.0 or later
  - Microsoft Online Services Sign-In Assistant PowerShell
  - SharePoint Online Management Shell
  - DotNetFrameworkVersion 4.5.2 or later

  A user, who configures the agent, must have administrative privileges along with privileges to enable the remote execution policy of PowerShell.
- To monitor Skype synthetic transactions, complete the following tasks:
  - Install the Skype 2013 client on the Windows client where the user wants to complete the synthetic transactions for Skype.
  - Set the default video device for Lync and Skype as a virtual audio video filter.

- Ensure that the user, who starts the Microsoft Office 365, has administrator privileges. Use an existing user with administrator privileges, or create a new user. Assign administrator privileges to the new user by adding the new user to the Administrators group.

Review hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Microsoft Office 365 agent.

About this task

The product version and the agent version often differ. The directions here are for the most current release of this agent. For more information about how to check the version of an agent in your environment, see Microsoft Office 365 agent. To access the documentation for V1.0.0, see the IBM Cloud Application Performance Management Knowledge Center.

You can start the Microsoft Office 365 agent after the agent is installed. However, manual configuration is required to view data for all the agent attributes.

To configure the agent, you can either use the IBM Cloud Application Performance Management window or the silent response file.

Verifying reachability of configured users

To verify reachability, the Microsoft Office 365 agent sends an email message to the configured users, and measures the amount of time to receive an automated response. Before you start the agent, you must configure all the users, which are configured in the Office 365 agent mailbox reachability setting, to automatically respond to email messages.

Before you begin

Before you configure the Exchange Online users for reachability, ensure that the following tasks are completed:

- A mailbox is created for each user on the Exchange Online that you want to monitor.
- The user that you created for the agent is a global Office 365 user.

Procedure

Complete the following steps for each Exchange Online user account for which you want to verify reachability:

1. Log in to Microsoft Outlook by specifying credentials of the user that you created.
2. Click **Tools > Rules and Alerts > New Rule**.
3. In the **Rules wizard**window, under **Start from a blank rule**, click **Apply rule on messages I receive** and click **Next**.
4. Select the following options:
   - **From people or public group**
   - **With specific words in the subject**
5. Under **Step 2** in the window, click **people or public group**.
6. In the **Rule address** window, select the user (global administrator) from which the messages are to be received and click **Next**.
7. Under **Step 2** in the window, click **Specific words**.
8. In the **Specify words or phrases to search for in the subject or body** field, type `Test Reachability`.
9. Click **Add**.
10. Click **OK** and then click **Next**.
11. Select **Have the server reply using a specific message** and click **a specific message**.
12. In the email message editor, type the following text in the subject field of the message:

    `Test Reachability`.

13. In the **To** list, add the global administrator.
14. Close the email message editor and click **Yes** to save these changes.
15. Click **Finish**.
16. Click **Apply** and then click **OK**.

What to do next

Configure the Microsoft Office 365 agent.

For help with troubleshooting, see the Troubleshooting section.

Configuring the agent on Windows systems

You can configure the Microsoft Office 365 agent on Windows operating systems by using the IBM Cloud Application Performance Management window. After you update the configuration values, you must start the agent to save the updated values.

About this task

You can configure the agent when the agent is running or stopped. The agent remains in the same state after configuration. For example, if the agent is running, it remains in the running state after configuration.

The Microsoft Office 365 agent provides default values for some parameters. You can specify different values for these parameters.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Monitoring Agent for Microsoft Office 365**, and click **Configure Agent**.
3. In the **Monitoring Agent for Microsoft Office 365** window, complete the following steps:
   a) On the **Office365 Subscription Details** tab, enter the user name and password of the Office 365 global administrator, and click **Next**.
   b) On the **Synthetic Transaction** tab, enter the list of email addresses that are delimited by semicolons in the **Reachability Email Addresses** field.
   c) To enable the data collection of Skype QoS metrics, select the **Skype QoS** check box, and click **Next**.
   d) On the **Mailbox and OneDrive Usage Monitoring** tab, select the duration for the collection interval in hours from the **Collection Interval** list,, and click **Next**.
4. In the **Monitoring Agent for Microsoft Office 365** window, click **Yes**.

What to do next

- Configure the Skype synthetic transaction utilities to monitor the Skype QoS synthetic transactions. For more information about monitoring the Skype QoS, see “Monitoring the Skype QoS” on page 535.
- Change the user account from the local user to the domain user. For details, see “Changing the user account” on page 533.
- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see “Starting the Cloud APM console” on page 957.
- For help with troubleshooting, see the Troubleshooting section.

Configuring the agent by using the silent response file

When you install the Microsoft Office 365 agent, the agent must be configured and started manually after providing the configuration settings. Use the silent response file to configure the custom settings.

Before you begin

Edit the response file to modify the following default configuration settings:

KMO_USER_NAME
The user name of the Office 365 global administrator.

KMO_PASSWORD
The password of the Office 365 global administrator.

KMO_MAIL_ADDRESSES1
A list of email addresses to be targeted for verifying mailbox reachability. The list of email addresses must be delimited using semicolons.

KMO_SKYPE
This parameter is used to enable the collection of the Skype QoS synthetic transactions.

KMO_DATA_COLLECTION_DURATION
The duration (in hours) for which the agent waits before fetching the mailbox and OneDrive usage data.

The response file is available at the following location:
`<CANDLEHOME>\samples`

About this task

You can configure the agent when the agent is running or stopped. The agent remains in the same state after configuration. For example, if the agent is running, it remains in the running state after configuration.

Procedure

To configure the Microsoft Office 365 agent, complete the following steps:

1. On the command prompt, change the path to the directory that contains the `microsoft_office365-agent.bat` file.
2. Enter the following command: `microsoft_office365-agent.bat` *absolute path to the response file*.

   The response file contains the following parameters:
3. If the agent is in the stopped state, start the agent.

What to do next

- Configure the Skype synthetic transaction utilities to monitor the Skype QoS synthetic transactions. For more information about monitoring the Skype QoS, see "Monitoring the Skype QoS" on page 535.
- Change the user account from the local user to the domain user. For details, see "Changing the user account" on page 533.
- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.
- For help with troubleshooting, see the Troubleshooting section.

Changing the user account

After you configure the Microsoft Office 365 agent, change the user account from the local user to the domain user.

About this task

By default, the Microsoft Office 365 agent runs under the local user account.

Procedure

1. Run the following command to verify which user ID is being used for starting the agent:

   ```
   install_dir\InstallITM\KinCinfo.exe -r
   ```
2. If the monitoring agent was started with a user ID that does not belong to the Administrator group, stop the agent.
3. Open the **Manage Monitoring Services** window.
4. In the **Manage Monitoring Services** window, right-click the agent instance, and click **Change Startup**.
5. Specify the fully qualified user ID as <Domain\User ID>, and then specify the password.
6. Start the Microsoft Office 365 agent.

Validating and granting access to the user

To safeguard against unauthorized access to the MS Office 365 application that the agent monitors, you need to validate the user and grant access to the MS Office 365 application.

Before you begin

Ensure that the user, who starts Microsoft Office 365, has the subscription ID and administrator rights.

About this task

You can validate the user and grant application access by adding the following user details to the agent configuration file:

- Tenant ID
- Client ID
- Secret ID

Procedure

To validate and grant access to the user, follow these steps:

1. Log in to Microsoft Office 365 by specifying your subscription credentials.
2. Click Microsoft 365 admin center. The Microsoft 365 admin center page opens.
3. In the left pane, click **Azure Active Directory**. The Azure Active Directory admin center page opens.
4. Click **App registration**.
5. Click **New registration** and enter any name of the application, for example, Office365API, and select the **Supported account type** and click **Register**.
   After registration, you will get the Application (client) ID and Directory (tenant) ID.
6. To generate the secret key, click **Certificates & Secrets> New Client Secrete**. The Add a client secret window opens.
7. Enter the **Description** name, select **Expires** option as **Never**, and click **Add**. You get a Secret key. Copy the secret key for user configuration as it is stored in the encrypted format later.
8. To give permissions to access the API, click **API permissions > Add a permission > Office 365 Management APIs**, and select **Application permissions**.

   Note: Applications are authorized to call APIs when they are granted permissions by users or administrators as part of the consent process.
9. Select the required permission and click **Add permissions**.
10. To provide grant admin for APIs, click **Grant admin consent**.
11. Click **Yes** when you see the status as Granted for domain.
12. To add user details, such as Client ID, Tenant ID, and Secret ID to the configuration file, follow these steps:

    a) Go to the agent installer folder, for example, `<APM Home\TMAITM6_x64`.

b) In the installer folder, open the `kmoOffice365CDP.exe.config` file and add the Client ID, Tenant ID, and Secret ID values that are generated from the Azure portal as mentioned in steps 1 - 8.
For example,

```
<add key="Office365ServiceAPIConnectionServiceUrl" value="https://manage.office.com/api/
v1.0/#TenantID#/ServiceComms" />
<add key="AuthURL" value="https://login.microsoftonline.com/#TenantID#/oauth2/v2.0/
token"/>
<add key="Client_id" value="702ce315-f8dd-4775-91d9-c7c7ec376835"/>
<add key="Client_secret" value="42_mW-xp-2xLkv~2dH8sDj6.wURkaok0re"/>
```

13. Save the `kmoOffice365CDP.exe.config` file.
14. If the user has multi-factor authentication (MFA), then skip the authentication by disabling the second-level (Mobile/ App) authentication from the Azure portal. To disable the second-level authentication, follow these steps:

    a) Go to Microsoft 365 admin center.

    b) Click **Azure Active Directory > Properties > Manage Security default > Enable Security defaults**.

    c) Select **No**, and click **Save**.
15. Reconfigure the agent.

Monitoring the Skype QoS

To monitor the Skype QoS, a user must configure the Skype synthetic transaction utilities, `kmoskypecaller.exe` and `Kmoskypereceiver.exe` on the Windows client where the agent is installed or a distributed environment where the Skype for Business client is configured.

Before you begin

To perform synthetic transactions, you must update the Skype caller name and the Skype receiver name in the `<CANDLEHOME>\tmaitm6_x64\kmoskypecallerlist.properties` file in the following format:
`skype caller = skype receiver`
For example, john@xyz.com = alan@xyz.com
You can add multiple Skype call receivers for a single Skype caller in the following format:
`skype caller = list of skype receiver`
For example, john@xyz.com = alam@xyz.com;bill@xyz.com;chuk@xyz.com

Remember: If you do not want to perform synthetic transactions but want to monitor the Skype QoS for real time users, updating the `<CANDLEHOME>\TMAITM6_x64\kmoskypecallerlist.properties` file is not required.

About this task

When the Microsoft Office 365 agent is configured and started, the following files and folders are created at `<CANDLEHOME>\TMAITM6_x64\`:

- `kmoskypecaller.properties`
- `kmoskypecallerlist.properties`
- `KMOSynthTransSkype.zip`
- `KMOSkypeTransReceiver.zip`

Also, the `kmoskypecaller.properties` file is updated with the server IP and port that are used for communication between the agent and the kmoskypecaller utility.

Procedure

To configure the Skype caller and Skype receivers and initiate synthetic transactions, such as instant messaging, audio and video calls, and application sharing sessions, complete the following steps:

1. Start the Office 365 agent.
2. Copy the `KMOSynthTransSkype.zip` file from the agent client to the Windows client from where the Skype call is to be initiated.
3. Extract the `KMOSynthTransSkype.zip` file.
4. Copy the `kmoskypecaller.properties` file from the agent client to the extracted `KMOSynthTransSkype` folder on the Windows client from where the Skype call is to be initiated..
5. Copy the `KMOSkypeTransReceiver.zip` file from the agent client on all Windows clients where Skype calls must be received.
6. Extract the `KMOSkypeTransReceiver.zip` file on all Windows clients where the Skype calls must be received, and run `KMOSkypeTransReceiver.exe` to start receiving messages.
7. To initiate the synthetic transactions, run the `KMOSynthTransSkype.exe` file, that is available in the extracted `KMOSynthTransSkype` folder on the Windows client. The Office 365 agent would start receiving the skype monitoring data from the caller client.

Results

The agent starts monitoring the Skype QoS.

Configuring local environment variables

You can configure local environment variables to change the behavior of the Microsoft Office 365 agent.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, from the **Actions** menu, click **Advanced > Edit ENV File**.
3. In the environment variable file, enter the values for the environment variables.

   For more information about the environment variables that you can configure, see "Local environment variables" on page 536.

Local environment variables

You can change the behavior of the Microsoft Office 365 agent by configuring the local environment variables.

Variables for defining the data collection method for the agent

To set the method for data collection of the agent, use the following environment variables:

- **`CDP_DP_INITIAL_COLLECTION_DELAY`**: Use this variable to set the time interval (in seconds) after which the thread pool begins its data collection.
- **`KMO_MAILBOX_REACHABILITY_INTERVAL`**: Use this variable to set the data collection interval (in minutes) for mailbox reachability attribute group.
- **`KMO_SKYPE_REPORT_INTERVAL`**: Use this variable to set the data collection interval (in hours) for Skype for Business usage statistics feature.
- **`KMO_SERVICE_API_INTERVAL`**: Use this variable to set the data collection interval (in minutes) for Office 365 service health feature.
- **`KMO_NETWORK_CONNECTION_INTERVAL`**: Use this variable to set the data collection interval (in minutes) for internet connectivity feature.
- **`KMO_NETWORK_PERFORMANCE_INTERVAL`**: Use this variable to set the data collection interval (in minutes) for Office 365 services network performance feature.
- **`KMO_SITE_CONNECTION_INTERVAL`**: Use this variable to set the data collection interval (in minutes) for Office 365 connectivity feature.
- **`KMO_SPSITE_COLLECTION_INTERVAL`**: Use this variable to set the data collection interval (in minutes) for SharePoint Sites details feature.

- **KMO_UASGE_STATS_INTERVAL**: Use this variable to set the data collection interval (in hours) for Office 365 Services usage and user statistics feature.
- **KMO_TENANT_INTERVAL**: Use this variable to set the data collection interval (in minutes) for Office 365 tenant details feature.
- **KMO_ONEDRIVE_CONNECTIVITY_INTERVAL**: Use this variable to set the data collection interval (in minutes) for Office 365 OneDrive connectivity feature.
- **KMO_TENANT_DOMAIN**: Use this variable to set the domain name of the tenant.

Configuring Microsoft SharePoint Server monitoring

When you install the Monitoring Agent for Microsoft SharePoint Server, the agent is automatically configured and started with the default configuration settings. Use the silent response file to modify the default configuration settings.

Before you begin

Ensure that you complete the following tasks:

- Ensure that the user, who connects to the Microsoft SharePoint Server environment or application, has administrator privileges. Use an existing user with administrator privileges, or create a new user. Assign administrator privileges to the new user by adding the new user to the Administrators group.

  Remember: To configure the Microsoft SharePoint Server agent, you can use a local or a domain user provided that the user has administrator privileges.
- Edit the response file and modify the default configuration parameters.

  The response file contains the following parameters:

  KQP_DB_User
  The user ID of the database.

  KQP_DB_Password
  The password of the database.

Review hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Microsoft SharePoint Server agent.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version. To access the documentation for earlier agent releases, see the following table:

Table 183. Agent versions and documentation

Microsoft SharePoint Server agent version	Documentation
06.31.09.00, 06.31.10.00	IBM Cloud Application Performance Management
06.31.09.00	IBM Performance Management 8.1.3 **Note:** The link opens an on-premises Knowledge Center topic.
06.31.07.00	IBM Performance Management 8.1.2 **Note:** The link opens an on-premises Knowledge Center topic.

Procedure

To configure the Microsoft SharePoint Server agent, complete the following steps:

1. Open the command prompt.
2. Change the path to the directory that contains the `ms_sharepoint_server-agent.bat` file.

3. Enter the following command: **`ms_sharepoint_server-agent.bat config`** *absolute path to the response file*
4. If the agent is in the stopped state, start the agent.

What to do next

After you configure the agent, you can change the user account from the local user to the domain user. For steps to change the user account, see "Changing the user account" on page 538.

Changing the user account

After you configure the Microsoft SharePoint Server agent, you can change the user account from the local user to the domain user.

About this task

With the domain user, the agent can monitor all the components of the Microsoft SharePoint Server agent.

Procedure

To change the user account, complete the following steps:

1. Run the following command to verify which user ID is being used for starting the agent.
   `install_dir\InstallITM\KinCinfo.exe –r`
2. If the monitoring agent was started with a user ID that does not belong to the Administrator group, stop the agent.
3. Open the **Manage Monitoring Services** window.
4. Right-click the agent instance, and click **Change Startup**.
5. Specify the fully qualified user ID as `<Domain\Userid>`, and then specify the password.
6. Start the monitoring agent.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Running Monitoring Agent for Microsoft SharePoint Server by a non-admin user

Local security policies are available to run a Monitoring Agent for Microsoft SharePoint Server by a non-admin user.

About this task

A combination of following two local security policies works to run the Microsoft SharePoint Server agent by a non-admin-user.

1. Debug Programs.
2. Log on as Service.

Follow the procedure that is given to avail the Local Security permissions for a non-administrator user.

Procedure

1. Go to TEMA and change the Microsoft SharePoint Server agent startup with non-admin user.
2. Add non-admin user under Registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office Server` directory and give read access to it.
3. Add non-admin user under Registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Shared Tools\Web Server Extensions` and give read access to it.
4. Add non-admin user manually under Registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Shared Tools\Web Server Extensions\16.0\Secure\` and give read access to it.

5. Add non-admin user under Registry key `HKEY_LOCAL_MACHINE\SOFTWARE\IBMMonitoring` directory and give full permissions to it.
6. Add non-admin user under Registry key `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib` directory and give read access to it.
7. Add non-admin user in SharePoint Agent installation folder (Candle folder for example, `C:\IBM\APM`) and give full permissions to it.
8. Run the **`secpol.msc`** command in **startmenu** to open the **Local Security Policy**.
9. Add non-admin user in Local Security Policy refer "Local Security Policy permissions" on page 539
10. Add non-admin user in the SQL Server Login user group. The user must have sysadmin SQL Server role permissions on the SQL Server.
11. Restart the Microsoft SharePoint Server agent.
12. Check Microsoft SharePoint Server agent status and Verify the data on APM portal.
13. The following attribute groups show data for users who are members of the Administrators group.
    a) Availability
    b) Web Service

Local Security Policy permissions

Local security policies are available to run a Microsoft SharePoint Server agent by a non-admin user. These policies help to start or stop, configure, and do data verification of the agent. Following two local security policies work to run the Microsoft SharePoint Server agent by a non-admin-user.

Granting Log on as Service permission

You can grant the Log-on as service permission.

About this task

To grant the Log-on as service permission, follow the procedure on Microsoft SharePoint Server agent as described here.

Procedure

1. Click **Start** > **Administrative Tools** > **Local Security Policy**. The **Local Security Settings** window opens.
2. In the navigation pane, expand **Local Policy** and click **User Rights Assignment**. The list of user rights opens.
3. Double-click **Log-on as service** policy. The **Log-on as service Properties** window opens.
4. Click **Add User or Group**. The **Select Users or Groups** window opens.
5. In the **Enter the object names to select** field, enter the user account name to whom you want to assign permissions, and then click **OK**.
6. Click **OK**.

Granting Debug Programs permission

You can grant the Debug Programs permission.

About this task

To grant the Debug Programs permission, follow the procedure on Microsoft SharePoint Server agent as described here:

Procedure

1. Click **Start** > **Administrative Tools** > **Local Security Policy**. The **Local Security Settings** window opens.
2. Expand **Local Policy** and click **User Rights Assignment**. The list of user rights opens.
3. Double-click **Debug Programs** policy. The **Debug Programs Properties** window opens.

4. Click **Add User or Group**. The **Select Users or Groups** window opens.
5. In the Enter the object names to select field, enter the user account name to whom you want to assign permissions, and then click **OK**.
6. Click **OK**.

Configuring Microsoft SQL Server monitoring

You must configure the Monitoring Agent for Microsoft SQL Server so that the agent can collect data from the application that is being monitored.

Before you begin

Review the hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Microsoft SQL Server agent.

You can install and configure the Microsoft SQL Server agent locally by using the command prompt interface. Ensure that the agent is installed on the server that is being monitored.

About this task

The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see “Change history” on page 54.

The Microsoft SQL Server agent is a multiple instance agent; you must configure and start each agent instance manually.

- To configure the agent, complete the following tasks:
  - Create a user and grant the required permissions
  - Select the databases for monitoring
  - Configure the local environment variables
- To run the agent in a cluster environment, complete the steps that are described in the "Running the agent in a cluster environment" topic.

Creating a user and granting permissions

On the Microsoft SQL Server, you must create a user under which the agent runs, and grant permissions to the user for monitoring Microsoft SQL Server. The process of granting permissions is the same for Microsoft SQL Server 2005, or later.

Before you begin

Install the Microsoft SQL Server agent. To create a user and grant permissions to the user, you must be a database administrator with the sysdamin authorization role.

About this task

Use the following procedure to determine if an existing SQL Server user has sufficient permissions to monitor Microsoft SQL Server:

- Windows “Checking the permissions of an existing SQL Server user” on page 541

Use one of the following procedures to create a user:

- Windows “Creating a SQL Server user ID with Windows authentication” on page 542
- Linux Windows “Creating a SQL Server user ID with SQL Server authentication” on page 542

Use the following procedure to grant permissions:

- Windows “Granting minimum permissions for data collection” on page 543

- Windows “Granting permission to the Perflib registry key for collecting data for few data sets” on page 544

Checking the permissions of an existing SQL Server user

Windows You can run the utility tool **koqVerifyPerminssions.exe** to check if an existing SQL Server user has sufficient permissions related to SQL Server databases.

About this task

The utility tool **koqVerifyPerminssions.exe** returns the message PASS if the user has **sysadmin** role or the minimum required permissions. The detailed checking result is logged in koqVerifyPermissions_log.

The following lists the minimum permissions:

- Permissions for server must include **View server state**, **View any database** and **View any definition**.

  These server level permissions are mandatory.
- For all system databases and the user-defined databases for monitoring, the database role membership must include **public** and **db_owner**.

  The **db_owner** permission is required to collect data for the following data sets:
  - Server details data set
  - Database Details data set
  - Database Mirroring data set
  - Server Summary data set
  - Job Summary data set
- For **msdb** database, the database role membership must include **db_datareader**, **SQLAgentReaderRole** and **SQLAgentUserRole**. These permissions are required for Job Details data set.

Procedure

1. Launch the command prompt and change to the following utility directory.
   - For 64-bits agents, *Agent_home*\TMAITM6_x64
   - For 32-bits agents, *Agent_home* \TMAITM6

   where *Agent_home* is the agent installation directory.
2. Run the **koqVerifyPerminssions.exe** by providing the parameters:

   ```
   koqVerifyPermissions.exe -S Instance_name -U Username -P Password
   ```

   Where:
   - *Instance_name* is the SQL Server instance name.
   - *Username* is the user name that is verified by the utility tool.
   - *Password* is the password of the user. This parameter is required if *username* is provided.

   Note: If the *username* and the *password* are not provided, the default user that is logon to the system is used. Example: NT AUTHORITY\SYSTEM.

Results

The detailed checking result is available in koqVerifyPermissions_log at the following directory:

- For 64-bits agents, *Agent_home*\TMAITM6_x64\logs
- For 32-bits agents, *Agent_home* \TMAITM6\logs

Where *Agent_home* is the agent installation directory.

Creating a SQL Server user ID with Windows authentication

Windows Create a new user with the Windows authentication and assign the required roles and permissions to the user.

Procedure

To create a user, complete the following steps:

1. In the **SQL Server Management Studio**, open **Object Explorer**.
2. Click ***Server_instance_name*** > **Security** > **Logins**.
3. Right-click **Logins** and select **New Login**.
4. On the **General** page, in the **Login name** field, type the name of a Windows user.
5. Select **Windows authentication**.
6. Depending on the role and permissions that you want to assign to this user, complete one of the following tasks:

   - On the Server Roles page, assign the *sysadmin* role to the new login ID.
   - If you do not want to assign the *sysadmin* role to the user, grant minimum permissions to the user by completing the steps that are mentioned in "Granting minimum permissions for data collection" on page 543.

   Important: By default, the *public* role is assigned to the new login ID.
7. Click **OK**.

Results

A user is created with the default *public* role and the permissions that you assigned, and is displayed in the **Logins** list.

Creating a SQL Server user ID with SQL Server authentication

Linux Windows Create a new user with the SQL Server authentication and assign the required roles and permissions to the user.

Procedure

To create a user, complete the following steps:

1. In the **SQL Server Management Studio**, open **Object Explorer**.
2. Click ***Server_instance_name*** > **Security** > **Logins**.
3. Right-click **Logins** and select **New Login**.
4. On the **General** page, in the **Login name** field, type the name for a new user.
5. Select **SQL Server authentication**.
6. In the **Password** field, type a password for the user.
7. In the **Confirm Password** field, retype the password that you entered in the **Password** field.
8. Depending on the role and permissions that you want to assign to this user, complete one of the following tasks:

   - On the Server Roles page, assign the *sysadmin* role to the new login ID.
   - If you do not want to assign the *sysadmin* role to the user, grant minimum permissions to the user by completing the steps that are mentioned in "Granting minimum permissions for data collection" on page 543.

   Important: By default, the *public* role is assigned to the new login ID.
9. Click **OK**.

Results

A user is created with the default *public* role and the permissions that you assigned, and is displayed in the **Logins** list.

Granting minimum permissions for data collection

Windows Apart from the default **public** role, you can assign the **sysadmin** role to a user or grant the minimum permissions to a user so that the agent can collect data for data sets.

About this task

You can grant the permissions via user interface or the utility tool **`permissions.cmd`**.

Procedure

- To grant the minimum permissions to the user via the user interface, complete these steps:
  a) Open the **Server Roles** page and verify that the **public** check box is selected.
  b) Open the **User Mapping** page, and then select the following check boxes for all the system databases and the user-defined databases which you want to monitor:
     - **public**
     - **db_owner**

     For the **msdb** database, select the following additional check boxes:
     - **db_datareader**
     - **SQLAgentReaderRole**
     - **SQLAgentUserRole**
  c) Open the **Securables** page, and then select the following check boxes for the server instance that you are monitoring:
     - view database
     - view definition
     - view server state
- To grant the minimum permissions to the user by using the utility tool **`permissions.cmd`**, complete the following:
  a) Launch the Windows Explorer and browse to the utility tool directory *Agent_grant_perm_dir*:
     - For 64-bits agent, *Agent_grant_perm_dir* is *`Agent_home`*`\TMAITM6_x64\scripts\KOQ\GrantPermission`.
     - For 32-bits agent, *Agent_grant_perm_dir* is *`Agent_home`*`\TMAITM6\scripts\KOQ\GrantPermission`.
     - *Agent_home* is the agent installation directory.

     ⚠ **Attention:** The utility tool **`permissions.cmd`** grants **`db_owner`** on all databases by default. To exclude certain databases, you must add the database names in the *`Agent_grant_perm_dir`*`\exclude_database.txt` file. The database names must be separated by the symbol alias **`@`**.

     Tip: For example, you want to exclude the databases **`MyDatabase1`** and **`MyDatabase2`**, add the following entry in the `exclude_database.txt` file:

     ```
     MyDatabase1@MyDatabase2
     ```
  b) Double click **`permissions.cmd`** to launch the utility tool.
  c) Enter the intended parameter values when prompted:

Table 184. Parameters

Parameters	Description
SQL Server name or SQL Server instance name	Enter the target SQL Server name or the target SQL Server instance name that you want to grant permissions to the user.
The existing SQL Server user's logon name	Enter the user name whose permissions will be altered.
Permissions options: **1** Grant **`db_owner`** permission **2** Grant **`db_datareader`**, **`SQLAgentReaderRole`** and **`SQLAgentUserRole`** permissions **3** Grant all required permissions	Enter **1** or **2** or **3** according to your requirement.
The user to grant permissions: **1** The user who is currently logon to the system **2** Another user	Enter **1** or **2**. If **2** is select, enter the target user name when prompted. **Note:** The users must have access to grant permissions to other users.

What to do next
Configure the agent.

Granting permission to the Perflib registry key for collecting data for few data sets

Windows To collect data for few date sets, you need to grant users read access to the Perflib registry key.

About this task

You need to grant this permission to the Windows user with which agent services are configured. There are many data sets that are affected in absence of Perflib permissions like MS SQL Database Detail, MS SQL Memory Manager, MS SQL Lock Resource Type Summary, MS SQL Job Summary, MS SQL Server Transactions Summary, MS SQL Server Summary, and so on.

Procedure

To grant permission to the Perflib registry key, complete these steps:

1. To open Registry Editor, click **Start** > **Run** > **Regedit.exe**, and press **Enter**.
2. Go to the `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib` registry key.
3. Right-click the **Perflib** key, and click **Permissions**.
4. Click **Add**, enter the windows user name with which the agent is installed and configured, and then click **OK**.
5. Click the user that you added.
6. Allow read access to the user by selecting the check box.
7. Click **Apply**, and then click **OK**.

Local environment variables

You can change the behavior of the Microsoft SQL Server agent by configuring the local environment variables.

Variables for checking availability of the SQL Server service

To check the availability of the SQL Server service, use the following environment variables:

- **`COLL_MSSQL_RETRY_INTERVAL`**: This variable provides the retry interval (in minutes) to check the SQL Server service status. If the value is less than or equal to zero, then the variable takes the default value of 1 minute.
- **`COLL_MSSQL_RETRY_CNT`**: This variable provides the number of retries that the SQL Server agent makes to check whether the SQL Server service is started or not. If the SQL Server service is not started after the number of retries that are specified in this variable, then collector stops working. If the value of the variable is less than or equal to zero, then the variable takes the default value of 3.

Variables for monitoring the SQL Server error log file

To monitor the MS SQL Error Event Details data set, use the following environment variables:

- **`COLL_ERRORLOG_STARTUP_MAX_TIME`**: This variable provides the time interval (T) for error collection before the agent starts. The default value is `0` minutes. This variable can take the following values:

  T = 0
  The agent starts monitoring the error log file from the time the agent starts or is restarted. The agent does not read the errors that were logged in the error log file before the agent was started.

  T = 1
  The agent monitors the error log file according to the following values that are set for the **`COLL_ERRORLOG_STARTUP_MAX_EVENT_ROW`** variable, which is represented by R:

  - If R < 0, the agent starts monitoring the error log file from the time the agent starts or is restarted.
  - If R = 1, the agent monitors all the errors that are logged in the error log file.
  - If R > 1 and the agent is installed for the first time, the agent monitors the error log file until R errors are monitored. If R > 1 and the agent is restarted, the agent monitors all the previously missed R errors.

  T > 1
  The agent monitors all previous errors that were logged up to T minutes from the time that the agent starts or restarts. The agent monitoring also depends on the following values that you set for the **`COLL_ERRORLOG_STARTUP_MAX_EVENT_ROW`** variable:

  - If R ≤ 0, the agent starts monitoring the error log file from the time the agent is started or the agent is restarted.
  - If R = 1, the agent monitors the error log file for all the errors that are logged up to T minutes.
  - If R > 1, the agent does not monitor more than R errors that are logged in last T minutes.
- **`COLL_ERRORLOG_STARTUP_MAX_EVENT_ROW`**: This variable provides the maximum number of errors that must be processed when the agent starts. The default value is `0`. You can assign following values to this variable:

  R = 0
  The agent starts monitoring the error log file from the time that the agent starts or restarts. The agent does not read errors that were created in the error log file before the agent was started.

  R = 1
  The agent monitors the errors that were logged in the last T minutes from the time that the agent starts or restarts.

  R > 1
  The agent monitors R errors that are logged in the last T minutes.

- **COLL_ERRORLOG_MAX_EVENT_ROW**: This variable provides the number of error rows. The default value is 50. You can assign following values to this variable:

  X = 0
  The agent does not display the error logs.

  X > 0
  The agent displays the X error rows.
- **COLL_ERRORLOG_RECYCLE_WAIT**: This variable provides the time interval (in seconds) for which the Microsoft SQL Server agent waits before collecting data of the MS SQL Error Event Detail attribute group when the situation on this attribute group is triggered. You can assign a value to this variable in the range of 1 to 30. If the value of this variable is less than zero, then the variable takes the default value of zero (seconds). If the value of this variable is greater than 30, then the variable takes the default value of 30 (seconds).

Variable for setting the query timeout interval

To set the query timeout interval for the SQL Server agent, use the following environment variables:

- **QUERY_TIMEOUT**: This environment variable defines the maximum amount of time (in seconds) that the SQL Server agent waits to receive a response for a query that is sent to the SQL Server. The value for this variable must be less than 45 seconds. However, if you set the value for this variable as 0 seconds, the SQL Server agent waits indefinitely to receive a response from the SQL Server. If the SQL Server agent accesses many locked databases, you must assign the value to this variable in the range of 10 - 20 seconds. If the query is not processed within the set timeout interval, the SQL Server agent skips the timed out query and moves to the next query in the queue. The agent does not display data for the query that timed out.
- **QUERY_THREAD_TIMEOUT**: This environment variable defines the maximum amount of time (in seconds) that the SQL Server agent waits to receive a response for a query that is sent to the SQL Server. This environment variable is applicable for few attribute groups that uses threaded collection. For example, KOQDBD, KOQTBLD, KOQDEVD, and so on. The value for this variable does not have any limit unlike QUERY_TIMEOUT variable. Otherwise, it works similar to QUERY_TIMEOUT variable.

Variable for viewing information about the enabled jobs

To view the information about enabled jobs in the MS SQL Job Detail data set, use the **COLL_JOB_DISABLED** environment variable. If you set the value of this variable as 1, the Microsoft SQL Server agent does not display information about disabled jobs. If you do not specify this variable, you can view information that is about enabled and disabled jobs.

Variable for limiting the rows in the MS SQL Filegroup Detail data set

To limit the number of rows that the collector service fetches for the MS SQL Filegroup Detail data set, use the **COLL_KOQFGRPD_MAX_ROW** environment variable. This environment variable defines the maximum number of rows that the collector service fetches for the Filegroup Detail data set. If you do not specify a value for this variable, the collector service fetches 10,000 rows for the Filegroup Detail data set. Use this environment variable to modify the default limit of maximum rows in the koqcoll.ctl file. Complete the following steps to modify the default limit:

1. Specify the maximum number of rows for KOQFGRPD in the koqcoll.ctl file.
2. Add the **COLL_KOQFGRPD_MAX_ROW** environment variable, and ensure that the value of this variable is the same as the value that you have specified in the koqcoll.ctl file.

If the value in the koqcoll.ctl file is less than the value that is specified in the **COLL_KOQFGRPD_MAX_ROW** environment variable, the value in the koqcoll.ctl file is treated as the value for the maximum number of rows.

If the value in the koqcoll.ctl file is greater than the value that is specified in the **COLL_KOQFGRPD_MAX_ROW** environment variable, the value in the **COLL_KOQFGRPD_MAX_ROW** environment variable is treated as the value for the maximum number of rows.

Variables for enhancing the collection for the MS SQL Filegroup Detail data set

Use the **COLL_DBD_FRENAME_RETRY_CNT** variable to specify the number of attempts that can be made to move the %COLL_HOME%_tmp_%COLL_VERSION%_%COLL_SERVERID%_%COLL_SERVERID%__FGRP_TEMP file to the %COLL_HOME%_tmp_%COLL_VERSION%_%COLL_SERVERID%_%COLL_SERVERID%__FGRP_PREV file.

If you do not specify a value for this variable, the Microsoft SQL Server agent makes 3 attempts to move the file.

Variable for limiting the rows in the MS SQL Device Detail data set

To limit the number of rows that the collector service fetches for the MS SQL Device Detail data set, use the **COLL_KOQDEVD_MAX_ROW** environment variable. This environment variable defines the maximum number of rows that the collector service fetches for the Device Detail data set. If you do not specify a value for this variable, the collector service fetches 10,000 rows for the Device Detail data set. Use this environment variable to modify the default limit of maximum rows in the koqcoll.ctl file. Complete the following steps to modify the default limit:

1. Specify the maximum number of rows for KOQDEVD in the koqcoll.ctl file.
2. Add the **COLL_KOQDEVD_MAX_ROW** environment variable, and ensure that the value of this variable is the same as the value that you have specified in the koqcoll.ctl file.

If the value in the koqcoll.ctl file is less than the value that is specified in the **COLL_KOQDEVD_MAX_ROW** environment variable, the value in the koqcoll.ctl file is treated as the value for the maximum number of rows.

If the value in the koqcoll.ctl file is greater than the value that is specified in the **COLL_KOQDEVD_MAX_ROW** environment variable, the value in the **COLL_KOQDEVD_MAX_ROW** environment variable is treated as the value for the maximum number of rows.

Variables for enhancing the collection for the MS SQL Device Detail data set

To enhance the MS SQL Device Detail data set collection, use the following environment variables:

- **COLL_KOQDEVD_INTERVAL**: This environment variable enables you to specify a time interval (in minutes) between two consecutive collections of the MS SQL Device Detail data set.

  Note: By default, the data collection for the Device Detail data set is demand based. Use the **COLL_KOQDEVD_INTERVAL** variable to start a thread based collection for the Device Detail data set and to set the time interval between two threaded collections.
- **COLL_DBD_FRENAME_RETRY_CNT**: Use this environment variable to specify the number of attempts that can be made to move the %COLL_HOME%_tmp_%COLL_VERSION%_%COLL_SERVERID%_%COLL_SERVERID%__DEVD_TEMP file to the %COLL_HOME%_tmp_%COLL_VERSION%_%COLL_SERVERID%_%COLL_SERVERID%__DEVD_PREV file.

If you do not specify a value for this variable, the Microsoft SQL Server agent makes 1 attempt to move the file.

Variables for enhancing the collection for the MS SQL Database Detail data set

To enhance the MS SQL Database Detail data set collection, use the following environment variables:

- **COLL_KOQDBD_INTERVAL**: Use this environment variable to specify a time interval (in minutes) between two consecutive thread-based collections of the MS SQL Database Detail data set. If you do not specify a value for this variable or the specified time interval is less than 3 minutes, then the Microsoft SQL Server agent defaults to 3 minutes interval. In case, the collection is taking more time or the data is frequently seen as NOT_COLLECTED, then you can check the collection time by referring to the Database Detail Collection completed in %d seconds log and set the variable value to a value that is greater than the collection time specified in the log.
- **COLL_DBD_FRENAME_RETRY_CNT**: Use this environment variable to specify the number of attempts that can be made to move the %COLL_HOME%_tmp_%COLL_VERSION%_%COLL_SERVERID%_

%COLL_SERVERID%__DBD_TEMP file to the %COLL_HOME%_tmp_%COLL_VERSION%_ %COLL_SERVERID%_%COLL_SERVERID%__DBD_PREV file.

If you do not specify a value for this variable, the Microsoft SQL Server agent makes 1 attempt to move the file.

Variables for enhancing the collection for the MS SQL Audit Details data set

To enhance the MS SQL Audit Details data set collection, use the following environment variables:

- **COLL_AUDIT_TYPE**: Use this variable to enable or disable the monitoring of specific logs. The default value of the variable is [AL][FL][SL]. By default, the agent monitors all three types of logs that include the application logs, audit files, and the security logs. The value of the variable includes two character code for each log type:
  - [AL] for application logs
  - [FL] for audit files
  - [SL] for security logs

  You can change the value of the variable to disable the monitoring of specific log type. For example, if you specify the value of the variable as [AL][SL] the audit files are not monitored. If no value is specified for the variable, audit details not monitored.
- **COLL_AUDIT_DURATION**: Use this variable to report the audit events that occurred during the time interval that you specify in this variable. For example, if you set this variable to 7, the audit events that occurred only in last 7 hours are reported by the Audit Details data set. The default value of the **COLL_AUDIT_DURATION** variable is 24 hours.
- **COLL_AUDIT_COLLECTION_INTERVAL**: The threaded collection in the Audit Details data set provides specifications of all the database that are present on the SQL server instance. Use this variable to set the interval for this threaded collection. For example, if you set this variable to 7, a fresh set of database specifications is extracted from the SQL server instance after every 7 hours. The default value of the **COLL_AUDIT_COLLECTION_INTERVAL** variable is 24.

Variable for enhancing the collection for the MS SQL Process Detail data set

To enhance the MS SQL Process Detail data set collection, use the **COLL_PROC_BLOCK_INTERVAL** variable with the following values:

- If **COLL_PROC_BLOCK_INTERVAL =** 0, the collection for the Blocking Process Duration attribute, and the Blocking Resource Duration attribute is disabled.
- If **COLL_PROC_BLOCK_INTERVAL =** *x*, the interval between the two consecutive data collections for the Blocking Process Duration and the Blocking Resource Duration attributes is *x* minutes.

If the **COLL_PROC_BLOCK_INTERVAL** variable is not set in the CANDLE_HOME directory, the interval between the two consecutive data collections is three minutes.

Variable for excluding the locked objects from the data collection

If the queries that are sent for the Database Detail, Filegroup Details, Database Mirroring, and Device Detail workspaces take long to execute, use the **COLL_DBCC_NO_LOCK** variable to run a query with the value WITH (NOLOCK). This variable causes the query not to wait in the queue when an object on which the query is run is locked.

Variable for setting the sorting criteria for the rows returned by the Table Details data set

The rows that are returned by the Table Details data set are sorted in a descending order depending on the value that is set for the **COLL_TBLD_SORTBY** variable. The default value for the **COLL_TBLD_SORTBY** variable is FRAG (fragmentation percent). The valid values are: ROWS (number of rows in a tables), SPACE (space used by the table), and OPTSAGE (the optimizer statistics age of the table).

Variable for enhancing the collection for the MS SQL Problem Detail and Problem Summary data sets

- **COLL_ALERT_SEV**: Use this variable to set the severity level of the error messages that are displayed in the Problem Detail and Problem Summary data sets. Error messages, which have a severity level that is equal to or greater than the value mentioned in this variable, are displayed in the Problem Detail and Problem Summary data sets. For example, if you set the value of this variable to 10, the error messages with severity level 10 or greater are displayed in the Problem Detail and Problem Summary data sets. If you do not specify a value for this variable, the error messages, which have a severity level that is equal to or greater than 17, are displayed in the Problem Detail and Problem Summary data sets.
- **COLL_SINCE_ERRORLOG_RECY**: Use this variable to monitor only the high severity errors in the current ERRORLOG file. If you do not specify a value for this variable, the value of the variable is 0, which means that for collecting the data, the Problem Summary data set also considers the high severity errors that are read from the previous ERRORLOG file. To monitor only the high severity errors in the current ERRORLOG file, set the value of this variable to 1.

Variables for setting the timeout interval

To set the timeout interval for the Microsoft SQL Server agent, you can use the following environment variables:

- **WAIT_TIMEOUT**: Use this variable to set the wait timeout interval for the Microsoft SQL Server agent. If any data set takes more than 45 seconds to collect data, then the agent might hang or situations might be incorrectly triggered. Check the log for the data sets that take more than 45 seconds to collect the data, and use the **WAIT_TIMEOUT** variable to increase the wait time between the agent process and the collector process.
- **COLL_DB_TIMEOUT**: Use this variable to define the wait interval (in seconds) for any request such as running a query on the existing SQL server connection to complete before returning to the application. If you set this value to 0, then there is no timeout. If you do not specify a value for this variable, the agent waits 15 seconds before returning to the application.

Variables for setting the properties of the collector log files

To set the properties of the collector log files, you can use the following environment variables:

- **COLL_WRAPLINES**: Use this variable to specify the maximum number of lines in a col.out file. The default value of this variable is 90,000 lines (about 2 MB).
- **COLL_NUMOUTBAK**: Use this variable to specify the number of backup copies of the collector log files that you want to create. By default, five backup copies of the collector log file are created. The backup file is named *.out. When this backup file is full, the file is renamed to *.ou1 and the latest logs are written in the *.out file. In this manner, for five backup files, the oldest logs are available in the *.ou5 file and the latest logs are available in the *.out file.

  You can create more than five backup copies of the collector log files by specifying one of the following values in the **COLL_NUMOUTBAK** variable:

  - For less than 10 backup files, specify the number of backup files that you want to create in the **COLL_NUMOUTBAK** variable. For example, if you specify 9 in the **COLL_NUMOUTBAK** variable, nine backup files will be created.
  - For more than 9 and less than 1000 backup files, in the **COLL_NUMOUTBAK** variable, specify the number of backup files preceded by a hyphen. For example, if you specify -352 in the **COLL_NUMOUTBAK** variable, three hundred and fifty-two backup files will be created.
- **COLL_DEBUG**: Use this variable to enable full tracing of the collector by setting the value of this variable to dddddddddd (10 times"d").

Variable for deleting the temporary files

COLL_TMPFILE_DEL_INTERVAL: Use this variable to specify the interval (in minutes) after which the KOQ_<timestamp> temporary files should be deleted. If you do not specify a value for this variable, the value of the variable is 0, which means that the temporary files must be deleted immediately.

Variable for changing driver used by the MS SQL Server agent

To change the driver that is used by the Microsoft SQL Server agent, use the **KOQ_ODBC_DRIVER** environment variable. This variable specifies the driver that the Microsoft SQL Server agent uses to connect to the SQL Server. If you do not specify a value for this variable, then agent uses the ODBC SQL Server Driver as a default driver.

Note: When you specify the Microsoft SQL Server driver, ensure that the driver name is correct and the driver is listed under the drivers' option in data source (ODBC).

Variable for connecting to an AlwaysOn enabled SQL Server database

KOQ_APPLICATION_INTENT: Use this variable to specify the connection option while connecting to SQL Server.
KOQ_APPLICATION_INTENT option details:

- **Readonly**: Connection is opened with **ApplicationIntent** as *readonly*.
- **Readwrite**: Connection is opened with **ApplicationIntent** as *readwrite*.
  When it is set to Readwrite, Microsoft SQL Server agent would not perform any write operations with the connection.

If this variable is not set, the connection is established without **ApplicationIntent** property.

Note: The driver is specified by the environment variable **KOQ_ODBC_DRIVER**. If this variable is not set, then the default SQL Server driver is used.
If the driver doesn't support **ApplicationIntent**, the connection is opened without **ApplicationIntent** property.

Configuration parameters of agent

You must provide the mandatory configuration parameters of the agent.

About this task

The following table contains the details of configuration parameters. Review the parameters and determine the value for each parameter.

Parameter Name	Description	Default Value	Mandatory Field
User Name	User name or login used to establish connection between agent and SQL Server	NA	Yes
Password	Password of the user or login	NA	Yes
Database Version	Version of SQL Server Database to be monitored	NA	Yes
Database Server Home Directory	Home Path of SQL Server Database	NA	Yes
Error Log File Path	Location where SQL Server Error Log file is present	NA	Yes

Configuring the agent on Windows systems

You can use the IBM® Cloud Application Performance Management window to configure the agent on Windows systems.

Before you begin

Before you configure the agent, ensure that you complete the following tasks:

- Create a user and grant the required permissions
- Review the local environment variables

About this task

The Microsoft SQL Server agent is a multiple instance agent; you must configure and start each agent instance manually.

- To configure the agent, complete the following tasks:
  - Select the databases for monitoring
  - Configure the local environment variables

Selecting the databases for monitoring
You can select the database that you want to monitor by using the **Configure Database Agents** window.

Procedure

1. Open the **IBM Performance Management** window.
2. In the **IBM Performance Management** window, click the **Task/SubSystem** column, right-click **Template**, and then **Configure Using Defaults**.
3. In the **Configure Database Agents** window, select the database server that you want to monitor from the **Database Servers Available**, and move it to the **Server to Monitor** list.
4. In the **Database Server Properties** window, values for the following fields are automatically populated:

   - Server Name
   - Database Version
   - Home Directory
   - Error Log File

   The following fields in the **Database Server Properties** window are optional:

   - Windows Authentication
   - Support Long Lived Database Connections
   - Extended Parms
   - Monitor all Databases
   - Day(s) Frequency
   - Weekly Frequency
   - Monthly Frequency
   - Collection Start Time
   - Table Detail Continuous Collection

   For more information about the configuration parameters in the **Database Server Properties** window, see "Configuration parameters for the Database Server properties" on page 552.
5. If you do not select the **Windows Authentication** field, enter your user ID and password in the **Login** and **Password** fields by using only ASCII characters.

6. In the **Extended Parms** field, enter the name of the data set to disable the data collection, and then click **OK**.

   For example:

   - Enter koqtbld to disable data collection for Table Detail data set.
   - Enter koqdbd to disable data collection for Database Detail data set.
   - Enter koqtbld,koqdbd to disable data collection for Table Detail and Database Detail data sets.
7. If you do not select the **Monitor All Databases** check box, specify the list of databases for which you want to enable or disable monitoring in the field of **Databases** group area.

   Remember: If you select the **Monitor All Databases** check box and specify the databases in **Databases** group area, the setting of **Monitor All Databases** check box takes precedence.
8. Specify the frequency for the collection of the MS SQL Table Detail data set. The possible values are daily, weekly, or monthly.
9. Select the **Table Detail Continuous Collection** check box to enable continuous collection of the MS SQL Table Detail data set. If you select the **Table Detail Continuous Collection** check box, enter a value in the **Interval Between Two Continuous Collection (in minutes)** field.
10. In the **Configure Database Agents** window, click **OK**, and then start the agent.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you need help with troubleshooting, see the Troubleshooting section.

Configuration parameters for the Database Server properties

In the **Database Server Properties** window, you can configure the Database Server properties, such as server name, database version, and home directory.

The following table contains detailed descriptions of the configuration settings in the **Database Server Properties** window.

*Table 185. Names and descriptions of configuration settings in the **Database Server Properties** window*

Parameter name	Description	Mandatory field	Examples
Server Name	The name of the Microsoft SQL Server instance that is to be monitored. Use MSSQLSERVER as the instance name for the default instance. The name must be short enough to fit within the total managed system name, which must be 2 - 32 characters in length.	Yes	If the Microsoft SQL Server instance that is monitored is the default Microsoft SQL Server instance, enter MSSQLSERVER in this field. If the Microsoft SQL Server instance that is monitored is a named instance where the instance name is mysqlserver and the host name is popcorn, enter mysqlserver in this field.

*Table 185. Names and descriptions of configuration settings in the **Database Server Properties** window (continued)*

Parameter name	**Description**	**Mandatory field**	**Examples**
Login	The Microsoft SQL Server user ID to be used to connect to the Microsoft SQL Server. The user ID is required only when **`Windows Authentication`** parameter is set to `False`. Use only ASCII characters for the User ID. When you configure the Microsoft SQL Server agent by specifying a login ID in the **Login** field, the agent uses this login ID to connect to the Microsoft SQL Server. **Important:** While configuring the agent if you select the **Windows Authentication** check box and specify a login ID in the **Login** field, the agent gives preference to the Windows Authentication.	No	
Password	The password for the Microsoft SQL Server user ID. Password is required only when **`Windows Authentication`** parameter is set to `False`. Use only ASCII characters for the password.	No	
Database Version	The version of SQL server instance.	Yes	The database versions for SQL server instance are as follows: • Microsoft SQL Server 2014 - 12.0.2000.8 • Microsoft SQL Server 2012 - 11.0.2100.60 • Microsoft SQL Server 2008 R2 - 10.50.1600.1 • Microsoft SQL Server 2008 - 10.0.1600.22 • Microsoft SQL Server 2005 - 9.0.1399.06

*Table 185. Names and descriptions of configuration settings in the **Database Server Properties** window (continued)*

Parameter name	**Description**	**Mandatory field**	**Examples**
Home Directory	The SQL server installation directory.	Yes	The default home directory path for the default Microsoft SQL Server 2005 instance is `C:\Program Files\Microsoft SQL Server\MSSQL`. A named Microsoft SQL Server 2005 instance has a default home directory path in the format `C:\Program Files\Microsoft SQL Server\MSSQL$`*`instance_name`*, where *instance_name* is the Microsoft SQL Server instance name.
Error Log File	The fully qualified location and name of the SQL Server error log.	Yes	The default error log path for the default Microsoft SQL Server 2005 instance is `C:\Program Files\Microsoft SQL Server\MSSQL\LOG\ERRORLOG`. A named Microsoft SQL Server 2005 instance has a default error log path in the format `C:\Program Files\Microsoft SQL Server\MSSQL$`*`instance_name`*`\LOG\ERRORLOG`, where *instance_name* is the Microsoft SQL Server instance name.

Table 185. Names and descriptions of configuration settings in the ***Database Server Properties*** *window (continued)*

Parameter name	Description	Mandatory field	Examples
Windows Authentication	Windows Authentication is a Windows account with which the agent services are configured, and is the default configuration option. If you select the **Windows Authentication** check box, Windows credentials are used for authentication. When the Microsoft SQL Server agent is configured with Windows Authentication, either **Local System account** or **This account** is used by the agent services to log on to the Microsoft SQL Server. • If the agent services are configured to use **Local System account** to log on, then the agent uses the NT AUTHORITY\SYSTEM user ID to access the Microsoft SQL Server. • If the agent services are configured to use **This account** to log on, then the agent uses the respective user ID to access the Microsoft SQL Server. **Remember:** If you do not select the **Windows Authentication** check box, you must specify values for the **Login** and **Password** parameters. If you do not specify these parameters and click **OK** in the **Database Server Properties** window, an error message is displayed in a pop-up window and the agent configuration does not finish. **Important:** If you configure the agent by selecting the **Windows Authentication** check box and specifying a login ID in the **Login** field, the agent gives preference to the Windows Authentication.	No	
Support Long Lived Database Connections	Enables or disables long lived database connections. The following data sets do not use long-lived database connections: • MS SQL Text • MS SQL Filegroup Detail • MS SQL Server Summary	No	

Table 185. Names and descriptions of configuration settings in the ***Database Server Properties*** *window (continued)*

Parameter name	Description	Mandatory field	Examples
Extended Parms	Disables data collection of any attribute group.	No	For example: To disable the data collection for Table Details data set, enter `koqtbld` in the **Extended Parms** field. To disable the data collection for Database Details data set, enter koqdbd in the **Extended Parms** field. To disable the data collection for Table Details and Database Details data sets, enter `koqtbld,koqdbd` in the **Extended Parms** field.

*Table 185. Names and descriptions of configuration settings in the **Database Server Properties** window (continued)*

Parameter name	Description	Mandatory field	Examples

Database	To select the databases for monitoring, specify a value for this parameter. To enable monitoring of all the databases that are available on the SQL server instance, select the **Monitor All Databases** check box in the **Databases** group area. The **Monitor All Databases** check box is selected by default.
	To enable or disable the monitoring of particular databases, clear the **Monitor All Databases** check box.
	• To monitor particular databases, select **Include** from the list, and then specify the names of the databases in the text field next to the list.
	• To exclude particular databases from being monitored, select **Exclude** from the list, and then specify the names of the databases in the text field next to the list.
	Use the text field to filter databases that you want to monitor.
	To specify database filter, you must first select a separator. A separator is a character that distinguishes a database name or database expression from the other database name or database expression.
	When you are selecting a separator, ensure that database names and database expression do not contain the character that you choose as a separator. You must not use the wildcard characters that are typically used in the T-SQL query (for example, %, _, [ ], ^, -) if they are used in the database names or database expression.
	When you are specifying database filter:
	• Database names must start with a separator.
	• Database expression must start with 2 separators.
	Database expression is a valid expression that can be used in the `LIKE` part of the `T-SQL` query. However, you cannot use the `T-SQL ESCAPE` clause when you are specifying the database expression. The following data sets are affected by database filter: Database Detail, Database Summary, Device Detail, Table Detail, Table Summary, Filegroup Detail, Additional Database Detail
		No	Examples of filters:
			Case 1: `% usage`
			Example:
			`@@%m%`
			Output: All the databases that have the character m in their names are filtered.
			Case 2: `_ usage`
			Example:
			`@@____`
			Output: All the databases that are of length four characters are filtered.
			Case 3: `[] usage`
			Example:
			`@@[m]___`
			Output: All the databases of length four characters and whose names start with the character m are filtered.
			Case 4: `[^] usage`
			Example:
			`@@[^m]%`
			Output: All the databases (of any length) except those whose names start with the character *m* are filtered.

Table 185. Names and descriptions of configuration settings in the ***Database Server Properties*** *window (continued)*

Parameter name	Description	Mandatory field	Examples

Database (continued)	**Remember:**
	• If you do not select the **Monitor All Databases** check box, you must specify the list of databases for which you want to enable or disable monitoring, in the text field that is present in the **Databases** group area. If you click **OK** in the **Database Server Properties** window without selecting the **Monitor All Databases** check box and specifying the list of databases, an error message is displayed in a pop-up window and the agent configuration does not finish.
	• If you select the **Monitor All Databases** check box and also specify the databases to monitor in the text field that is present in the **Databases** group area, then priority is given to the value of the **Monitor All Databases** check box. The list of databases that you specify in the text field is ignored.
			Case 5: Wrong input
			Example:
			`@%m%`
			Output: None of the databases are filtered.
			Case 6: Default
			Example: Field is blank (No query is typed)
			Output: All the databases are filtered.
			Case 7: Mixed patterns
			Example:
			`@@[m-t]_d%`
			Output: All the databases (of any length) whose names start with the characters m, n, o, p, q, r, s, t, followed by any character, with the character d in the third place are filtered.

Day(s) Frequency	Use this feature to define the frequency of collecting data of Table Detail attributes. The values can be from zero to 31.	No	
Weekly Frequency	Use this feature to specify a particular day for collecting data for Table Detail attributes. The values can be from zero to 7.	No	
Monthly Frequency	Use this feature to define the data collection of Table Detail attributes on a particular day of the month. The possible values are 1, 2, 3, and so on.	No	
Collection Start Time	The collection start time can be entered in HH:MM format. The possible values for hours are zero to 23. The default value is zero. The possible values for minutes are from zero to 59. The default value is zero.	No	

*Table 185. Names and descriptions of configuration settings in the **Database Server Properties** window (continued)*

Parameter name	Description	Mandatory field	Examples
Table Detail Continuous Collection	Use this feature for the continuous background collection of Table Detail data. The **Table Detail Continuous Collection** check box is selected by default.	No	
Interval Between Two Continuous Collection (in min.)	Specify the time for the interval between two collections in minutes. The minimum interval time is 3 minutes. You can select the **Interval Between Two Continuous Collection (in min.)** check box or you can use Scheduling to specify continuous collection of the Table Detail data set. If you select the **Interval Between Two Continuous Collection (in min.)** check box, you must specify the time interval for collection. If you use Scheduling to specify the collection of the Table Detail data set, the minimum time interval is 1 day. The default interval between two continuous collections is 3 minutes.	No	

The agent collects the data at the time interval for which data collection occurs frequently. For example, if you specify all frequencies (daily, weekly, and monthly) for collecting data, the agent starts the data collection according to the following conditions:

- If day(s) frequency ≤ 7, the day(s) frequency settings are selected, and the weekly and monthly frequency settings are ignored.
- If day(s) frequency > 7, the weekly frequency settings are selected, and the day(s) and monthly frequency settings are ignored.

Remember: If the **Table Detail Continuous Collection** check box is selected, the agent collects the data at the interval that is mentioned in the **Interval Between Two Continuous Collection (in min.)** field and not according to the daily, weekly, or monthly frequencies.

Configuring local environment variables on Windows systems

You can configure local environment variables to change the behavior of the Microsoft SQL Server agent.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, from the **Actions** menu, click **Advanced > Edit Variables**.
3. In the Monitoring Agent for Microsoft SQL Server: **Override Local Variable Settings** window, click **Add**.
4. In the **Add Environment Setting Override** window, enter the variable and the corresponding value.

   Note: Refer to "Local environment variables" on page 545 for the complete list of configurable environment variables.

Running as a non-administrator user

You can run the monitoring agent for Microsoft SQL Server as a non-administrator user.

About this task

The Microsoft SQL Server agent can be run as a non-administrator user from Domain Users group.

Procedure

1. Start Windows application Active Directory Users and Computers and create a domain user.
   - Make sure that the new user is a member of the *Domain Users* group.
   - Make sure that the SQL Server is a member of *Domain Computers*.
2. Add the newly created domain user in the *SQL Server Login* user group. The domain user should have **`sysadmin`** SQL Server role permission on the SQL Server or the permissions that are mentioned in https://www.ibm.com/support/knowledgecenter/SSMKFH/com.ibm.apmaas.doc/install/sql_config_agent_grant_permission_sqlserver.htm.
3. Log on to the SQL Server as the domain administrator.
4. Grant **Modify** permission to every drive that the Microsoft SQL Server agent accesses. Complete the following procedures to propagate the permission to all sub directories:
   a) Go to **My Computer**.
   b) Right-click the **drive**.
   c) Click the **Security** tab.
   d) Add the newly created user.
   e) Give **Modify** permission to the newly created user.
   f) Click **OK**. This procedure takes a few minutes to apply permission to all sub directories.
5. By using the Windows Registry, grant read access to `HKEY_LOCAL_MACHINE`, and propagate the settings. Complete the following steps to propagate the settings:
   a) Right-click the `HKEY_LOCAL_MACHINE` directory and select **Permissions**.
   b) Add the newly created user.
   c) Select the newly created user.
   d) Select the **Allow Read** check box.
   e) Click **OK**. This procedure takes a few minutes to propagate the settings to the entire `HKEY_LOCAL_MACHINE` tree.
6. By using the Windows Registry, grant the agent-specific registry permissions according to the following list.
   - If you installed a 32-bit agent on a 32-bit operating system, grant full access to the `KEY_LOCAL_MACHINE\SOFTWARE\IBMMonitoring` directory, and then propagate the settings.
   - If you installed a 32-bit agent on a 64-bit operating system, grant full access to the `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Candle` directory, and then propagate the settings.
   - If you installed a 64-bit agent on a 64-bit operating system, grant full access to the `KEY_LOCAL_MACHINE\SOFTWARE\IBMMonitoring` directory, and then propagate the settings.

   Complete the following steps to propagate settings:
   a) Right-click the directory for which you have full access and select **Permissions**.
   b) Add the newly created user.
   c) Select the newly created user.
   d) Select the **Allow Full Control** check box.
   e) Click **OK**. This procedure takes a few minutes to propagate the settings to the entire `KEY_LOCAL_MACHINE\SOFTWARE\IBMMonitoring` tree.

7. Add a new Domain User to the **Performance Monitor Users** group.
8. Verify that Domain Users are members of the *Users* group.
9. Grant the following permissions to the Windows directory to run as a non-administrator user:

   - If a 32-bit agent is installed on a 32-bit operating system, grant read and write access to the `OS_installation_drive:\Windows\system32` directory
   - If a 32-bit agent is installed on a 64-bit operating system, grant read and write access to the `OS_installation_drive:\Windows\SysWOW64` directory

   Note: Permissions for Windows directory are not necessary for Windows Server 2008, Windows Server 2008 R2, and Windows Server 2012, Windows Server 2012 R2, Windows Server 2016.
10. Grant **Modify** permission to the SQL Server data file and log file:

    - The default path of the SQL Server data file is *`SQLServer_root_dir`*`\DATA`, where *SQLServer_root_dir* is the root directory of the SQL Server instance. For example, if the root directory of the SQL Server instance is `C:\Program Files\Microsoft SQL Server \MSSQL.1\MSSQL`, the data file path is `C:\Program Files\Microsoft SQL Server \MSSQL.1\MSSQL\DATA`.
    - The default path of the SQL Server log file is *`SQLServer_root_dir`*`\LOG`, where *SQLServer_root_dir* is the root directory of the SQL Server instance. For example, if the root directory of the SQL Server instance is `C:\Program Files\Microsoft SQL Server \MSSQL.1\MSSQL`, the log file path is `C:\Program Files\Microsoft SQL Server \MSSQL.1\MSSQL\LOG`.
11. Grant full permissions to the `Candle_Home` directory. The default path is `C:\IBM\ITM`.
12. Apply local security permissions, refer "Local Security Policy permissions" on page 561.
13. Restart the SQL Server to ensure that local security permissions are applied effectively.
14. Change the logon settings for the SQL Server agent services to the non-administrator user by completing the following steps:
    a) Click **Start > Administrative Tools > Services**.
    b) Right-click the **Monitoring Agent For SQL Server *instance_name***, and click **Properties**. The **SQL Service Properties** window opens.
    c) Click **Log On** tab.
    d) Click **This account** and type the user name.
    e) In the **Password** and **Confirm Password** fields, enter the password, and click **OK**.
    f) Repeat steps b to e for the **Monitoring Agent For SQL Server Collector *instance_name***, where *instance_name* is the Microsoft SQL Server instance name.

Local Security Policy permissions

Local security policy administers the system and its security policy. It plays an important part in keeping the agent and the system in which the agent is installed secure. This policy works by giving access rights, permissions to users. For, Microsoft SQL Server agent, make sure that the user has following permissions to adhere to local security permission policy.

Log on as Service permission

About this task

To grant the Log-on as service permission, complete the following steps.

Procedure

1. Click **Start > Administrative Tools > Local Security Policy**. The **Local Security Settings** window opens
2. Click **Local Policies** to expand the list.

3. Click **User Rights Assignment**. The list of user rights opens.
4. Double-click **Log-on as service** policy. The **Log-on as service Properties** window opens.
5. Click **Add User or Group**. The **Select Users or Groups** window opens.
6. In the **Enter the object names to select** field, enter the user account name to whom you want to assign the permissions, and click **OK**.
7. Click **OK**.

Debug Programs Permission

About this task

To grant the debug program permission, complete the following procedure on Microsoft SQL Server agent .

Procedure

1. Click **Start > Administrative Tools > Local Security Policy**. The **Local Security Settings** window opens.
2. Click **Local Policies** to expand the list.
3. Click **User Rights Assignment**.The list of user rights opens.
4. Double-click **Debug Programs** policy. The **Debug programs Properties** window opens.
5. Click **Add User or Group**. The **Select Users or Groups** window opens.
6. In the **Enter the object names to select** field, enter the user account name to whom you want to assign permissions, and click **OK**.
7. Click **OK**.

Impersonate a client after authentication

About this task

To grant the Impersonate a client after authentication permission, complete the following procedure on Microsoft SQL Server agent .

Procedure

1. Click **Start > Administrative Tools > Local Security Policy**. The **Local Security Settings** window opens.
2. Click **Local Policies** to expand the list.
3. Click **User Rights Assignment**.The list of user rights opens.
4. Double-click **Impersonate a client after authentication** policy. The **Impersonate a client after authentication Properties** window opens.
5. Click **Add User or Group**. The **Select Users or Groups** window opens.
6. In the **Enter the object names to select** field, enter the user account name to whom you want to assign permissions, and then click **OK**.
7. Click **OK**.

Configuring the agent on Linux systems

To configure the agent on Linux operating systems, you must run the script and respond to prompts.

Before you begin

Before you configure the agent, ensure that you complete the following tasks:

- Review the local environment variables

About this task

The Microsoft SQL Server agent is a multiple instance agent; you must configure and start each agent instance manually.

Procedure

1. On the command line, change the path to the agent installation directory.

   Example:

   ```
   cd /opt/ibm/apm/agent/bin
   ```
2. Run the following command where *instance_name* is the name that you want to give to the instance:

   ```
   ./mssql-agent.sh config instance_name
   ```
3. When the command prompt displays the following message, type 1 and enter:

   `Edit ‘Monitoring Agent for MSSQL setting? [1=Yes, 2=No]`
4. Specify values for the configuration parameters when you are prompted.

   For information about the configuration parameters, see Configuration parameters of agent.
5. Run the following command to start the agent:

   ```
   ./mssql-agent.sh start instance_name
   ```
6. Run the following command to stop the agent:

   ```
   ./mssql-agent.sh stop instance_name
   ```

Configuring local environment variables on Linux systems

You can configure local environment variables to change the behavior of the Microsoft SQL Server agent on Linux systems.

Procedure

1. Launch a terminal or the system file manager, change directory to agent installation directory:

   Example:

   ```
   /opt/ibm/apm/agent
   ```
2. Run the following command to stop the agent:

   ```
   ./mssql-agent.sh stop instance_name
   ```

   Where *instance_name* is the agent instance name.
3. Open the file `.oq.environment` that exists in the following config directory:

   Example:

   ```
   install_dir/config
   ```

   Where *install_dir* is the agent installation directory.
4. Add the required environment variables at end of the file `.oq.environment` by following the name-value pair format.

   `export VARIABLE_NAME=VARIABLE_VALUE`

   Example:

   ```
   export KOQ_ODBC_DRIVER=ODBC Driver 17 for SQL Server
   ```

   Note:

- Refer to “Local environment variables” on page 545 for the complete list of configurable environment variables.
- The custom variables added are not preserved after agent upgrade.

5. Save the file.
6. Start the agent from the agent installation directory:

   ```
   cd /opt/ibm/apm/agent/bin
   ```

   ```
   ./mssql-agent.sh start instance_name
   ```

Configuring the agent by using the silent response file

You can use the silent response file to configure the agent or multiple instances of the agent.

Before you begin

To configure multiple instances of the agent, ensure that the configuration details of all the agent instances are specified in the silent response file.

About this task

Run the configuration script to change the configuration settings. You can edit the silent response file before you run the configuration script.

Procedure

To configure the agent, complete the following steps:

1. Launch a text editor, open the silent response file that is available at the following location:

   - Windows *install_dir*\samples/mssql_silent_config.txt
   - Linux *install_dir*/samples/mssql_silent_config.txt

   Where *install_dir* is the agent installation directory.

   Example:

   - Windows C:\IBM\APM\samples\mssql_silent_config.txt
   - Linux /opt/ibm/apm/agent/samples/mssql_silent_config.txt

   Note: For information about the agent configuration parameters, see “Configuration parameters of agent ” on page 550.
2. Launch a command prompt, change the directory to the following:

   - Windows

     ```
     cd install_dir\bin
     ```

   - Linux

     ```
     cd install_dir/bin
     ```

3. Run the following command:

   - Windows

     ```
     mssql-agent.bat config install_dir\samples\mssql_silent_config.txt
     ```

   - Linux

     ```
     mssql-agent.sh config  instance_name install_dir/samples/mssql_silent_config.txt
     ```

4. Start the agent.

- Windows In the **IBM Performance Management** window, right-click the agent instance that you created, and click Start.
- Linux Run the following command:

  ```
  cd /opt/ibm/apm/agent/bin
  ```

  ```
  ./mssql-agent.sh start instance_name
  ```

What to do next
Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you need help with troubleshooting, see the Troubleshooting section.

Running the agent in a cluster environment

Windows You can configure the Microsoft SQL Server agent in a cluster environment. Multiple instances of the Microsoft SQL Server and the Microsoft SQL Server agent can run on a single node.

After you install and configure the Microsoft SQL Server agent, complete the following tasks to run the agent in a cluster environment:

- Add environment variables
- Change the startup type of the agent service and the collector service
- Add the agent and the collector to the cluster environment

You can set up a cluster environment for the following versions of the Microsoft SQL Server:

- Microsoft SQL Server 2005
- Microsoft SQL Server 2008
- Microsoft SQL Server 2008 R2
- Microsoft SQL Server 2012
- Microsoft SQL Server 2014
- Microsoft SQL Server 2016

Important: On Windows systems, the agent must be installed in the same directory where the OS agent is installed. Install the agent on the nodes system disk of each cluster node.

Adding environment variables

You must configure the environment variables that are used by the agents that are installed on each cluster node.

About this task
You must specify values for the following environment variables:

- *CTIRA_HOSTNAME*: This variable is used to configure each instance of the Microsoft SQL Server agent. The value of this variable is limited to 31 characters and is common for all monitoring agents. Set the value of this variable to the cluster name to navigate to all the monitoring agents for that cluster in the Application Performance Dashboard.
- *CTIRA_NODETYPE*: This variable is used to identify the agent. By default, the value of this variable is set to **`MSS`** for the Microsoft SQL Server agent.
- *CTIRA_SUBSYSTEMID*: This variable is used to distinguish the multiple instances of the Microsoft SQL Server agent. By default, the value of this variable is set to **`Microsoft SQL Virtual Server`** for the Microsoft SQL Server agent.

- *COLL_HOME*: This variable is used to collect data and store log files for attribute groups that use configuration files at a shared location. Set the value of the variable to `X:\shared-location`, where *X* is a shared drive that is accessible to the cluster nodes. For example, set the value for the *COLL_HOME* variable when you define the configuration settings for the MS SQL Table Detail attribute group or MS SQL Error Event Details attribute group.
- *CTIRA_HIST_DIR*: This variable is used to specify the path to the shared disk directory. If history for the Microsoft SQL Server agent is configured to be stored at the monitoring agent, each instance of the agent must be configured with a common *CTIRA_HIST_DIR* variable that refers to the shared disk directory.

  Remember: If history is stored at the Cloud APM server, you need not specify a value for the *CTIRA_HIST_DIR* variable. Storing history at the Cloud APM server increases the load on that server.

To add these variables, see the steps that are described in "Configuring local environment variables on Windows systems" on page 559.

What to do next

Change the startup type of the agent service and the collector service to **Manual** by completing the steps that are described in "Changing the startup type of the agent service and the collector service" on page 566.

Changing the startup type of the agent service and the collector service

By default the startup type of the agent service and the collector service is **Automatic**. Change the startup type of the agent service and the collector service to **Manual** so that the cluster resource can control the starting and stopping of the monitoring agent

Procedure

To change the startup type of the agent service, complete the following steps:

1. Click **Start** > **Run**, type the command `services.msc`, and then click **OK**.
2. Right-click the agent and click **Properties**.
3. In the **Monitoring Agent for Microsoft SQL Server Properties** window, from the **Startup type** list, select **Manual**, click **Apply**, and then **OK**.

What to do next

- Use the same procedure to change the startup type of the collector service to **Manual**.
- Add the agent and the collector to the cluster environment by completing the steps that are described in "Adding the agent and collector to the cluster environment " on page 566.

Adding the agent and collector to the cluster environment

You must add the agent and the collector to the cluster environment.

Procedure

1. Click **Start > Control Panel > Administrative Tools > Failover Cluster Management**.
2. Expand **Failover Cluster Management**.
3. Expand **Services and Applications** and right-click the SQL instance that you want to configure.
4. Click **Add a resource > Generic Service**. The New Resource Wizard opens.
5. On the Select Service page, select the service name, and then click **Next**.

   Examples of Windows Services names:

   - `Monitoring Agent for Microsoft SQL Server: SQLTEST#INSTANCE1`
   - `Monitoring Agent for Microsoft SQL Server: Collector SQLTEST#INSTANCE1`
   - `Monitoring Agent for Microsoft SQL Server: SQLTEST2#INSTANCE2`
   - `Monitoring Agent for Microsoft SQL Server: Collector SQLTEST2#INSTANCE2`

6. On the Confirmation page, check the details, and then click **Next**.
7. On the Summary page, click **Finish**. The Microsoft SQL Server agent is now added.

   Remember: Use the same steps to add the collector to the cluster environment.
8. To bring the agent online, right-click the agent, and click **Bring this resource online**.
9. To bring the collector online, right-click the collector, and click **Bring this resource online**.

Results

The Microsoft SQL Server agent is now running in a cluster environment.

Remember: If you want to configure the agent again, you must first take the agent and the collector offline, or edit the agent variables on the node where the agent and collector run. When you complete the agent configuration, bring the agent and the collector back online.

Configuring the agent by using the cluster utility

Windows You can use the cluster utility to add multiple Microsoft SQL Server agent instances to a cluster group in a cluster environment.

The cluster utility automatically adds the agent service and the collector service of each Microsoft SQL Server agent instance as a generic service resource to the cluster group. You can use the cluster utility to complete the following tasks:

- Adding an SQL Server agent instance to the cluster
- Updating an existing SQL Server agent instance in a cluster
- Removing an SQL Server agent instance from a cluster

Prerequisites for using the cluster utility

You must ensure that your system environment meets the prerequisites for running the cluster utility.

Ensure that the following prerequisites are met:

- Run the cluster utility on a computer that has at least one group in the cluster environment.
- Start the remote registry service for all nodes in the cluster.
- You must have the cluster manager authorization to access the cluster utility.
- The service name of agent and collector must be same on all cluster node.

  For example, if the agent service name is Monitoring Agent for Microsoft SQL Server: `SQLTEST#INSTANCE1` and the collector name is Monitoring Agent for Microsoft SQL Server: Collector `SQLTEST#INSTANCE1` then the same service name must be present on all nodes of cluster.

Adding an Microsoft SQL Server agent instance to the cluster

You can use the cluster utility to add an Microsoft SQL Server agent instance to a cluster group in a cluster environment.

Procedure

1. To run the utility, complete one of the following steps:
   - For a 64-bit agent, go to the *candle_home*\TMAITM6_x64 directory.
   - For a 32-bit agent, go to the *candle_home*\TMAITM6 directory.
2. To run the Cluster Utility, double-click the `KoqClusterUtility.exe`.
3. In the SQL **Server Agent Instances Available** area, select a Microsoft SQL Server agent instance, and click **Add**.
4. In the **Select cluster group name** window, select a cluster group.

   The cluster group that you select must be the SQL Server instance that is monitored by the Microsoft SQL Server agent.

5. In the **Select Path for Shared Location** window, navigate to the path where the agent and collector logs are stored.

   If you do not select the path, by default, the `CANDLEHOME/TMAITM6(_x64)/logs` location is selected for storing the agent and collector logs.
6. To add the Microsoft SQL Server agent instance to the cluster environment, click **OK**.

   The activity logs of the cluster utility are displayed in the **History** pane.

Updating an existing Microsoft SQL Server agent instance in a cluster

You can use the cluster utility to update the location where the agent and collector logs are stored for an SQL Server instance in a cluster.

Procedure

1. To update an existing Microsoft SQL Server agent instance, open the **Cluster Utility** window.
2. In the **SQL Server Agent Instances Configured** area, select a Microsoft SQL Server agent instance, and click **Update**.
3. In the **Set Path for Shared Location** window, navigate to the path where the agent and collector logs are stored.

   If you do not select the path, the agent and collector logs are stored at the location that was set while adding the Microsoft SQL Server agent instance in a cluster.
4. Click **OK**.

   The activity logs of the cluster utility are displayed in the **History** pane.

Removing a Microsoft SQL Server agent instance from a cluster

You can use the cluster utility to remove a Microsoft SQL Server agent instance from a cluster group.

Procedure

1. Open the **Cluster Utility** window.
2. In the **SQL Server Agent Instances Configured** area, select a Microsoft SQL Server agent Instance, and click **Remove**.
3. In the **Please Confirm Action** dialog box, click **Yes** to delete the Microsoft SQL Server agent instance from the cluster.

   The activity logs of the cluster utility are displayed in the **History** pane.

Configuring multiple collations for ERRORLOG file

The Microsoft SQL Server agent version 06.31.17.00 or later for Application Performance Management version 8.1.4.0.4 supports multiple collations in `ERRORLOG` file. You can now configure the agent to parse more than one collations in the `ERRORLOG` file for **Problem Detail** attribute group. Note that the multiple collations in `ERRORLOG` file is not applicable for **Error Event Detail** attribute group.

Before you begin

To configure multiple collations of the agent, ensure that the agent is installed.

About this task

The default collation is English. For other languages of SQL Server, the agent will parse the ERRORLOG file based on the collations in the configuration file `koqErrConfig.ini`. So you must add the collations that are in used in `koqErrConfig.ini` file.

Procedure

To configure multiple collations for the agent, complete the following steps:

1. Go to the agent directory *agent_directory*.

   Windows

- For 64-bits agent, *agent_directory* is `Agent_home`\TMAITM6_x64.
- For 32-bits agent, *agent_directory* is `Agent_home`\TMAITM6.

Linux

- For 64-bits agent, *agent_directory* is `Agent_home`/TMAITM6_x64.

Where *Agent_home* is the agent installation directory.

2. Open the configuration `koqErrConfig.ini` file:
3. Move to the end of the file to add the new collations.

   For example, to enable collation for French, add the following collation settings in **`name-value`** pair format at the end of `koqErrConfig.ini` file.

   ```
   [French]
   Error = Erreur :
   Severity = Gravité :
   State = État :
   ```

   Note: The sample list of collations is available in *`agent_directory`*\koqErrConfigSample.ini.

   Where Windows

   - For 64-bits agent, *agent_directory* is `Agent_home`\TMAITM6_x64.
   - For 32-bits agent, *agent_directory* is `Agent_home`\TMAITM6.

   Linux

   - For 64-bits agent, *agent_directory* is `Agent_home`/TMAITM6_x64.

   Where *Agent_home* is the agent installation directory.

   If the target collation is not available in `koqErrConfigSample.ini`, you can determine the collation keyword values from the ERRORLOG file.
   Adhere to the following collation format when configure the collation settings in `koqErrConfig.ini`.

   ```
   [Section_name]
   Error = Error_value
   Severity = Severity_value
   State = State_value
   ```

   Where

   - *Section_name* is the SQL Server collation name. Ensure the collation name is enclosed with an open bracket "**[**" and a closed bracket "**]**".
   - *Error_value* is the corresponding error keyword found in the ERRORLOG file of your target collation.
   - *Severity_value* is the corresponding severity keyword found in the ERRORLOG file of your target collation.
   - *State_value* is the corresponding state keyword found in the ERRORLOG file of your target collation.

   Important: The keyword values must be the same as the keyword values found in the ERRORLOG file, including the special characters.
4. Save the configuration `koqErrConfig.ini` file.

   Agent restart is not required.

   If the configuration file `koqErrConfig.ini` is not available or the configuration file `koqErrConfig.ini` is empty,the ERRORLOG file will show the default collation as English error message with severity level more than the default severity level, if any.

   If the configuration file `koqErrConfig.ini` is configured correctly, the ERRORLOG file will show the corresponding error messages with severity level more then the default severity level, if any.

   The default severity level is 17.

⚠ **Attention:** The changes made in the `koqErrConfig.ini` file is not preserved during agent upgrade, you must make a backup before performing agent upgrade.

What to do next

Check the **Errorlog Alert** widget or the **Problem Detail** attribute group on Application Performance Management dashboard as the result of the collation settings..

Configuring MongoDB monitoring

The Monitoring Agent for MongoDB requires an instance name. You must manually configure and start the agent instance. The MongoDB agent supports local as well as remote monitoring. Refer the following prerequisites for configuring MongoDB agent for both remote and local monitoring.

Before you begin

- Review the hardware and software prerequisites. For the up-to-date system requirement information, see the the Software Product Compatibility Reports (SPCR) for the MongoDB agent.
- Ensure that the user, who configures the MongoDB agent, has the required roles to collect data for all attributes.
  - To configure the agent on the MongoDB database version 2.4 and version 2.6, the clusterAdmin, readAnyDatabase, and dbAdminAnyDatabase roles must be assigned to the user
  - To configure the agent on the MongoDB database version 3.x and 4.x, the clusterMonitor, readAnyDatabase, and dbAdminAnyDatabase roles must be assigned to the user

  To know about the attribute groups for which these user roles are required, see Table 186 on page 571.
- Use an existing user or create a user in the admin database.

  Important: Before you create a user and grant the required roles to the user, you must connect to the MongoDB database and change the database to admin database. If the mongod or mongos process is running in the authentication mode, enter the required credentials to connect to MongoDB database.

  1. Run the following command to connect to the MongoDB database:

     ```
     mongo IP:port
     ```

     Where

     - *IP* is the IP address of the mongod or mongos process
     - *port* is the port number of the mongod or mongos process
  2. Change the database to the admin database:

     ```
     use admin
     ```
  3. Run one of the following commands to add a user in the MongoDB admin database and assign the required roles to the user:
     - For the MongoDB database version 2.4, run the following command:

       ```
       db.addUser({ user: "username", pwd: "password", roles: [ 'clusterAdmin',
       'readAnyDatabase', 'dbAdminAnyDatabase' ] })
       ```
     - For the MongoDB database version 2.6, run the following command:

       ```
       db.createUser({user: "username", pwd: "password", roles:
       [ 'clusterAdmin', 'readAnyDatabase', 'dbAdminAnyDatabase' ] })
       ```
     - For the MongoDB database version 3.x and 4.x, run the following command:

       ```
       db.createUser({user: "username", pwd: "password", roles:
       [ 'clusterMonitor', 'readAnyDatabase', 'dbAdminAnyDatabase' ] })
       ```
  4. Run the following command to verify that the user is added to the admin database:

```
db.auth("username", "password")
```

Return code **1** indicates that the user is added, whereas the return code **0** indicates that the user addition failed.

The following table contains information about the user roles and the attributes for which these user roles are required:

Table 186. Attributes groups and their required user roles

Roles	**MongoDB database version**	**Attribute groups**
dbAdminAnyDatabase	2.x, 3.x and 4.x	Response Times
readAnyDatabase	2.x, 3.x and 4.x	• Mongod Listing • General Shard Information • Collection Storage • Database Names • Shard Details • Collection Storage Details
clusterAdmin	2.x, 3.x and 4.x	• Mongo Instance Information • Mongo Inst IO Info • MII Copy For APMUI One • MII Copy For APMUI Two • Mongo Inst DB Lock • Locks • MongoDB Locks • WiredTiger Details • MMAPv1 Details
clusterMonitor	2.x, 3.x and 4.x	• Mongo Instance Information • Mongo Inst IO Info • MII Copy For APMUI One • MII Copy For APMUI Two • Mongo Inst DB Lock • Locks • MongoDB Locks • WiredTiger Details • MMAPv1 Details

- For remote monitoring of the MongoDB server, see the two prerequisites

  1. Since MongoDB agent requires mongo shell to collect information remotely from the MongoDB server, the system on whichMongoDB agent is installed and configured must have an instance of MongoDB server. The mongo shell of the MongoDB server on the agent machine is used to connect to the remote MongoDB server for monitoring.
  2. In `/etc/hosts` file of the system that hosts the agent, there is an entry of the remote machine.

About this task

The managed system name includes the instance name that you specify. For example, you can specify the instance name as *`instance_name`*`:`*`host_name`*`:`*pc*, where *pc* is the two character product code of your agent. The managed system name can contain up to 32 characters. The instance name can contain up to 28 characters, excluding the length of your host name. For example, if you specify `Mongo2` as your instance name, your managed system name is `Mongo2:hostname:KJ`.

Important: If you specify a long instance name, the managed system name is truncated and the agent code is not completely displayed.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the "Change history" on page 54.

Remember:

- For the agent to successfully collect data, start the agent with the super (root) user, or use the same user ID to start the agent and the mongod process.
- In an environment where MongoDB runs as a cluster, ensure that you install the agent on the same computer where the router process is running. Configure the agent on the same computer with the IP address and port number of that computer and the setup **TYPE** as 1.
- In an environment where MongoDB runs as a cluster in authentication mode, ensure that you add the same user ID with the required rights on all the shards in the cluster.

You can configure the agent by using the default settings, by editing the silent response file, or by responding to prompts.

Configuring the agent with default settings

For a typical environment, use default settings to configure the agent. When default settings are used for the agent configuration, the agent does not run in the authentication mode.

Procedure

1. Run the following command:

   ```
   install_dir/bin/mongodb-agent.sh config instance_name install_dir/samples/
   mongodb_silent_config.txt
   ```

   Where

   - *instance_name* is the name that you specify for the unique application instance.
   - *install_dir* is the installation directory of the MongoDB agent.

   The default installation directory is `/opt/ibm/apm/agent`.
2. Run the following command to start the agent:

   ```
   install_dir/bin/mongodb-agent.sh start instance_name
   ```

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you need help with troubleshooting, see the Troubleshooting section.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters with default values defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters, and configure the agent.

Before you begin

To run the MongoDB database in the authentication mode, ensure that you configure the agent with a user who has the clusterAdmin, readAnyDatabase, and dbAdminAnyDatabase roles on the MongoDB database.

Procedure

1. In a text editor, open the silent response file that is available at the following path: `install_dir/samples/mongodb_silent_config.txt`.
2. For the **TYPE** parameter, enter one of the following values:
   - 1 for a cluster
   - 2 for a replication set
   - 3 for a stand-alone instance

   By default, the agent monitors a cluster.
3. For the **PORT** parameter, specify the port number of the router for a MongoDB cluster or a mongod instance of the MongoDB replication set that is being monitored.

   Remember: If you do not specify any port number, the agent automatically discovers the port number of the appropriate MongoDB process that is active on the default interface. If no MongoDB process is active on the default interface, then the agent selects the port number of the appropriate MongoDB process that is active on the secondary interface.
4. For the **HOST** parameter, specify the IP address of the MongoDB host system.

   Remember: If you do not specify any IP address, the agent automatically detects the IP address of the appropriate MongoDB process that is active on the default interface. If no MongoDB process is active on the default interface, then the agent detects the IP address of the appropriate MongoDB process that is active on the secondary interface.
5. For the **AUTHENTICATION** parameter, specify YES to indicate that mongoDB is running in the authentication mode. The default value is NO, which indicates that the agent is not running in the authentication mode.

   Remember: When the MongoDB database is running in the authentication mode, the MongoDB agent or any MongoDB client cannot connect to the MongoDB database without credentials. To connect to the database that runs in the authentication mode, specify YES for the **AUTHENTICATION** parameter.

   If you specify YES, complete the following steps:

   a) For the **User Name** parameter, specify a user name for the router or the mongod instance. Ensure that minimum roles are assigned to the user. For information about user roles, see Table 186 on page 571.

   b) For the **Password** parameter, specify the password.
6. Save and close the `mongodb_silent_config.txt` file, and run the following command:

   ```
   install_dir/bin/mongodb-agent.sh config instance_name install_dir/samples/
   mongodb_silent_config.txt
   ```

   Where
   - *instance_name* is the name that you specify for the instance.
   - *install_dir* is the installation directory of the MongoDB agent.
7. Run the following command to start the agent:

   ```
   install_dir/bin/mongodb-agent.sh start instance_name
   ```

   Important: If you upgrade the agent to V1.0.0.9 or later and want to run the agent in the authentication mode, then you must configure the agent again to provide a user name and a password. For collecting data, you must stop and restart the agent after configuration.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you need help with troubleshooting, see the Troubleshooting section.

Configuring the agent by responding to prompts

To configure the agent with custom settings, you can specify values for the configuration parameters when prompted while the script is being run.

Procedure

1. Run the following command:
   `install_dir/bin/mongodb-agent.sh config instance_name`

   Where

   - *instance_name* is the name that you specify for the instance.
   - *install_dir* is the installation directory of the MongoDB agent.
2. When you are prompted to provide a value for the **TYPE** parameter, press `Enter` to accept the default value, or specify one of the following values, and then press `Enter`:

   - 1 for a cluster
   - 2 for a replication set
   - 3 for a stand-alone instance

   By default, the agent monitors a cluster.
3. When you are prompted to provide a value for the **PORT** parameter, press `Enter` to accept the default value, or specify the port number of the router for a MongoDB cluster or a mongod instance of the MongoDB replication set that is being monitored, and then press `Enter`.

   Remember: If you do not specify any port number, the agent automatically discovers the port number of the appropriate MongoDB process that is active on the default interface. If no MongoDB process is active on the default interface, then the agent selects the port number of the appropriate MongoDB process that is active on the secondary interface.
4. When you are prompted to provide a value for the **HOST** parameter, press `Enter` to accept the default value, or specify the IP address of the MongoDB host system, and then press `Enter`.

   Remember: If you do not specify any IP address, the agent automatically detects the IP address of the appropriate MongoDB process that is active on the default interface. If no MongoDB process is active on the default interface, then the agent detects the IP address of the appropriate MongoDB process that is active on the secondary interface.
5. When you are prompted to provide a value for the **AUTHENTICATION** parameter, press `Enter` to accept the default value, or specify whether the agent is running in the authentication mode.

   The default value is `NO`, which indicates that the agent is not running in the authentication mode. Specify YES to indicate that mongoDB is running in the authentication mode.

   Remember: When the MongoDB database is running in the authentication mode, the MongoDB agent or any MongoDB client cannot connect to the MongoDB database without credentials. To connect to the database that runs in the authentication mode, specify YES for the **AUTHENTICATION** parameter.

   If you specify YES, complete the following steps:

   a) For the **User Name** parameter, specify a user name for the router or the mongod instance. Ensure that minimum roles are assigned to the user. For information about user roles, see Table 186 on page 571.

   b) For the **Password** parameter, specify the password.
6. Run the following command to start the agent:
   `install_dir/bin/mongodb-agent.sh start instance_name`

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you need help with troubleshooting, see the Troubleshooting section.

Configuring MySQL monitoring

The Monitoring Agent for MySQL requires an instance name and the MySQL server user credentials. You can change the configuration settings after you create the first agent instance.

Before you begin

- Ensure that a user is created in the MySQL database for running the agent. The user does not require any specific privileges on the MySQL database that is being monitored.
- Review the hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the MySQL agent.

About this task

The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see “Change history” on page 54.

The managed system name includes the instance name that you specify, for example, `*instance_name*:*host_name*:*pc*`, where *pc* is your two character product code. The managed system name can contain up to 32 characters. The instance name that you specify can contain up to 28 characters, excluding the length of your host name. For example, if you specify `MySQL2` as your instance name, your managed system name is `MySQL2:hostname:SE`.

Important: If you specify a long instance name, the managed system name is truncated and the agent code is not completely displayed.

Configuring the agent on Windows systems

You can use the IBM Cloud Application Performance Management window to configure the agent on Windows systems.

Procedure

1. Click **Start > All Programs > IBM Monitoring agents > IBM Performance Management**.
2. In the **IBM Performance Management** window, complete these steps:
   a) Double-click the **Monitoring Agent for MySQL** template.
   b) In the **Monitoring Agent for MySQL** window, specify an instance name and click **OK**.
3. In the **Monitoring Agent for MySQL** window, complete these steps:
   a) In the **IP Address** field, enter the IP address of a MySQL server that you want to monitor remotely. If the agent is installed on the server to be monitored, retain the default value.
   b) In the **JDBC user name** field, enter the name of a MySQL server user. The default value is root.
   c) In the **JDBC password** field, type the password of a JDBC user.
   d) In the **Confirm JDBC password** field, type the password again.
   e) In the **JDBC Jar File** field, click **Browse** and locate the directory that contains the MySQL connector Java file and select it.
   f) Click **Next**.
   g) In the **JDBC port number** field, specify the port number of the JDBC server.

      The default port number is 3306.
   h) From the **Java trace level** list, select a trace level for Java.

      The default value is `Error`.
   i) Click **OK**.

      The instance is displayed in the **IBM Performance Management** window.
4. Right-click the **Monitoring Agent for MySQL** instance, and click **Start**.

Remember: To configure the agent again, complete these steps in the **IBM Performance Management** window:

a. Stop the agent instance that you want to configure.
b. Right-click the **Monitoring Agent for MySQL** instance, and click **Reconfigure**.
c. Repeat steps 3 and 4.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring the agent on Linux systems

You run the configuration script to configure the agent on Linux systems.

Procedure

1. Run the following command:

```
install_dir/bin/mysql-agent.sh config instance_name
```

   Where *instance_name* is the name you want to give to the instance, and *install_dir* is the installation directory for the MySQL agent.

2. When you are prompted to enter a value for the following parameters, press Enter to accept the default value, or specify a different value and press enter.

   - IP Address
   - JDBC user name
   - JDBC password
   - Re-type:JDBC password
   - JDBC Jar File
   - JDBC port number (Default port number is 3306.)
   - Java trace level (Default value is `Error`.)

   For information about the configuration parameters, see "Configuring the agent by using the silent response file" on page 576.

3. Run the following command to start the agent.

```
install_dir/bin/mysql-agent.sh start instance_name
```

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring the agent by using the silent response file

Use the silent response file to configure the agent without responding to prompts when you run the configuration script. You can use the silent response file for configuring the agent on both Windows and Linux systems.

About this task

The silent response file contains the configuration parameters. You edit the parameter values in the response file, and run the configuration script to create an agent instance and update the configuration values.

Procedure

1. In a text editor, open the response file that is available at the following path:

   Linux *install_dir*`/samples/mysql_silent_config.txt`

   Windows *install_dir*`\samples\mysql_silent_config.txt`

   Where *install_dir* is the installation directory of the MySQL agent.
2. In the response file, specify a value for the following parameters:
   - For the **`Server Name`** parameter, specify the IP address of a MySQL server that you want to monitor remotely. Otherwise, retain the default value as `localhost`.
   - For the **`JDBC user name`** parameter, retain the default user name value of `root` or specify the name of a user with privileges to view the INFORMATION_SCHEMA tables.
   - For the **`JDBC password`** parameter, enter a JDBC user password.
   - For the **`JDBC Jar File`** parameter, retain the default path if this path to the MySQL connector for the Java jar file is correct. Otherwise, enter the correct path. The connector is available at the following default path:

     Linux `/usr/share/java/mysql-connector-java.jar`

     Windows `C:\Program Files (x86)\MySQL\Connector J 5.1.26\mysql-connector-java-5.1.26-bin.jar`
   - For the **`JDBC port number`** parameter, retain the default port number of 3306 or specify a different port number.
   - For the **`Java trace level`** parameter, retain the default value of `Error` or specify a different level according to the IBM support instructions.
3. Save and close the response file, and run the following command to update the agent configuration settings:

   Linux *install_dir*`/bin/mysql-agent.sh config` *instance_name* *install_dir*`/samples/mysql_silent_config.txt`

   Windows *install_dir*`\BIN\mysql-agent.bat config` *instance_name* *install_dir*`\samples\mysql_silent_config.txt`

   Where *instance_name* is the name that you want to give to the instance, and *install_dir* is the installation directory of MySQL agent.

   Important: Be sure to include the absolute path to the silent response file. Otherwise, no agent data is displayed in the dashboards.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring NetApp Storage monitoring

The Monitoring Agent for NetApp Storage monitors the NetApp storage systems by using the NetApp OnCommand Unified Manager, the OnCommand API Services, and the OnCommand Performance Manager.

Before you begin

- Review the hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the NetApp Storage agent.
- Ensure that the following components are installed on your machine:
  - OnCommand Unified Manager
  - OnCommand Performance Manager

  - OnCommand API Services

  For information about installing these components, see the NetApp documentation.

- Ensure that the versions of the OnCommand API Services, the OnCommand Unified Manager, and the OnCommand Performance Manager are compatible. For example, to configure the OnCommand API Services V1.0, pair the OnCommand Unified Manager V6.2, V6.1, or V6.0 with the OnCommand Performance Manager V1.1. For compatible product versions, see the Interoperability Matrix Tool.
- Ensure that the user, who connects to the OnCommand Unified Manager, has the GlobalRead privilege for the NetApp storage system that is being monitored. Use an existing user ID with this privilege, or create a new user ID. For information about creating the user ID in your NetApp storage system, see the NetApp documentation.
- Ensure that the user, who you use to configure the OnCommand API Services, is an administrator or a monitor. These user types have default permissions to run the rest API.
- Download the NetApp Manageability SDK JAR file (`manageontap.jar`) from the NetApp website and install the file in the monitoring agent `lib` directory by completing the steps that are mentioned in "Downloading and installing the NetApp Manageability SDK JAR file" on page 578.

About this task

The NetApp Storage agent is a multiple instance agent. You must create the first instance, and start the agent manually.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the "Change history" on page 54.

- To configure the agent on Windows systems, you can use the **IBM Performance Management** window or the silent response file.
- To configure the agent on Linux systems, you can run the script and respond to prompts, or use the silent response file.

Downloading and installing the NetApp Manageability SDK JAR file

The NetApp Storage agent requires the NetApp Manageability SDK JAR file to communicate with a NetApp OCUM server.

About this task

After you install the NetApp Storage agent, download the NetApp Manageability SDK JAR file (`manageontap.jar`) from the NetApp website and install the file in the monitoring agent `lib` directory.

Procedure

1. Download the compressed file that contains the JAR file from the following website: http://communities.netapp.com/docs/DOC-1152.
2. Extract this compressed file and copy the `manageontap.jar` file to the following locations:
   - For 32-bit Windows systems, copy the file to *install_dir*/tmaitm6
   - For 64-bit Windows systems, copy the file to *install_dir*/tmaitm6_x64
   - For 32-bit Linux systems, copy the file to *install_dir*/li6263/nu/lib
   - For 64-bit x86-64 Linux systems, copy the file to *install_dir*/lx8266/nu/lib
   - For 64-bit zLinux systems, copy the file to *install_dir*/ls3266/nu/lib

What to do next

Complete the agent configuration.

Configuring the agent on Windows systems

You can configure the agent on Windows operating systems by using the **IBM Performance Management** window. After you update the configuration values, you must start the agent to save the updated values.

About this task

The NetApp Storage agent provides default values for some parameters. You can specify different values for these parameters.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Monitoring Agent for NetApp Storage**, and then click **Configure agent**.

   Remember: After you configure the agent for the first time, the **Configure agent** option is disabled. To configure the agent again, click **Reconfigure**.
3. In the Monitoring Agent for NetApp Storage window, complete the following steps:
   a) Enter a unique name for the NetApp Storage agent instance, and click **OK**.
   b) On the **Data Provider** tab, specify values for the configuration parameters, and then click **Next**.
   c) On the **OnCommand Unified Manager** tab, specify values for the configuration parameters, and then click **Next**.
   d) On the **OnCommand API Service** tab, specify values for the configuration parameters, and then click **OK**.

   For information about the configuration parameters in each tab of the Monitoring Agent for NetApp Storage window, see the following topics:

   - "Configuration parameters for the data provider" on page 581
   - "Configuration parameters for the OnCommand Unified Manager" on page 582
   - "Configuration parameters for the OnCommand API Service" on page 583
4. In the **IBM Performance Management** window, right-click **Monitoring Agent for NetApp Storage**, and then click **Start**.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you need help with troubleshooting, see the Troubleshooting section.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

The silent response file contains the agent configuration parameters with default values that are defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

- To configure the NetApp Storage agent in the silent mode, complete the following steps:

a) In a text editor, open the netapp_storage_silent_config.txt file that is available at the following path:

   - Linux *install_dir*/samples/netapp_storage_silent_config.txt

     Example /opt/ibm/apm/agent/samples/netapp_storage_silent_config.txt

   - Windows *install_dir*\samples\netapp_storage_silent_config.txt

     Example C:\IBM\APM\samples\netapp_storage_silent_config.txt

b) In the netapp_storage_silent_config.txt file, specify values for all mandatory parameters. You can also modify the default values of other parameters.

   For information about the configuration parameters, see the following topics:

   - "Configuration parameters for the data provider" on page 581
   - "Configuration parameters for the OnCommand Unified Manager" on page 582
   - "Configuration parameters for the OnCommand API Service" on page 583

c) Save and close the netapp_storage_silent_config.txt file, and run the following command:

   - Linux *install_dir*/bin/netapp_storage-agent.sh config *instance_name* *install_dir*/samples/netapp_storage_silent_config.txt

     Example **/opt/ibm/apm/agent/bin/netapp_storage-agent.sh config instance_name /opt/ibm/apm/agent/samples/netapp_storage_silent_config.txt**

   - Windows *install_dir*\bin\netapp_storage-agent.bat config *instance_name* *install_dir*\samples\netapp_storage_silent_config.txt

     Example **C:\IBM\APM\bin\netapp_storage-agent.bat config instance_name C:\IBM\APM\samples\netapp_storage_silent_config.txt**

     Where

     instance_name
        Name that you want to give to the instance.

     install_dir
        Path where the agent is installed.

   Important: Ensure that you include the absolute path to the silent response file. Otherwise, the agent data is not shown in the dashboards.

d) Run the following command to start the agent:

   - Linux *install_dir*/bin/netapp_storage-agent.sh start *instance_name*

     Example **/opt/ibm/apm/agent/bin/netapp_storage-agent.sh start instance_name**

   - Windows *install_dir*\bin\netapp_storage-agent.bat start *instance_name*

     Example **C:\IBM\APM\bin\netapp_storage-agent.bat start instance_name**

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you need help with troubleshooting, see the Troubleshooting section.

Configuring the agent by responding to prompts

To configure the agent on Linux systems, you must run the script and respond to prompts.

Procedure

1. On the command line, enter the following command:

   ```
   install_dir/bin/netapp_storage-agent.sh config instance_name
   ```

   Example **/opt/ibm/apm/agent/bin/netapp_storage-agent.sh config instance_name**

   Where

   instance_name
   Name that you want to give to the instance.

   install_dir
   Path where the agent is installed.

2. Respond to the prompts by referring to the following topics:
   - "Configuration parameters for the data provider" on page 581
   - "Configuration parameters for the OnCommand Unified Manager" on page 582
   - "Configuration parameters for the OnCommand API Service" on page 583
3. Run the following command to start the agent:

   ```
   install_dir/bin/netapp_storage-agent.sh start instance_name
   ```

   Example **/opt/ibm/apm/agent/bin/netapp_storage-agent.sh start instance_name**

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you need help with troubleshooting, see the Troubleshooting section.

Configuration parameters for the data provider

When you configure the NetApp Storage agent, you can change the default values of the parameters for the data provider, such as the maximum number of data provider log files, the maximum size of the log file, and the level of detail that is included in the log file.

The following table contains detailed descriptions of the configuration parameters for the data provider.

Table 187. Names and descriptions of the configuration parameters for the data provider

Parameter name	Description	Mandatory field
Instance Name (**KNU_INSTANCE_NAME**)	The name of the instance. **Restriction:** The Instance Name field displays the name of the instance that you specify when you configure the agent for the first time. When you configure the agent again, you cannot change the instance name of the agent.	Yes
Maximum number of Data Provider Log Files (**KNU_LOG_FILE_MAX_COUNT**)	The maximum number of log files that the data provider creates before it overwrites the previous log files. The default value is 10.	Yes
Maximum Size in KB of Each Data Provider Log (**KNU_LOG_FILE_MAX_SIZE**)	The maximum size in KB that a data provider log file must reach before the data provider creates a new log file. The default value is 5190 KB.	Yes

Table 187. Names and descriptions of the configuration parameters for the data provider (continued)

Parameter name	Description	Mandatory field
Level of Detail in Data Provider Log (**KNU_LOG_LEVEL**)	The level of detail that can be included in the log file that the data provider creates. The default value is 4 (Info). The following values are valid: • 1 (Off): No messages are logged. • 2 (Severe): Only errors are logged. • 3 (Warning): All errors and messages that are logged at the Severe level and potential errors that might result in undesirable behavior. • 4 (Info): All errors and messages that are logged at the Warning level and high-level informational messages that describe the state of the data provider when it is processed. • 5 (Fine): All errors and messages that are logged at the Info level and low-level informational messages that describe the state of the data provider when it is processed. • 6 (Finer): All errors and messages that are logged at the Fine level plus highly-detailed informational messages, such as performance profiling information and debug data. Selecting this option can adversely affect the performance of the monitoring agent. This setting is intended only as a tool for problem determination in conjunction with IBM support staff. • 7 (Finest): All errors and messages that are logged at the Fine level and the most detailed informational messages that include low-level programming messages and data. Choosing this option might adversely affect the performance of the monitoring agent. This setting is intended only as a tool for problem determination in conjunction with IBM support staff. • 8 (All): All errors and messages are logged.	Yes

Configuration parameters for the OnCommand Unified Manager

When you configure the NetApp Storage agent, you can change the default values of the parameters for the OnCommand Unified Manager (OCUM), such as the IP address of the OCUM server, user name, and password.

The following table contains detailed descriptions of the configuration parameters for the data source.

Table 188. Names and descriptions of the configuration parameters for the OnCommand Unified Manager

Parameter name	Description	Mandatory field
Server (**KNU_DATASOURCE_HOST_ ADDRESS**)	The host name or IP address of the NetApp OCUM server to be monitored.	Yes
User (**KNU_DATASOURCE_ USERNAME**)	A user name on the NetApp OCUM server with sufficient privileges to collect data. The default value is admin.	Yes
Password (**KNU_DATASOURCE_ PASSWORD**)	The password of the user that you specify in the **User** parameter.	Yes

Table 188. Names and descriptions of the configuration parameters for the OnCommand Unified Manager (continued)

Parameter name	Description	Mandatory field
Confirm Password	The same password that you specified in the **`Enter Password`** parameter.	Yes
Protocol (**KNU_DATASOURCE_ PROTOCOL**)	The protocol to be used for communicating with the NetApp OCUM server. The default value is HTTPS.	Yes

Configuration parameters for the OnCommand API Service

When you configure the NetApp Storage agent, you can change the default values of the parameters for the OnCommand API Service, such as the host address, user name, and password.

The following table contains detailed descriptions of the configuration parameters for the data source.

Table 189. Names and descriptions of the configuration parameters for the OnCommand API Service

Parameter name	Description	Mandatory field
Host Address (**KNU_API_SERVICES_HOST_ ADDRESS**)	The host name or IP address of the OnCommand API service.	Yes
User (**KNU_API_SERVICES_ USERNAME**)	A user name with sufficient privileges to connect to the OnCommand API service. The default value is admin.	Yes
Password (**KNU_API_SERVICES_ PASSWORD**)	The password of the user that you specify in the **`User`** parameter.	
Confirm Password	The same password that you specified in the **`Enter Password`** parameter.	Yes
KNU_API_SERVICES_PORT	The port indicates the communication protocol of API Services. The default value is 443.	Yes

Configuring Environment Variables

You can configure environment variables to change the behavior of the agent.

About this task

Note: For Windows platform, user can edit the environment variable that will change the behavior of the specific agent instance.
For non-Windows platform, environment variable setting will impact all the running agent instances.

Procedure

1. Stop all the agent instances.
2. Locate the environment variable file.
   - Windows:

     Locate the KNUENV_*instance_name* file by navigating to the agent folder, where *instance_name* is the agent instance name.

    - Agent of 32-bit system: `%CANDLEHOME%\TMAITM6`
    - Agent of 64-bit system:`%CANDLEHOME%\TMAITM6_x64`
  - Non-Windows:

    Locate the `.vm.environment` file by navigating to the agent folder.

    - Agent of 32-bit system: `$CANDLEHOME/config`
    - Agent of 64-bit system: `$CANDLEHOME/config`
3. Edit environment variable according to the requirement and save the file.
   Example:

   ```
   KNU_DATA_PROVIDER_CONNECTION_RETRY_COUNT=0
   ```
4. Start the agent instance.

Configuring Node.js monitoring

You can use either the Node.js agent or the stand-alone Node.js data collector to monitor your Node.js applications. If you want a simpler installation process and transaction tracking function, use the Node.js data collector.

Before you begin

- The directions here are for the most current release of this agent and data collector. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see "Change history" on page 54.
- Make sure that the system requirements for the Node.js agent or Node.js data collector are met in your environment.
  - For the up-to-date system requirement information of Node.js agent, see the Software Product Compatibility Reports (SPCR) for the Node.js agent.
  - For the up-to-date system requirement information of Node.js data collector, see the Software Product Compatibility Reports (SPCR) for the Node.js data collector.

About this task

The following procedure is a roadmap for configuring the Monitoring Agent for Node.js and the stand-alone Node.js data collector, which includes both required and optional steps.

- To monitor on-premises applications, you can configure the stand-alone Node.js data collector or the Node.js agent. If you want to enable transaction tracking for your Node.js applications, configure the stand-alone data collector.
- To monitor IBM Cloud(formerly Bluemix) or Kubernetes applications, configure the stand-alone Node.js data collector.

Complete the following steps according to your needs.

Procedure

- Configure the Node.js agent to monitor your on-premises applications.
  a) Add one agent data collector to your Node.js applications for the agent to work properly. See "Configuring the Node.js agent" on page 585.
  b) Optional: To change the monitoring behavior of your agent, see Configuring the Node.js agent data collector.
  c) Optional: To configure diagnostic data collection and display, see Configuring the diagnostics data collector.
- Configure the stand-alone Node.js data collector to monitor IBM Cloud applications.

a) To configure the stand-alone Node.js data collector, see "Configuring the stand-alone Node.js data collector for IBM Cloud(formerly Bluemix) applications" on page 590.

b) To change the behavior of the stand-alone Node.js data collector, see "Customizing the stand-alone Node.js data collector for IBM Cloud applications" on page 592.

- Configure the stand-alone Node.js data collector to monitor on-premises applications.

  a) To configure the stand-alone Node.js data collector, see "Configuring the stand-alone Node.js data collector for on-premises applications" on page 595.

  b) To change the behavior of the stand-alone Node.js data collector, see "Customizing the Node.js data collector for on-premises applications" on page 597.

- Configure the stand-alone Node.js data collector to monitor applications on Kubernetes.

  a) To configure the stand-alone Node.js data collector, see "Configuring the stand-alone Node.js data collector for Kubernetes applications" on page 600.

  b) To change the behavior of the stand-alone Node.js data collector, see "Customizing the stand-alone Node.js data collector for Kubernetes applications" on page 601.

Configuring the Node.js agent

You must add an agent data collector to your Node.js application and restart it before the agent can start monitoring your application.

Before you begin

Before you reconfigure the agent settings within the same version, use the following steps to clean the data collector files that were created by the previous configuration:

1. Go to the *install_dir*/lx8266/nj/bin directory.
2. Run the ./uninstall.sh command to remove existing data collector files.

About this task

The Node.js agent is a single instance agent. It registers subnodes for each monitored Node.js application. The subnode is in the following structure:

NJ:hostname_port:NJA

Tip: If one Node.js application listens on multiple port numbers, then the lowest port number is used.

You must add an agent data collector to your Node.js application, and restart your application before the agent can begin monitoring your application. The agent data collectors collect data that is forwarded to the Node.js agent. Currently, the following agent data collectors are provided:

- The resource data collector collects resource monitoring data from your Node.js applications.
- The diagnostics data collector collects diagnostic data and resource monitoring data from your Node.js applications.
- The method trace data collector collects method traces, diagnostics data, and resource monitoring data from your Node.js applications.

Procedure

1. Make sure that the user ID that is used to run the application server has full permission to the install_dir directory of the agent.
2. Go to the directory *install_dir*/bin and run the following command:

   ```
   ./nodejs-agent.sh config
   ```

3. Follow the prompts to specify values for the following configuration options:

   KNJ_NODEJS_RUNTIME_BIN_LOCATION
   The directory to the bin folder of your Node.js runtime. The default directory is /usr/local/bin.

KNJ_NPM_RUNTIME_BIN_LOCATION
The directory to the bin folder of you **npm** command. The default directory is /usr/local/bin.

KNJ_NPM_LIB_LOCATION
The directory to the lib folder of your npm package global installation directory. The default directory is /usr/local/lib. For example, if you install npm package by running npm install -g command, the package is installed to */nodejs_home*/lib/node_modules and the **KNJ_NPM_LIB_LOCATION** is */nodejs_home*/lib.

CP_PORT
The port that the agent listens on for data from socket clients. A value of 0 indicates that an ephemeral port will be used. The default value is 63336.

Note: Don't use the port number that is already used in your system. To check whether the port is already in use, run the netstat -apn | grep *port_number* command.

4. Start the agent by running the following command:

```
./nodejs-agent.sh start
```

5. Verify that the Node.js agent is started successfully. The *KNJ_NPM_LIB_LOCATION*/node_modules/ibmapm folder is generated if the agent starts successfully.
6. Based on the offering that you have and your requirements, insert one of the following entries to the .js file of your Node.js application to configure the agent data collectors:

   Note: Only one entry can be added to your Node.js application to enable agent data collector capabilities. Also, if you enable capabilities that are not included in the offering, unnecessary overhead can happen, which decreases application execution efficiency.

   - If you have only resource monitoring capabilities, you can add the resource data collector. To add it, insert the following line in the beginning of the Node.js application file:

     ```
     require('KNJ_NPM_LIB_LOCATION/node_modules/ibmapm');
     ```

     If the value of **KNJ_NPM_LIB_LOCATION** on your environment is /usr/local/lib, the line is

     ```
     require('/usr/local/lib/node_modules/ibmapm');
     ```

   - If you have diagnostics in addition to resource-level monitoring capabilities, you can choose to add one of the following agent data collectors:
     - To add the method trace data collector, insert the following line in the beginning of the Node.js application file:

       ```
       require('KNJ_NPM_LIB_LOCATION/node_modules/ibmapm/methodtrace.js');
       ```

     - To add the diagnostics data collector, insert the following line in the beginning of the Node.js application file:

       ```
       require('KNJ_NPM_LIB_LOCATION/node_modules/ibmapm/deepdive.js');
       ```

     - To add the resource monitoring data collector, insert the following line in the beginning of the Node.js application file:

       ```
       require('KNJ_NPM_LIB_LOCATION/node_modules/ibmapm');
       ```

     To guarantee best performance, add the method trace data collector only for debugging purposes.

   Note: The code of the plug-ins changes from Cloud APM March, 2017. If you upgrade your agent from earlier versions, you must update the code of the existing data collectors in your applications for the monitoring ability to work properly.
7. Restart your Node.js application to enable the agent data collector.

Results

You have successfully configured the Node.js agent.

What to do next

- Now, you can verify that the Node.js agent data is displayed in the Cloud APM console. For instructions on how to start the Cloud APM console, see Starting the Cloud APM console. For information about using the Applications editor, see Managing applications.

  Important: To add your application to the Cloud APM console, choose **Node.js** in the application editor.
- You can change the runtime behavior of Node.js agent data collectors. For more information, see Configuring the Node.js agent data collector.
- You can enable diagnostics data collection and display by configuring the diagnostics data collector. For more information, see Configuring the diagnostics data collector.

Configuring the Node.js agent data collector

You can change the behavior of each Node.js agent data collector by changing its runtime configuration in its configuration file.

Runtime configuration file

The Node.js data collector code is in the following directory:

`KNJ_NPM_LIB_LOCATION/node_modules/ibmapm`

where *KNJ_NPM_LIB_LOCATION* is the directory to the lib folder of your npm package global installation directory. The default directory is `/usr/local/lib`.

There is also a runtime configuration file for each agent data collector in the same folder. The agent data collector reads the configuration file every minute.

Tip: The runtime configuration file is named in the following format:

`plugin_application port number_conf.json`

When you change the content of the configuration file, the behavior of the associated agent data collector changes. There are two types of information in the configuration file that you can change:

- URL filtering rules
- agent data collector logging parameters

URL filtering rules

You can change URL filtering rules in the runtime configuration file. Regular expressions are used to map the URL path name to a user customized path name. You can map the URL to a customized path name to satisfy the following requirements:

- Aggregating URLs with similar paths. For example, you have the following URL paths:

  ```
  /demo/poll/1
  /demo/poll/2
  /demo/poll/3
  ...
  ```

  On the web server, requests for these paths are likely served by a common routine, so you can aggregate the paths to a single URL type by using the filter in the following example.

  ```
  "filters":
  [
      {
          "pattern": "/demo/poll/.+",
          "to": "/demo/poll/"
      }
  ```

  This filter results in all requests to URL paths like `"/demo/poll/xxx"` being mapped to a URL path type of `"/demo/poll"`. The response time for all requests to URL paths of this type are then averaged to a single value. Aggregating in this way can help you make more efficient use of available resources.

- Ignoring URL paths to static files or filtering out certain types of requests. For example, if a web page includes images that generate separate server download requests, you might not be interested in seeing response times for these types of requests.

  To filter out a type of request, set the "to" value to empty as in the format of the following example:

  ```
  "filters":
  [
      {
          "pattern": "GET /css/.+\\.css$",
          "to": ""
      }
  ```

  This filter causes the requests to get a `.css` file to be ignored. As a result, you can use the available resources more efficiently on the requests that you need to monitor.

In the configuration file, URL filter rules are provided in a JSON array named `filters`:

```
"filters":
[
    {
        "pattern": ".+\\.png$",
        "to": ""
    },
    {
        "pattern": ".+\\.jpg$",
        "to": ""
    },
    {
        "pattern": "GET /js/.+\\.js$",
        "to": ""
    },
    {
        "pattern": "GET /css/.+\\.css$",
        "to": ""
    }
]
```

Each member in the array is a filtering rule. When an HTTP request is received by the agent data collector, the agent data collector extracts the URL path name from the request and compares it with each `"pattern"`. If the path name does not match a `"pattern"`, the original URL path name is kept and used for measurements.

Agent data collector logging parameter

You can change logging behaviors by modifying the parameter in the `config.properties` configuration file in the *KNJ_NPM_LIB_LOCATION*/node_modules/ibmapm/etc directory. The following logging parameter is provided for you to change:

The log level

The entry in the configuration file for log level is `KNJ_LOG_LEVEL=info`, which means the summary information about the actions is printed in the log. You can set the log level by changing the value of `KNJ_LOG_LEVEL`. The default value is `info` and the log is printed to standard output.

The following five log level values are supported:

off

Logs are not printed.

error

Information is only logged on an error condition.

info

Information is logged when the Node.js agent data collector is running normally. The raw monitoring data that is sent to the agent is also logged.

debug

Debug, info, and error information are printed in the log, for example, collected data, data that is sent to server, and server response.

all

All information is printed in the log.

Configuring the Node.js agent diagnostics data collector

Support for diagnostics data collection is disabled by default. If you have diagnostic capabilities, you must set and adjust the data collection for specific Node.js applications.

Procedure

- To modify the data collector settings of a specific application that is running:
  1. Navigate to the *KNJ_NPM_LIB_LOCATION*/`node_modules/ibmapm` directory, and open the file `plugin_`*`port`*`_conf.json` in a text editor.

     Tip: For information about *KNJ_NPM_LIB_LOCATION*, see the parameter description of "KNJ_NPM_LIB_LOCATION" on page 586
  2. Use the following table for information on modifying the data collector settings:

Table 190. Data collector settings

Diagnostic data category	Description	Property	Action
Minimum time delta for stack trace reporting	Specifies the response time threshold for collecting the stack trace of a request or a method call. If the response time of a request or method call exceeds this value, the data collector collects its stack trace.	`minClockStack`	Set to a value in milliseconds
Minimum time delta to report requests	Specifies the response time threshold for collecting the method trace of a request instance. If the response time of a request instance exceeds this threshold, the data collector collects its method trace.	`minClockTrace`	Set to a value in milliseconds
Maximum number of events per file	Specifies the maximum number of events to be recorded in a `.jso` file. The `.jso` file records the diagnostic data for these events.	`eventsPerFile`	Set to a maximum number of events value

Table 190. Data collector settings (continued)

Diagnostic data category	Description	Property	Action
Maximum amount of time to report to one file	Specifies the maximum amount of time for the `.jso` file to record diagnostic data	`fileCommitTime`	Set to the maximum time in seconds
Maximum number of files to keep before the oldest ones are deleted	Specifies the maximum number of `.jso` files to be kept before the oldest ones are deleted.	`maxFiles`	Set to the maximum number of files
Request sampling period	Specifies the sampling period for requests.	`sampling`	Set to the wanted sampling period. The default value is `10`. A value of `10` means that the agent collects one of every 10 requests.

- Optional: Set the *SECURITY_OFF* environment variable if you want the diagnostics data collector to collect user sensitive information such as cookies, HTTP request contexts, and database requests context. This information is not collected by default.

  Use caution when you are setting this variable because it might cause information to be leaked.

  Linux For example, to set this environment variable, issue the following command:

  ```
  export SECURITY_OFF=true
  ```

Results

The configuration of the diagnostics data collector is changed for the running application that you specified or for all applications.

Configuring the stand-alone Node.js data collector for IBM Cloud(formerly Bluemix) applications

To collect information about Node.js applications on IBM Cloud, you must configure the stand-alone Node.js data collector.

Before you begin

1. Make sure that your Node.js application can run successfully locally. The stand-alone Node.js data collector can monitor Node.js V8.0.0 and future fix packs, V10.0.0 and future fix packs, and V12.0.0 and future fix packs.
2. Download the data collector package from IBM Marketplace. For detailed instructions, see "Downloading your agents and data collectors" on page 105.

Procedure

1. Extract files from the data collector package. The `nodejs_datacollector_8.1.4.0.6.tgz` package is included in the extracted directory.
2. Extract the `nodejs_datacollector_8.1.4.0.6.tgz` file, for example, by running the following command:

   ```
   tar -zxf nodejs_datacollector_8.1.4.0.6.tgz
   ```

3. Extract the `ibmapm.tgz` file in the `nodejs_dc` folder by running the following command:

   ```
   tar -zxf nodejs_dc/ibmapm.tgz
   ```

   You will get an `ibmapm` folder.

4. Copy the `ibmapm` folder that is extracted from the data collector package to the home directory of your application, for example, by running the following command:

   ```
   cp –r directory_to_the_ibmapm_folder  home_directory_of_your_Node.js_application
   ```

   Tip: The home directory of your Node.js application is determined by the command that you use to start the Node.js application and the directory that contains your main file. If you use the **node app.js** command to start your Node.js application and the `app.js` main file is in the `/root/nodejs_app` directory, `/root/nodejs_app` is the home directory of your application.

5. In the `package.json` file of your Node.js application, add the following line to the dependencies section:

   ```
    “ibmapm": "./ibmapm"
   ```

   Remember: Do not miss the comma at the end of each line in the file except the last one, and keep the `package.json` file in good form.

   Example:

   ```
    "dependencies": {
       "ibmapm": "./ibmapm",
       "cors": "^2.5.2",
       "helmet": "^1.3.0",
       "loopback": "^2.22.0",
       "loopback-boot": "^2.6.5",
       "loopback-datasource-juggler": "^2.39.0",
       "serve-favicon": "^2.0.1",
       "strong-error-handler": "^1.0.1"
     }
   ```

6. Add the following line to the beginning of the main file of your Node.js application:

   ```
   require('ibmapm');
   ```

   If you start your application by running the **node app.js** command, `app.js` is the main file of your application.

7. From the directory that contains the `manifest.yml` file of your Node.js application, log in to IBM Cloud and then run the following command:

   ```
   cf push
   ```

   Tip: For a sample `manifest.yml` file, see “Sample manifest.yml file” on page 188.

Results
The data collector is configured and is connected to the Cloud APM server.

What to do next

You can verify that the monitoring data for your IBM Cloud application is displayed in the Cloud APM console. For instructions on how to start the Cloud APM console, see Starting the Cloud APM console. For information about using the Applications editor, see Managing applications.

Remember: To add your application to the Cloud APM console, choose **Node.js Runtime** in the application editor.

Customizing the stand-alone Node.js data collector for IBM Cloud applications

You can add environment variables on the IBM Clouduser interface to customize the monitoring for your IBM Cloud application.

User-defined environment variables for the Node.js data collector

You can use the information in the following table to customize Node.js monitoring on IBM Cloud.

Table 191. Supported user-defined environment variables for Node.js monitoring on IBM Cloud

Variable name	Importance	Value	Description
KNJ_SAMPLING	Optional	Sampling count of requests	The number of requests based on which a sample is taken. The default value is `10`, which means that one out of every 10 requests is monitored. If you do not set this variable, the default value `10` is used.
KNJ_MIN_CLOCK_TRACE	Optional	Response time threshold for collecting method trace, in milliseconds	If the response time of a request instance exceeds the value of this variable, the data collector collects its method trace. The default value is `0`. If you do not set this variable, the default value `0` is used.
KNJ_MIN_CLOCK_STACK	Optional	Response time threshold for collecting stack trace, in milliseconds	If the response time of a request instance exceeds the value of this variable, the data collector collects its stack trace. The default value is `0`. If you do not set this variable, the default value `0` is used.
KNJ_ENABLE_METHODTRACE	Optional	• `True` • `False`	Enables or disables method trace. • If you set this variable to true, method trace for requests is disabled. • If you set this value to false, method trace for requests is enabled. This is the default value. If you do not set this variable, the default value `False` is used and method trace for requests is enabled.
KNJ_ENABLE_DEEPDIVE	Optional	• `True` • `False`	If you set this variable to true, then the diagnostic data is sent to the server. By default, this value is set to false, which means that diagnostic data is not sent to the server.

Table 191. Supported user-defined environment variables for Node.js monitoring on IBM Cloud (continued)

Variable name	Importance	Value	Description
KNJ_ENABLE_TT	Optional	• `true` • `false`	Enables or disables the transaction tracking of AAR. • If you set this variable to true, transaction tracking of AAR is enabled. • If you set this variable to false, transaction tracking of AAR is disabled. By default, this value is not set, which means transaction tracking is disabled.
KNJ_AAR_BATCH_FREQ	Optional	Interval at which AAR data is sent, in seconds	Specifies the interval at which the AAR data is batched and sent to the server, in seconds. The default value is 60, which means the AAR data is batched and sent to the server every minute. **Note:** This variable works with KNJ_AAR_BATCH_COUNT to determine when AAR data is batched and sent to the server. When the condition that is set by either of the two variables is met, AAR data is batched and sent. When the requests that the AAR data contains reaches the maximum number, for example 100, at a shorter interval than what is set, the data is still batched and sent immediately.
KNJ_AAR_BATCH_COUNT	Optional	Maximum number of requests that a batch of AAR data contains	Specifies the maximum number of requests that a batch of AAR data can contain before it is sent to the server. The default value is 100, which means when the number of requests that a batch of AAR data contains reaches 100, this batch of AAR data is sent to the server.

Table 191. Supported user-defined environment variables for Node.js monitoring on IBM Cloud (continued)

Variable name	Importance	Value	Description
KNJ_LOG_LEVEL	Optional	Level of information that is printed in the log	Controls the level of information that is printed in the log. The following levels are provided: **off** Logs are not printed. **error** Information is only logged on an error condition. **info** Information is logged when the Node.js agent data collector is running normally. The raw monitoring data that is sent to the agent is also logged. **debug** The debug, info, and error information are printed in the log, for example, collected data, data that is sent to server, and server response. **all** All information is printed in the log. By default, the log level is `info`, which means the summary information about the data collector actions is printed in the log. Logs are printed to standard output.
SECURITY_OFF	Optional	• `true` • `false`	Enables or disables collection of user sensitive information, such as cookies, HTTP request context, and database request context. • If you set this variable to true, user sensitive information is collected. • If you set this variable to false, user sensitive information is not collected. This is the default value. If you do not specify this variable, the default value of `false` is used and user sensitive information is not collected.

Unconfiguring the stand-alone Node.js data collector for IBM Cloud applications

If you do not need to monitor your Node.js environment or if you want to upgrade the stand-alone Node.js data collector, you must first unconfigure previous settings for the stand-alone Node.js data collector.

Procedure

1. Remove the `require('ibmapm');` line from the application main file.

Tip: If you start your application by running the **node app.js** command, app.js is the main file of your application.

2. Remove the following dependencies from the package.json file.

```
“ibmapm": "./ibmapm"
```

Remember: Do not remove the dependencies that your application needs.

3. Delete the ibmapm folder from the home directory of your application.

Results

You have successfully unconfigured the stand-alone Node.js data collector.

What to do next

After you unconfigure the data collector, the Cloud APM console continues to display the data collector in any applications that you added the data collector to. The Cloud APM console will show that no data is available for the application and will not indicate that the data collector is offline. For information about how to remove the data collector from applications and from resource groups, see “Removing data collectors from Cloud APM console” on page 188.

Configuring the stand-alone Node.js data collector for on-premises applications

If you installed the Node.js application in on-premise environment, you must configure the Node.js data collector to collect information about the Node.js application.

Before you begin

1. Make sure that your Node.js application can run successfully locally. The stand-alone Node.js data collector can monitor Node.js V8.0.0 and future fix packs, V10.0.0 and future fix packs, and V12.0.0 and future fix packs.
2. Download the data collector package from IBM Marketplace. For detailed instructions, see “Downloading your agents and data collectors” on page 105.

Procedure

1. Extract files from the data collector package. The nodejs_datacollector_8.1.4.0.6.tgz package is included in the extracted directory.
2. Determine the home directory of your application.
   - For typical Node.js applications, if you use the **node app.js** command to start your Node.js application and the app.js main file is in the /root/nodejs_app directory, /root/nodejs_app is the home directory of your application.
   - For collective members in the IBM API Connect environment, run the **wlpn-server list** command to display the list of all your collective member on the same machine. The home directory of your collective member is in the following format:

     ```
     user_directory/collective-member_name/package
     ```

     For example, if you get /root/wlpn/rock-8345a96-148538-1/package as a command output, /root/wlpn is the user directory and rock-8345a96-148538-1 is the collective member name.
   - For Developer Portal applications in the IBM API Connect environment, you can run the **ps -ef | grep node** command to find the home directory. If you get the following command output, for example, the home directory is /home/admin/bgsync and the main file of your application is rest_server.js:

     ```
     admin 19085 1 0 Jun25 ?  00:06:53 /usr/local/bin/node /home/admin/bgsync/
     rest_server.js
     ```

3. From the home directory of your application, run the following command to extract files from the data collector package:

   ```
   tar -zxf nodejs_datacollector_8.1.4.0.6.tgz
   ```

4. Extract the `ibmapm.tgz` file in the `nodejs_dc` folder by running the following command:

   ```
   tar -zxf nodejs_dc/ibmapm.tgz
   ```

   You will get an `ibmapm` folder.

5. Run the following command to install the data collector to your application:

   ```
   npm install ./ibmapm
   ```

6. Add the following line to the beginning of the main file of your Node.js application:

   ```
   require('ibmapm');
   ```

   - If you start your application by running the **`node app.js`** command, `app.js` is the main file of your application.
   - For collective members in the IBM API Connect environment, the main file is defined in the `package.json` file in the home directory or its sub-folders. By default, the main file is *`home_directory`*`/server/server.js`, where *home_directory* is the home directory to your collective member.
   - For Developer Portal applications in the IBM API Connect environment, you can run the **`ps -ef | grep node`** command to find the main file. If you get the following command output, for example, the main file of your application is `rest_server.js`.

     ```
     admin 19085 1 0 Jun25 ?  00:06:53 /usr/local/bin/node /home/admin/bgsync/
     rest_server.js
     ```

7. Restart your application.

   Tip:

   - To restart your collective member, run the `wlpn-server stop` *`collective_member_name`* command. The collective member automatically restarts after you run this command. If it does not start, run the `wlpn-server start` *`collective_member_name`* command to restart it manually.
   - To restart your Developer Portal applications, first run the `/etc/init.d/restservice stop` command to stop the application, and then run the `/etc/init.d/restservice start` command to start it.

Results

The data collector is configured and is connected to the Cloud APM server.

What to do next

- You can verify that the monitoring data for your application is displayed in the Cloud APM console. For instructions on how to start the Cloud APM console, see Starting the Cloud APM console. For information about using the Applications editor, see Managing applications.
- To view topology information for your API Connect environment, enable transaction tracking. For more instructions, see description of the *KNJ_ENABLE_TT* variable in "Customizing the Node.js data collector for on-premises applications" on page 597.

Remember: To add your application to the Cloud APM console, choose **Node.js Runtime** in the application editor.

Customizing the Node.js data collector for on-premises applications

By modifying files in the data collector package, you can set the environment variables to customize the monitoring for your Node.js application.

You can set the variables by customizing the environment variables or editing the `config.properties` file. You can find the `config.properties` file in the `ibmapm/etc` folder where your Node.js data collector is installed.

Table 192. Supported variables

Variable name	Importance	Value	Description
KNJ_SAMPLING	Optional	Sampling count of requests	The number of requests based on which a sample is taken. The default value is `10`, which means that one out of every 10 requests is monitored. If you do not set this variable, the default value `10` is used.
KNJ_MIN_CLOCK_TRACE	Optional	Response time threshold for collecting method trace, in milliseconds	If the response time of a request instance exceeds the value of this variable, the data collector collects its method trace. The default value is `0`. If you do not set this variable, the default value `0` is used.
KNJ_MIN_CLOCK_STACK	Optional	Response time threshold for collecting stack trace, in milliseconds	If the response time of a request instance exceeds the value of this variable, the data collector collects its stack trace. The default value is `0`. If you do not set this variable, the default value `0` is used.
KNJ_ENABLE_METHODTRACE	Optional	• `True` • `False`	Enables or disables method trace. • If you set this variable to true, method trace for requests is disabled. • If you set this value to false, method trace for requests is enabled. This is the default value. If you do not set this variable, the default value `False` is used and method trace for requests is enabled.
KNJ_ENABLE_DEEPDIVE	Optional	• `True` • `False`	If you set this variable to true, then the diagnostic data is sent to the server. By default, this value is set to false, which means that diagnostic data is not sent to the server.

Table 192. Supported variables (continued)

Variable name	Importance	Value	Description
KNJ_ENABLE_TT	Optional	• `true` • `false`	Enables or disables the transaction tracking of AAR. • If you set this variable to true, transaction tracking of AAR is enabled. • If you set this variable to false, transaction tracking of AAR is disabled. By default, this value is not set, which means transaction tracking is disabled.
KNJ_AAR_BATCH_FREQ	Optional	Interval at which AAR data is sent, in seconds	Specifies the interval at which the AAR data is batched and sent to the server, in seconds. The default value is 60, which means the AAR data is batched and sent to the server every minute. **Note:** This variable works with KNJ_AAR_BATCH_COUNT to determine when AAR data is batched and sent to the server. When the condition that is set by either of the two variables is met, AAR data is batched and sent. When the requests that the AAR data contains reaches the maximum number, for example 100, at a shorter interval than what is set, the data is still batched and sent immediately.
KNJ_AAR_BATCH_COUNT	Optional	Maximum number of requests that a batch of AAR data contains	Specifies the maximum number of requests that a batch of AAR data can contain before it is sent to the server. The default value is 100, which means when the number of requests that a batch of AAR data contains reaches 100, this batch of AAR data is sent to the server.

Table 192. Supported variables (continued)

Variable name	Importance	Value	Description
KNJ_LOG_LEVEL	Optional	Level of information that is printed in the log	Controls the level of information that is printed in the log. The following levels are provided: **off** Logs are not printed. **error** Information is only logged on an error condition. **info** Information is logged when the Node.js agent data collector is running normally. The raw monitoring data that is sent to the agent is also logged. **debug** The debug, info, and error information are printed in the log, for example, collected data, data that is sent to server, and server response. **all** All information is printed in the log. By default, the log level is `info`, which means the summary information about the data collector actions is printed in the log. Logs are printed to standard output.
SECURITY_OFF	Optional	• `true` • `false`	Enables or disables collection of user sensitive information, such as cookies, HTTP request context, and database request context. • If you set this variable to true, user sensitive information is collected. • If you set this variable to false, user sensitive information is not collected. This is the default value. If you do not specify this variable, the default value of `false` is used and user sensitive information is not collected.

Unconfiguring the stand-alone Node.js data collector for on-premises applications

If you do not need to monitor your Node.js environment or if you want to upgrade the stand-alone Node.js data collector, you must first unconfigure previous settings for the stand-alone Node.js data collector.

Procedure

1. Remove the `require('ibmapm');` line from the application main file.

Tip: If you start your application by running the **node app.js** command, app.js is the main file of your application.

2. Remove “ibmapm": "./ibmapm" from the dependencies section in the package.json file of your Node.js application.
3. Delete the node_modules folder from the home directory of your application.
4. Run the npm install command to install application dependencies.

Results

You have successfully unconfigured the stand-alone Node.js data collector.

What to do next

After you unconfigure the data collector, the Cloud APM console continues to display the data collector in any applications that you added the data collector to. The Cloud APM console will show that no data is available for the application and will not indicate that the data collector is offline. For information about how to remove the data collector from applications and from resource groups, see “Removing data collectors from Cloud APM console” on page 188.

Configuring the stand-alone Node.js data collector for Kubernetes applications

If you installed the Node.js application on Kubernetes, you can configure the Node.js data collector to collect information about the Node.js application.

Before you begin

1. Make sure that your Node.js application can run successfully. The stand-alone Node.js data collector can monitor Node.js V8.0.0 and future fix packs, V10.0.0 and future fix packs, and V12.0.0 and future fix packs.
2. Download the data collector package from IBM Marketplace. For detailed instructions, see “Downloading your agents and data collectors” on page 105.

Procedure

1. Extract files from the data collector package. The nodejs_datacollector_8.1.4.0.6.tgz package is included in the extracted directory.
2. Extract the nodejs_datacollector_8.1.4.0.6.tgz file, for example, by running the following command:

   ```
   tar -zxf nodejs_datacollector_8.1.4.0.6.tgz
   ```

3. Extract the ibmapm.tgz file in the nodejs_dc folder by running the following command:

   ```
   tar -zxf nodejs_dc/ibmapm.tgz
   ```

   You will get an ibmapm folder.
4. In the package.json file of your Node.js application, add the following line to the dependencies section:

   ```
   "ibmapm": "./ibmapm"
   ```

   Remember: Do not miss the comma at the end of each line in the file except the last one, and keep the package.json file in good form.
5. Add the following line to the beginning of the main file of your Node.js application:

   ```
   require('./ibmapm');
   ```

   If you start your application by running the **node app.js** command, app.js is the main file of your application.
6. Rebuild your docker image.

Note: If you run your Node.js app on other Docker environments, for example, Docker Swarm or AWS Docker services,you need to dockerize the steps.

What to do next

If you want to customize the monitoring, you can add environment variables in your deployment yaml file. For details, see "Customizing the stand-alone Node.js data collector for Kubernetes applications" on page 601.

Customizing the stand-alone Node.js data collector for Kubernetes applications

You can add environment variables to the deployment yaml file to customize the monitoring for your Kubernetes application.

User-defined environment variables for the Node.js data collector

You can use the information in the following table to customize Node.js monitoring on Kubernetes.

Table 193. Supported user-defined environment variables for Node.js monitoring on Kubernetes

Variable name	Importance	Value	Description
KNJ_SAMPLING	Optional	Sampling count of requests	The number of requests based on which a sample is taken. The default value is `10`, which means that one out of every 10 requests is monitored. If you do not set this variable, the default value `10` is used.
KNJ_MIN_CLOCK_TRACE	Optional	Response time threshold for collecting method trace, in milliseconds	If the response time of a request instance exceeds the value of this variable, the data collector collects its method trace. The default value is `0`. If you do not set this variable, the default value `0` is used.
KNJ_MIN_CLOCK_STACK	Optional	Response time threshold for collecting stack trace, in milliseconds	If the response time of a request instance exceeds the value of this variable, the data collector collects its stack trace. The default value is `0`. If you do not set this variable, the default value `0` is used.

Table 193. Supported user-defined environment variables for Node.js monitoring on Kubernetes (continued)

Variable name	Importance	Value	Description
KNJ_ENABLE_METHODTRACE	Optional	• `True` • `False`	Enables or disables method trace. • If you set this variable to true, method trace for requests is disabled. • If you set this value to false, method trace for requests is enabled. This is the default value. If you do not set this variable, the default value `False` is used and method trace for requests is enabled.
KNJ_ENABLE_DEEPDIVE	Optional	• `True` • `False`	If you set this variable to true, then the diagnostic data is sent to the server. By default, this value is set to false, which means that diagnostic data is not sent to the server.
KNJ_ENABLE_TT	Optional	• `true` • `false`	Enables or disables the transaction tracking of AAR. • If you set this variable to true, transaction tracking of AAR is enabled. • If you set this variable to false, transaction tracking of AAR is disabled. By default, this value is not set, which means transaction tracking is disabled.
KNJ_AAR_BATCH_FREQ	Optional	Interval at which AAR data is sent, in seconds	Specifies the interval at which the AAR data is batched and sent to the server, in seconds. The default value is 60, which means the AAR data is batched and sent to the server every minute. **Note:** This variable works with KNJ_AAR_BATCH_COUNT to determine when AAR data is batched and sent to the server. When the condition that is set by either of the two variables is met, AAR data is batched and sent. When the requests that the AAR data contains reaches the maximum number, for example 100, at a shorter interval than what is set, the data is still batched and sent immediately.

Table 193. Supported user-defined environment variables for Node.js monitoring on Kubernetes (continued)

Variable name	Importance	Value	Description
KNJ_AAR_BATCH_COUNT	Optional	Maximum number of requests that a batch of AAR data contains	Specifies the maximum number of requests that a batch of AAR data can contain before it is sent to the server. The default value is 100, which means when the number of requests that a batch of AAR data contains reaches 100, this batch of AAR data is sent to the server.
KNJ_LOG_LEVEL	Optional	Level of information that is printed in the log	Controls the level of information that is printed in the log. The following levels are provided: **off** Logs are not printed. **error** Information is only logged on an error condition. **info** Information is logged when the Node.js agent data collector is running normally. The raw monitoring data that is sent to the agent is also logged. **debug** The debug, info, and error information are printed in the log, for example, collected data, data that is sent to server, and server response. **all** All information is printed in the log. By default, the log level is `info`, which means the summary information about the data collector actions is printed in the log. Logs are printed to standard output.

Table 193. Supported user-defined environment variables for Node.js monitoring on Kubernetes (continued)

Variable name	Importance	Value	Description
SECURITY_OFF	Optional	• `true` • `false`	Enables or disables collection of user sensitive information, such as cookies, HTTP request context, and database request context. • If you set this variable to true, user sensitive information is collected. • If you set this variable to false, user sensitive information is not collected. This is the default value. If you do not specify this variable, the default value of `false` is used and user sensitive information is not collected.

Example of yaml file

```
spec:
      containers:
      - name: testapp
        image: mycluster.icp:8500/default/testapp:v1
        imagePullPolicy: Always
        ports:
        - containerPort: 3000
          protocol: TCP
        env:
        - name: KNJ_LOG_LEVEL
          value: "debug"
        - name: KNJ_ENABLE_TT
          value: "true"
        - name: KNJ_ENABLE_DEEPDIVE
          value: "true"
```

Unconfiguring the stand-alone Node.js data collector for Kubernetes applications

If you do not need to monitor your Node.js environment or if you want to upgrade the stand-alone Node.js data collector, you must first unconfigure previous settings for the stand-alone Node.js data collector.

Procedure

1. Remove the `require('ibmapm');` line from the application main file.

   Tip: If you start your application by running the **node `app.js`** command, `app.js` is the main file of your application.

2. Remove the following dependencies from the `package.json` file.

   ```
   "ibmapm": "./ibmapm"
   ```

   Remember: Do not remove the dependencies that your application needs.

Results

You have successfully unconfigured the stand-alone Node.js data collector.

What to do next

After you unconfigure the data collector, the Cloud APM console continues to display the data collector in any applications that you added the data collector to. The Cloud APM console will show that no data is available for the application and will not indicate that the data collector is offline. For information about how to remove the data collector from applications and from resource groups, see "Removing data collectors from Cloud APM console" on page 188.

Configuring OpenStack monitoring

You must configure the Monitoring Agent for OpenStack before the agent can automatically monitor the OpenStack agent environment.

Procedure

1. Configure the agent by responding to prompts. For instructions, see "Configuring the OpenStack agent" on page 605.
2. If you want to collect process-related information, configure the data collector for the OpenStack agent. For instructions, see "Enabling process-related information collection and SSH connections" on page 607.
3. Enter configuration values for the agent to operate. For instructions, see "Adding the configuration values" on page 608.

Configuring the OpenStack agent

For a typical environment, if you want the OpenStack agent to automatically monitor the OpenStack environment, you must configure the agent first.

Before you begin

Make sure that you have installed all required software as described in Preinstallation on Linux systems.

Procedure

You have two options to configure the OpenStack agent on a Linux system:

- To configure the agent by running the script and responding to prompts, see "Interactive configuration" on page 605.
- To configure the agent by editing the silent response file and running the script with no interaction, see "Silent configuration" on page 606.

Interactive configuration

Procedure

1. To configure the agent, run the following command:

   ```
   install_dir/bin/openstack-agent.sh config instance_name
   ```

   where *install_dir* is the installation directory of your OpenStack agent. The default installation directory is `/opt/ibm/apm/agent`.
2. When prompted to `Enter instance name`, specify an instance name.

   Important: The OpenStack agent is a multiple instance agent and requires an instance name for each agent instance. The instance name that you specify is included in the managed system name `instance_name:host_name:sg`. The length of the instance name you specify is limited to 28 characters minus the length of your host name. For example, if you specify OS1 as your instance name, your managed system name is `OS1:hostname:SG`.
3. When prompted to `Edit Monitoring Agent for OpenStack`, press Enter to continue.
4. When prompted to `Edit OpenStack environment authentication information`, provide the following information:

   ```
   OpenStack authentication url (default is: http://localhost:identity/v3):
   OpenStack username (default is: admin):
   Enter OpenStack password (default is: ):
   ```

```
Re-type: OpenStack password (default is: ):
OpenStack tenant name (default is: admin):
```

5. When prompted for `Python Executable Location`, specify the Python executable location, for example, `/usr/bin/python`.

   You can find the fully qualified path by running the following command in your environment:

   ```
   which python
   ```

6. When prompted for `Port Number`, accept the default value or specify a port number.

   This port is used for monitoring internal communication between the OpenStack agent data collector and the OpenStack agent, both of which are installed on a local server only. The agent listens on this port for data from the data collector. The default value of 0 indicates that an ephemeral port is used when the agent starts. On a server with strict security rules on ports, you can configure one specific port for the agent to use. This port is for internal use by the agent and is not related to the OpenStack environment.

7. Edit the `/etc/hosts` file on your system to add host mapping for each monitored node.

Silent configuration

Procedure

1. Open the `sg_silent_config.txt` file in a text editor. The file is in the *`install_dir`*`/samples` directory, where *install_dir* is the installation directory of the OpenStack agent.
2. Edit the `sg_silent_config.txt` configuration file for the OpenStack agent.
3. Specify values for the parameters that are identified in the file. The response file contains comments that define the available parameters and the values to specify.
4. Save the file and exit.
5. Edit the `/etc/hosts` file on your system to add host mapping for each monitored node.
6. From the *`install_dir`*`/samples` directory, run the following command to configure the agent:

   ```
   install_dir/bin/openstack-agent.sh config instance_name path_to_response_file
   ```

   where *install_dir* is the name of the instance to be configured, and *path_to_responsefile* is the full path of the silent response file. Specify an absolute path to this file.

   For example, if the response file is in the default directory, run the following command.

   ```
   /opt/ibm/apm/agent/bin/openstack-agent.sh config instance_name
   /opt/ibm/apm/agent/samples/sg_silent_config.txt
   ```

Results

The agent is configured.

What to do next

- After you finish configuring the agent, you can start the agent instance by running the command:

  ```
  install_dir/bin/openstack-agent.sh start instance_name
  ```

  where *instance_name* is the name of the agent instance to be configured.
- To connect the OpenStack agent to an SSL-enabled OpenStack environment, specify the directory of your OpenStack server SSL certificate by setting the following variable:

  ```
  OS_cert_path=directory of the certificate.crt file
  ```

  The `OS_cert_path` variable is in the `OS_authentication_info` section in the `ksg_dc_`*`instance_name`*`.cfg` file.

- If you want to collect process-related information, configure the data collector for the OpenStack agent by completing the steps in "Enabling process-related information collection and SSH connections" on page 607.
- If you want to change the trace level of the agent for troubleshooting purpose, edit the value of the variable **KBB_RAS1** in the *install_dir*/config/sg.environment file according to the instructions in the file.

Enabling process-related information collection and SSH connections

If you want to collect process-related information, configure the agent data collector for the OpenStack agent and set up SSH connections with the target OpenStack component server.

About this task

You must set up an SSH connection to collect process information before you start the OpenStack agent. To set up the connection, use the **ksg_setup_key.sh** or **ksg_ssh_setup.py** assistance tool provided by the product and described in the following procedure.

If you are familiar with setting up SSH connections, you can also use the **ssh-keygen** and **ssh-copy-id** Linux commands to set up the connection

Procedure

1. Go to the *install_dir*/config directory, where *install_dir* is the agent installation directory.
2. Edit the ksg_dc_*instance_name*.cfg file, where *instance_name* is the name you specified for this agent instance.

   The file is created after the agent instance starts. If the file does not exist, copy *install_dir*/lx8266/sg/bin/ksg_dc.cfg to the *install_dir*/config directory and change the file name to ksg_dc_*instance_name*.cfg.

   For example, if the instance name is OS1, change the name to ksg_dc_OS1.cfg.
3. In the ksg_dc_*instance_name*.cfg file, set the value of the parameter **collect_process_information** to YES.
4. In the **OS_process_collection** section, specify the value for the **ssh_user_host** parameter with the users and host names or IP addresses of the OpenStack component servers according to the format of the following example:

   ```
   ssh_user_host=root@9.112.250.248,user1@hostname
   ```
5. Save the settings.
6. For the settings to take effect, restart the agent instance by running the following commands:

   ```
   install_dir/bin/openstack-agent.sh stop instance_name
   install_dir/bin/openstack-agent.sh start instance_name
   ```
7. Set up the SSH connections with the target component server by using one of the following ways:

   - Set up the connections one by one by using the ksg_setup_key.sh script: go to *install_dir*/lx8266/sg/bin directory and run the ksg_setup_key.sh script with host name or IP and user to build the SSH connections with component servers that are specified in Step 4. If you follow the example that is given in Step 4, you must run the script twice to set up the connection one by one:

     ```
     ./ksg_setup_key.sh 9.112.250.248 root
     ./ksg_setup_key.sh hostname user1
     ```

     Note: You must provide the passwords when you run the scripts for the first time. You don't need to provide the passwords again.
   - Set up the connections one by one or in a batch job by using the ksg_ssh_setup.py tool that is provided by the OpenStack agent in *install_dir*/lx8266/sg/bin. You must install Python pexpect library before you can use this tool.
     - To set up SSH connections one by one, run the command:

```
python ksg_ssh_setup.py -single
```

This command helps you to set up the SSH connection to the remote target server. You must provide the following information:

```
Enter the remote target machine host name or IP address: (Type 'END' to end input.)
Enter the account to access the remote machine(e.g. root):
Enter the above user's password:
```

– To set up SSH connections in a batch job, run the command:

```
python ksg_ssh_setup.py -ssh SSH_file
```

where *SSH_file* is the file that contains the target server, user, and password information. You must create the file according to the `ksg_dc_ssh_list.txt` file in the same directory as the Python tool, and specify the host and user information in the file according to the format of the examples:

```
hostname root passw0rd
9.112.250.248 user1 passw0rd
```

Note: You must set up the connections again only when the user name or password for the target server changes. You do not need to set up the connections again after you restart the agent or change the agent configuration.

Results

The data collector is configured and the SSH connections are set up properly. Now, you can log in to the Cloud APM console and use the Applications editor to add the OpenStack agent instance to the Application Performance Dashboard. When you add the agent instance, choose **OpenStack Environment** from the component list when adding the agent instance.

What to do next

- When you click **API End Point Summary By Sevice Type** > **API Endpoint Details**, you see a `No data available` message in the **API Detecting Failure Times History** and **API Detecting Failure Percent History** group widgets. Click an API endpoint shown in **API Endpoint Details** and you can view monitoring data in the two group widgets.
- When you click **Process Summary By Component** > **Process Details** or **SSH Server Connection Status** > **Process Details**, you see a `No data available` message in the **Process CPU Usage History** and **Process Memory Usage History** group widgets. Click a process shown in **Process Details** and you can view monitoring data in the two group widgets.

Adding the configuration values

For both local and remote configuration, you provide the configuration values for the agent to operate.

When you use the interactive mode to configure the agent, a panel is displayed so you can enter each value. When a default value exists, this value is pre-entered into the field. If a field represents a password, two entry fields are displayed. You must enter the same value in each field. The values that you type are not displayed to help maintain the security of these values.

When you use the silent mode to configure the agent, you can edit the *`response_file`* in the *`install_dir`*`/samples` directory to add the configuration values. After you save the changes, follow the instructions in Step "6" on page 606 and run the following command for the changes to take effect:

```
install_dir/bin/openstack-agent.sh start instance_name
```

where *instance_name* is the name of the agent instance to be configured.

After the configuration completes, you can find the configured values in the `.cfg` file of the agent instance, for example, *`hostname`*`_sg_`*`instance_name`*`.cfg`.

The configuration for this agent is organized into the following groups:

OpenStack environment authentication information (OPENSTACK_CONNECTION)

The OpenStack environment authentication information

The configuration elements defined in this group are always present in the agent's configuration.

This group defines information that applies to the entire agent.

OpenStack authentication url (KSG_OPENSTACK_AUTH_URL)

The auth_url of OpenStack environment

The type is string.

This value is required.

Default value: `http://localhost:identity/v3`

OpenStack password (KSG_OPENSTACK_PASSWORD)

The administrator user's password

The type is password.

This value is required.

Default value: None

OpenStack tenant name (KSG_OPENSTACK_TENANT_NAME)

The OpenStack tenant name, also known as the project name

The type is string.

This value is required.

Default value: admin

OpenStack username (KSG_OPENSTACK_USERNAME)

The administrator user to log in to the OpenStack environment

The type is string.

This value is required.

Default value: admin

Python (KSG_PYTHON)

Python executable location

The configuration elements defined in this group are always present in the agent's configuration.

This group defines information that applies to the entire agent.

Python Executable Location (KSG_PYTHON_LOCATION)

The python executable that will be used to run the OpenStack agent data collector. You can find the fully qualified path by running the following command in your terminal: "which python".

The type is string.

This value is required.

Default value: None

Socket (KSG_SOCKET)

Socket Data Source

The configuration elements defined in this group are always present in the configuration of the agent.

This group defines information that applies to the entire agent.

Port Number (CP_PORT)

The port that the agent will use to listen on for data from socket clients. A value of 0 indicates an ephemeral port will be used. This port does NOT correspond to any ports used by your application. This port is for internal use by the agent.

The type is numeric.

This value is optional.

Default value: 0

Configuring Oracle Database monitoring

The Monitoring Agent for Oracle Database provides monitoring capabilities for the availability, performance, and resource usage of the Oracle database. You can configure more than one Oracle Database agent instance to monitor different Oracle databases. Remote monitoring capability is also provided by this agent.

Before you begin

- Before you configure the Oracle Database agent, you must grant privileges to the Oracle user account that is used by the Oracle Database agent. For more information about privileges, see Granting privileges to the Oracle Database agent user.
- If you are monitoring an Oracle database remotely, the agent must be installed on a computer with either the Oracle database software or the Oracle Instant Client installed.

About this task

The directions here are for the most current release of the agent, except as indicated. For information about how to check the version of an agent in your environment, see Agent version.

For general Oracle database performance monitoring, the Oracle Database agent provides monitoring for the availability, performance, resource usage, and activities of the Oracle database, for example:

- Availability of instances in the monitored Oracle database.
- Resource information such as memory, caches, segments, resource limitation, tablespace, undo (rollback), system metric, and system statistics.
- Activity information, such as OS statistics, sessions, contention, and alert log.

The Oracle Database agent is a multiple-instance agent. You must create the first instance and start the agent manually. Additionally, each agent instance can monitor multiple databases.

The Managed System Name for the Oracle Database agent includes a database connection name that you specify, an agent instance name that you specify, and the host name of the computer where the agent is installed. For example, `pc:connection_name-instance_name-host_name:SUB`, where *pc* is your two character product code and *SUB* is the database type (Possible values are RDB, ASM, or DG). The Managed System Name is limited to 32 characters. The instance name that you specify is limited to 23 characters, minus the length of your host name and database connection. For example, if you specify **dbconn** as your database connection name, **`Oracle02`** as your agent instance name, and your host name is *Prod204a*, your managed system name is `RZ:dbconn-oracle02-Prod204a:RDB`. This example uses 22 of the 23 characters available for the database connection name, agent instance name, and host name.

- If you specify a long instance name, the Managed System name is truncated and the agent code does not display correctly.
- The length of the *connection_name*, *instance_name*, and *hostname_name* variables are truncated when they exceed 23 characters.
- To avoid a subnode name that is truncated, change the subnode naming convention by setting the following environment variables: **`KRZ_SUBNODE_INCLUDING_AGENTNAME`**, **`KRZ_SUBNODE_INCLUDING_HOSTNAME`**, and **`KRZ_MAX_SUBNODE_ID_LENGTH`**.
- If you set **`KRZ_SUBNODE_INCLUDING_AGENTNAME`** to `NO`, the subnode ID part of the subnode name does not include the agent instance name. For example,
  - Default subnode name: *DBConnection-Instance-Hostname*
  - Subnode name with environment variable set to `NO`: *DBConnection-Hostname*

- If you set **KRZ_SUBNODE_INCLUDING_HOSTNAME** to NO, the subnode ID part of the subnode name does not include the host name. For example,
  - Default subnode name: *DBConnection-Instance-Hostname*
  - Subnode name with environment variable set to NO: *DBConnection-Instance*

Procedure

1. To configure the agent on Windows systems, you can use the **IBM Performance Management** window or the silent response file.
   - "Configuring the agent on Windows systems" on page 611.
   - "Configuring the agent by using the silent response file" on page 619.
2. To configure the agent on Linux and UNIX systems, you can run the script and respond to prompts, or use the silent response file.
   - "Configuring the agent by responding to prompts" on page 615.
   - "Configuring the agent by using the silent response file" on page 619.

What to do next

For advanced configuration only, the Oracle database administrator must enable the Oracle user to run the `krzgrant.sql` script to access the database, see Running the krzgrant.sql script.

In the Cloud APM console, go to your Application Performance Dashboard to view the data that was collected. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you are unable to view the data in the agent dashboards, first check the server connection logs and then the data provider logs. The default paths to these logs are as follows:

- Linux AIX `/opt/ibm/apm/agent/logs`
- Windows `C:\IBM\APM\TMAITM6_x64\logs`

For help with troubleshooting, see the Cloud Application Performance Management Forum.

Configuring the agent on Windows systems

You can configure the agent on Windows operating systems by using the **IBM Performance Management** window. After you update the configuration values, start the agent to save the updated values.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Cloud Application Performance Management**.
2. In the **IBM Performance Management** window, right-click the **Monitoring Agent for Oracle Database** template, and then click **Configure agent**.

   Remember: After you configure an agent instance for the first time, the **Configure agent** option is disabled. To configure the agent instance again, right-click on it and then click **Reconfigure**.
3. In the Monitoring Agent for Oracle Database window, complete the following steps:
   a) Enter a unique instance name for the Monitoring Agent for Oracle Database instance, and click **OK**.
4. On the Default Database Configuration pane of the **Configure ITCAM Extended Agent for Oracle Database** window, perform the following steps:
   a) Enter the **`Default Username`**. This is the default database user ID for database connections.

      This user ID is the ID that the agent uses to access the monitored database instance. This user ID must have select privileges on the dynamic performance views and tables that are required by the agent.

b) Enter the **Default Password**. This is the password that is associated with the specified default database user ID.

c) If the Oracle agent version is 8.0, perform this step.

1) Enter the **Oracle JDBC Jar File**. This is the full path to the Oracle JDBC driver jar file used to communicate with the Oracle database. The Oracle Java Database Connectivity (JDBC) driver that supports the Oracle database versions monitored by the Oracle agent must be available on the agent computer.

d) If the Oracle agent version is 6.3.1.10, perform these steps.

1) If the Oracle Database agent is installed on the Oracle database server that is monitored, select **Use libraries in Oracle home** and enter the **Oracle Home Directory**. Optionally for local monitoring, the `Oracle Home Directory` setting can be left blank and the *ORACLE_HOME* system environment variable is used.

2) If the Oracle Database agent is remote from the Oracle database server that is monitored, select **Use libraries in Oracle instant client** and enter the **Oracle Instant Client Installation Directory**.

e) If you need to set advanced configuration options, check **Show advanced options** otherwise, proceed to step 5.

f) `Net Configuration Files Directories` can be left blank and the default directory is used. If the Oracle agent version is 6.3.1.10, you can enter multiple net configuration file directories by using a semi-colon (;) to separate the directories. For Oracle agent version 8.0, only one directory is supported.

This setting contains the Oracle database net configuration file or files. The directory is defined by the *TNS_ADMIN* environment variable for each Oracle database instance. The default directory is `%ORACLE_HOME%\NETWORK\ADMIN`. If this item is not configured, the default directory is used. To disable the use of the default directory, set the following agent environment variable to false: `KRZ_LOAD_ORACLE_NET=false`.

g) Leave the `Customized SQL definition file name` blank. It is not used.

h) Choose whether the default dynamic listener is configured at this workstation.

The default dynamic listener is `(PROTOCOL=TCP)(HOST=localhost)(PORT=1521)`. If the default dynamic listener is configured at this workstation, set this value to Yes.

i) Click **Next**.

5. On the **Instance configuration** pane of the **Configure ITCAM Extended Agent for Oracle Database** window, perform the following steps:

This is where the actual database connection instances are defined. You need to add at least one. This is also where you edit and delete database connection instances. If multiple database connection instance configurations exist, use the **Database connections** option to choose the instance to edit or delete.

a) Press **New** in the **Database connections** section.

b) Enter a **Database Connection Name** as an alias for the connection to the database.

This alias can be anything that you choose to represent the database connection with the following restrictions. Only letters, Arabic numerals, the underline character, and the minus character can be used in the connection name. The maximum length of a connection name is 25 characters.

c) Choose a **Connection Type**

1) (Optional) Basic

The default and most common connection type is **Basic**. If you are unsure which connection type you need, it is suggested that you choose this connection type.

a) Select the **Basic** connection type when the target monitored database is a single instance, such as a standard file system instance or an ASM single instance.

b) Enter the `Hostname` as the host name or IP address for the database.

c) Enter the `Port` number that is used by the database.

d) Select either **Service Name** or **SID**.

i. When **Service Name** is selected, enter the name of the service that is a logical representation of a database, a string that is the global database service name.

A service name is a logical representation of a database, which is the way that a database is presented to clients. A database can be presented as multiple services and a service can be implemented as multiple database instances. The service name is a string that is the global database name, that is, a name composed of the database name and domain name, entered during installation or database creation. If you are not sure what the global database name is, then you can obtain it from the value of the SERVICE_NAMES parameter in the initialization parameter file.

ii. When **SID** is selected, enter the Oracle System Identifier that identifies a specific instance of a running database.

This is the Oracle System Identifier that identifies a specific instance of a database.

Proceed to step 5d.

2) (Optional) TNS

a) Select the **TNS** connection type if the *ORACLE_HOME* system environment variable is set and the TNS alias for the target monitored database is defined in the `$ORACLE_HOME/network/admin/tnsnames.ora` file.

b) Enter the **`TNS alias`** name.

Proceed to step 5d.

3) (Optional) Advanced

a) Select the **Advanced** connection type when there is more than one Oracle Instance across multiple physical nodes for the target monitored database. For example, an ASM with Real Applications Cluster (RAC) database.

b) Enter the **`Oracle Connection String`**.

This attribute supports all Oracle Net naming methods as follows:

- SQL Connect URL string of the form:`//host:port/service name`. For example, `//dlsun242:1521/bjava21`.
- Oracle Net keyword-value pair. For example,

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=dlsun242) (PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=bjava21)))
```

- **TNSNAMES** entries, such as **inst1,** with the *TNS_ADMIN* or *ORACLE_HOME* environment variable set and the configuration files configured.

Proceed to step 5d.

d) Check **Use a different user name and password** for this connection to use different credentials than the default credentials that you set in step 4a and step 4b. Otherwise, proceed to step 5g.

e) Enter the **`Database Username`** for this connection.

This user ID is the ID that the agent uses to access the monitored database instance. This user ID must have select privileges on the dynamic performance views and tables that are required by the agent.

f) Enter the **`Database Password`**. The password that is associated with the specified database user ID.

g) Select a **Role** that matches the permissions that are granted to the database connection's credentials.

The role is the set of privileges to be associated with the connection. For a user that was granted the SYSDBA system privilege, specify a role that includes that privilege. For ASM instances, use the **SYSDBA** or **SYSASM** role.

h) Check **Show remote log monitoring options** if you monitor remote Oracle alert logs from this agent instance, otherwise proceed to step 5k.

i) Enter a path or use **Browse** to select the **`Oracle Alert log file paths`**.

The absolute file paths of mapped alert log files for remote database instances in this database connection. The agent monitors alert logs by reading these files. Usually found at `$ORACLE_BASE/diag/rdbms/`*`DB_NAME`*`/`*`SID`*`/trace/alert_`*`SID`*`.log`. For example, if the *DB_NAME* and *SID* are both db11g and *ORACLE_BASE* is `/home/dbowner/app/oracle`, then the alert log would be found at `/home/dbowner/app/oracle/diag/rdbms/db11g/db11g/trace/alert_db11g.log`.

Windows If the Oracle Database agent runs and reads the alert log files through the network, the remote file path must follow the universal naming convention for Windows systems. For example, `\\tivx015\path\alert_orcl.log`.

Windows

Important: Enter the path and alert log file name together. A mapped network driver is not supported for the alert log path.

Linux AIX If the Oracle Database agent is on a remote server, a locally mounted file system is required for to monitor its remote alert logs.

Windows Multiple files are separated by a semicolon (;).

Linux AIX Multiple files are separated by a colon (:).

Each file is matched to a database instance by using the `alert_`*`instance`*`.log` file name pattern or if it is unmatched, it is ignored.

Local database instance alert log files are discovered automatically.

j) Select or enter the **`Oracle Alert Log File Charset`**. This is the code page of the mapped alert log files.

If this parameter is blank, the system's current locale setting is used, for example:

- ISO8859_1, ISO 8859-1 Western European encoding
- UTF-8, UTF-8 encoding of Unicode
- GB18030, Simplified Chinese GB18030 encoding
- CP950, Traditional Chinese encoding
- EUC_JP, Japanese encoding
- EUC_KR, Korean encoding

For the full list of all the supported code pages, see the ICU supported code pages.

k) Click **Apply** to save this database connection instance's settings in the **Database connections** section.

l) (Optional) Test the new database connection.

1) Select the new database connection in the **Database connections** section.

2) Click **Test connection**.

3) Observe the results in the **Test connection** result window.

- Example successful **Test Result**:

```
Testing connection config1 ...
Success
```

- Example unsuccessful **Test Result**:

```
Testing connection config1 ...
KBB_RAS1_LOG; Set MAXFILES to 1
ORA-12514: TNS:listener does not currently know of service requested in connect
descriptor
Failed
```

m) Click **Next**.

6. Read the information on the **Summary** pane of the **Configure ITCAM Extended Agent for Oracle Database** window, then click **OK** to finish configuration of the agent instance.
7. In the **IBM Performance Management** window, right-click **Monitoring Agent for Oracle Database**, and then click **Start**.

What to do next

- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring the agent by responding to prompts

To configure the agent on Linux and UNIX operating systems, run the command line configuration script and respond to its prompts.

Procedure

1. Open the *`install_dir`*`/bin` directory, where *install_dir* is the installation directory for the Oracle Database agent.
2. (Optional) To list the names of any existing configured agent instances, run the following command: **`./cinfo -o rz`**.
3. To configure the Oracle Database agent, run the following command: **`./oracle_database-agent.sh config`** *instance_name*.
4. When prompted to `Edit 'Monitoring Agent for Oracle Database' settings`, press **Enter**. The default value is Yes.
5. To enter the `Default Database Configuration` information, perform the following steps:

   Note: The `Default Database Configuration` section is not the database connection instance configuration. It is a template section for setting what is used as the default values when you add the actual database connection instance configurations, which begin in step 6.

   a) When prompted for the `Default Username`, type the default database user ID for database connections and press **Enter**.

   This user ID is the ID that the agent uses to access the monitored database instance. This user ID must have select privileges on the dynamic performance views and tables that are required by the agent.

   b) When prompted to `Enter Default Password`, type the password that is associated with the specified default database user ID, and press **Enter**. Then, if prompted, confirm the password.

   c) If the Oracle agent version is 8.0, perform this step.

   1) Enter the **`Oracle JDBC Jar File`**. This is the full path to the Oracle JDBC driver jar file used to communicate with the Oracle database. The Oracle Java Database Connectivity (JDBC) driver that supports the Oracle database versions monitored by the Oracle agent must be available on the agent computer.

   d) If the Oracle agent version is 6.3.1.10, perform these steps.

   1) When prompted for the `Oracle Home Directory`, if the Oracle Database agent is installed on the Oracle database server that is monitored, type the `Oracle` home directory, and press **Enter**. If the Oracle Database agent is not installed on the Oracle database server that will be monitored, leave this setting blank, press **Enter**, and perform the next step. If you want to clear the value for the `Oracle Home Directory` directory, press the space bar, and then press **Enter**.

Note: Optionally for local monitoring, `Oracle Home Directory` and `Oracle Instant Client Installation Directory` can be left blank and the *ORACLE_HOME* system environment variable is used.

2) If the Oracle Database agent is remote from the Oracle database server that is monitored, type the `Oracle Instant Client Installation Directory` directory, and press **Enter**. If you set `Oracle Home Directory` in step "5.d.i" on page 615, this value is ignored.

e) `Net Configuration Files Directories` can be left blank and the default directory is used. If the Oracle agent version is 6.3.1.10, you can enter multiple net configuration file directories by using Windows ";" or Linux AIX ":" to separate the directories. For Oracle agent version 8.0, only one directory is supported. Press **Enter**.

This setting contains the Oracle database net configuration file or files. The directory is defined by the *TNS_ADMIN* environment variable for each Oracle database instance. The default directory is Linux AIX `$ORACLE_HOME/network/admin` or Windows `%ORACLE_HOME%\NETWORK\ADMIN`. If this item is not configured, the default directory is used. To disable the use of the default directory, set the following agent environment variable to false: `KRZ_LOAD_ORACLE_NET=false`.

f) Choose whether the default dynamic listener is configured at this workstation, and press **Enter**.

The default dynamic listener is `(PROTOCOL=TCP)(HOST=localhost)(PORT=1521)`. If the default dynamic listener is configured at this workstation, set this value to True.

g) Leave the `Customized SQL definition file` name blank. It is not used.

6. You are prompted to `Edit 'Database Connection' settings` after seeing the following output on the screen:

```
Instance Configuration :
Summary :
Database Connection :
```

Note: This step is where the actual database connection instances are defined. You need to add at least one. This is also where you edit and delete database connection instances. If multiple database connection instance configurations exist, use the `Next` option to skip the instances that do not need to be edited or deleted until you arrive at the instance you need to edit or delete.

7. To add a new database connection, type 1, and press **Enter**.
8. To enter the database connection information, perform the following steps:

a) When prompted for the `Database Connection Name`, type an alias for the connection to the database and press **Enter**.

This alias can be anything that you choose to represent the database connection with the following restrictions. Only letters, Arabic numerals, the underline character, and the minus character can be used in the connection name. The maximum length of a connection name is 25 characters.

b) When prompted for the `Connection Type`, select one of the following types of connection:

1) (Optional) Basic

The default and most common connection type is **Basic**. If you are unsure which connection type you need, it is suggested that you choose this connection type.

a) Select the **Basic** connection type if the target monitored database is a single instance, such as a standard file system instance or an ASM single instance.

b) When prompted for the `Hostname`, type the host name or IP address for the Oracle database, and press **Enter**.

c) When prompted for the `Port`, type the port number, and press **Enter**.

d) Enter one of the next two settings. Either `Service Name` or `SID`.

i. (Optional) When prompted for the `Service Name`, type the name of the service that is a logical representation of a database, a string that is the global database service name, press **Enter** and proceed to step 8c.

A service name is a logical representation of a database, which is the way that a database is presented to clients. A database can be presented as multiple services and a service can be implemented as multiple database instances. The service name is a string that is the global database name, that is, a name composed of the database name and domain name, entered during installation or database creation. If you are not sure what the global database name is, then you can obtain it from the value of the SERVICE_NAMES parameter in the initialization parameter file. This parameter can be left blank if you set the `SID` in step "8.b.i.4.b" on page 617.

ii. (Optional) When prompted for the `SID`, type the Oracle System Identifier that identifies a specific instance of a running database, press **Enter** and proceed to step 8c.

This parameter is the Oracle System Identifier that identifies a specific instance of a database. If `Service Name` was defined in step "8.b.i.4.a" on page 616, you can leave this item blank.

2) (Optional) TNS

a) Select the **TNS** connection type when the *ORACLE_HOME* system environment variable is set and the TNS alias for the target monitored database is defined in the `$ORACLE_HOME/network/admin/tnsnames.ora` file.

b) Type the TNS alias name, and press **Enter** and proceed to step 8c.

3) (Optional) Advanced

a) Select the **Advanced** connection type when there is more than one Oracle Instance across multiple physical nodes for the target monitored database. For example, an ASM with Real Applications Cluster (RAC) database.

b) Type the Oracle connection string, press **Enter** and proceed to step 8c.

This attribute supports all Oracle Net naming methods as follows:

- SQL Connect URL string of the form:`//host:port/service name`. For example, `//dlsun242:1521/bjava21`.
- Oracle Net keyword-value pair. For example,

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=dlsun242) (PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=bjava21)))
```

- **TNSNAMES** entries, such as **inst1,** with the *TNS_ADMIN* or *ORACLE_HOME* environment variable set and the configuration files configured.

Note: The description that is shown during command-line configuration might have a backslash before colons (\:) and before equal sign symbols (\=). Do not type backslashes in the connection string. They are displayed in the description to escape the normal behavior of interpreting the equals sign as part of a command, and instead interpret it merely as text.

c) Proceed to step 8c.

c) When prompted for the `Database Username`, type the database user ID for the connection, and press **Enter**.

For standard file system instances, this user ID must have select privileges on the dynamic performance views and tables that are required by the agent.

For ASM instances, use an account with the **SYSDBA** or **SYSASM** role. For example, the sys account.

d) When prompted to `Enter Database Password`, type the password that is associated with the specified database user ID.

e) When prompted for `Role`, choose the role that matches the permissions that are granted to the specified user ID, and press **Enter**.

The role is the set of privileges to be associated with the connection. For a user that was granted the SYSDBA system privilege, specify a role that includes that privilege.

For ASM instances, use the **SYSDBA** or **SYSASM** role.

f) When prompted for `Oracle Alert Log File Paths (including alert log file name)`, type the alert log paths, and press **Enter**.

This parameter is for any absolute file paths of mapped alert log files for remote database instances in this database connection. The agent monitors alert logs by reading these files. Usually found at `$ORACLE_BASE/diag/rdbms/`*`DB_NAME`*`/`*`SID`*`/trace/alert_`*`SID`*`.log`. For example, if the *DB_NAME* and *SID* are both db11g and *ORACLE_BASE* is `/home/dbowner/app/oracle`, then the alert log would be found at `/home/dbowner/app/oracle/diag/rdbms/db11g/db11g/trace/alert_db11g.log`.

Windows If the Oracle Database agent runs and reads the alert log files across the network, the remote file path must follow the universal naming convention for Windows systems. For example, `\\tivx015\path\alert_orcl.log`.

Important: Enter the path and alert log file name together. A mapped network driver is not supported for the alert log path.

Linux AIX If the Oracle Database agent runs, a locally mounted file system is required for remote alert logs.

Windows Multiple files are separated by a semicolon (;).

Linux AIX Multiple files are separated by a colon (:).

Each file is matched to a database instance by using the `alert_`*`instance`*`.log` file name pattern or if it is unmatched, it is ignored.

Local database instance alert log files can be discovered automatically.

g) When prompted for the **`Oracle Alert Log File Charset`**, type the code page of the mapped alert log files, and press **Enter**.

If this parameter is blank, the system's current locale setting is used, for example:

- ISO8859_1, ISO 8859-1 Western European encoding
- UTF-8, UTF-8 encoding of Unicode
- GB18030, Simplified Chinese GB18030 encoding
- CP950, Traditional Chinese encoding
- EUC_JP, Japanese encoding
- EUC_KR, Korean encoding

For the full list of all the supported code pages, see the ICU supported code pages.

9. When prompted again to `Edit 'Database Connection' settings`, you see the name of the database connection that you set in step 8a. You can edit it again or delete it. If you have more than one database connection instance that is already configured, use **Next** to step through them.
10. (Optional) To add another database connection to monitor multiple database instances with this agent instance, type 1, press **Enter**, and return to Step 8.
11. When you are finished modifying database connections, type 5, and press **Enter** to exit the configuration process.
12. To start the agent, enter:
    `install_dir``/bin/oracle_database-agent.sh start `*`instance_name`*.

What to do next

- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

The silent response file contains the agent configuration parameters with default values that are defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

1. Open the `oracle_silent_config.txt` file in a text editor:
   - Linux AIX *install_dir*`/samples/oracle_database_silent_config.txt`.
   - Windows *install_dir*`\samples\oracle_database_silent_config.txt`
2. For **`Default Username`**, type the name of the default database user for database connections that are created for this agent instance. For example, **KRZ_CONN_USERID=**`user1`.

   Note: This user must have sufficient privileges to complete the tasks that this agent performs while it is connected to the database, such as querying tables.
3. For **`Default Password`**, you must enter the password that is associated with the specified default database user. For example, **KRZ_CONN_PASSWORD=**`Password`.
4. If the Oracle agent version is 8.0, perform this step.

   a) Enter the **`Oracle JDBC Jar File`**. This is the full path to the Oracle JDBC driver jar file used to communicate with the Oracle database.

   The Oracle Java Database Connectivity (JDBC) driver that supports the Oracle database versions monitored by the Oracle agent must be available on the agent computer.
5. If the Oracle agent version is 6.3.1.10, perform these steps.

   a) If the Oracle Database agent is installed on the Oracle database server that is monitored, type the `Oracle home` directory. For example, **KRZ_ORACLE_HOME=**`home_path`.

   Note: For optional parameters like this one, remove the leading hash symbol (#) to use them.

   If the Oracle Database agent is not installed on the Oracle database server that will be monitored, leave this setting blank and complete the next step.

   Note: Optionally for local monitoring, `Oracle Home Directory` and `Oracle Instant Client Installation Directory` can be left blank (commented out using a hash symbol (#) in the first position of the parameter line in the silent configuration text file) and the *ORACLE_HOME* system environment variable is used.

   b) If the Oracle Database agent is remote from the Oracle database server that is monitored, type the `Oracle Instant Client Installation Directory` directory. If you enter the `Oracle Home Directory` directory in the previous step, this value is ignored.

   - Windows Define the full folder path of the **`Oracle Home`** directory that contains the `Oracle Call Interface (OCI)` library files. If the full path of the `oci.dll` file is `C:\instantclient_10_2\oci.dll` you must define this `C:\instantclient_10_2` path. For example, **KRZ_INSTANT_CLIENT_LIBPATH=**`C:\instantclient_10_2`
   - AIX Define the full folder path of the **`Oracle Home`** directory that contains the `Oracle Call Interface (OCI)` library files. If the full path of the `libocci.so.10.1` file is `/home/tivoli/oci/libocci.so.10.1`, you must define this `/home/tivoli/oci` path. For example, **KRZ_INSTANT_CLIENT_LIBPATH=**`/home/tivoli/oci`

6. `Net Configuration Files Directories` can be left blank and the default directory is used. The Oracle Database agent uses this file path to obtain the `tnsnames.ora` file. This directory is defined by the *TNS_ADMIN* environment variable for each Oracle database instance. The default directory is Linux AIX `$ORACLE_HOME/network/admin` or Windows `%ORACLE_HOME%\NETWORK\ADMIN`. If you enter this setting with multiple net configuration file directories, use Windows ";" or Linux AIX ":" to separate the directories.

   If you are monitoring Oracle databases remotely, you can copy net configuration files from the remote system to the system where the agent is installed. Also, you can merge the content of net configuration files on the remote system to the net configuration files on the system where the agent is installed.
7. For **`Dynamic listener`**, check if the default dynamic listener is configured. The default dynamic listener is (PROTOCOL=TCP)(HOST=localhost)(PORT=1521). If the default dynamic listener is configured, set this value to TRUE as shown here; **`KRZ_DYNAMIC_LISTENER=`**`TRUE`.

   The valid values are TRUE and FALSE.
8. Leave the `Customized SQL definition file` name blank. It is not used.
9. Beginning here the actual database connection instances are defined. You need to add at least one. Entries for one instance are given in the `oracle_silent_config.txt` with the instance name *config1*. If you change the instance name, be sure to change all references.

   This alias can be anything that you choose to represent the database connection with the following restrictions. Only letters, Arabic numerals, the underline character, and the minus character can be used in the connection name. The maximum length of a connection name is 25 characters.
10. For **`Connection Type`**, specify one of the following connection types: **Basic**, **TNS**, or **Advanced**. For example, **`KRZ_CONN_TYPE.config1=`**`Basic`.
11. For the connection type that you selected in the previous step, specify the required parameters:

    Basic

    - For **`Hostname`**, specify the host name or the IP address of the Oracle database, for example: **`KRZ_CONN_HOST.config1=`** `hostname`.
    - For **`Port`**, specify the Listener port for the Oracle database, for example: **`#KRZ_CONN_PORT.config1=`** `1521`.
    - For **`Service Name`**, specify the logical representation of the database by using a string for the global database name, for example: **`KRZ_CONN_SERVICE.config1=`** `orcl`.

      Important: If you do not define the Service Name, you must specify the Oracle System Identifier (SID).

      For the **`Oracle System Identifier (SID)`**, specify an SID that identifies a specific instance of a running database, for example: **`KRZ_CONN_SID.config1=`** `sid`.

    TNS

    For **`TNS alias`**, specify the Network alias name from the `tnsnames.ora` file. For example, **`KRZ_CONN_TNS.config1=`** `tnsalias`.

    Advanced

    For **`Oracle Connection String`**, specify the database connection string for OCI. For example, **`KRZ_CONN_STR.config1=`** `//host:port/service`

    This string supports all Oracle Net naming methods as shown here.

    - For an SQL Connect URL string:

      ```
      //host:[port][/service name]
      ```

    - For an Oracle Net keyword-value pair:

      ```
      "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=dlsun242) (PORT=1521))
      (CONNECT_DATA=(SERVICE_NAME=bjava21)))"
      ```

This string also supports **TNSNAMES** entries, for example, **inst1** where the *TNS_ADMIN* or the *ORACLE_HOME* environment variable is set and the configuration files are configured.

Important: This attribute applies only to the advanced type of connection.

12. For **`Database Username`**, you can specify the name of the database user for the connection, for example: **`KRZ_CONN_USERID=`**`UserID`.

    This user must have sufficient privileges to complete the tasks that the agent requires while it is connected to the database, for example, creating, editing, and deleting tables.

    If this field is empty, the agent uses the default user name in the default database configuration section. If **`Database Username`** was not configured, the default user name is used for this connection.

13. For **`Database Password`**, you can specify the password that is associated with the specified database user, for example: **`KRZ_CONN_PASSWORD=`**`Passsword`.

    If this field is empty, the agent uses the default password in the default database configuration section. If **`Database Password`** was not configured, the default password is used for this connection.

14. For **`Role`**, you can specify the set of privileges that are associated with the connection, for example: **`KRZ_CONN_MODE.config1=`**`DEFAULT`.

    The valid values include *SYSDBA*, *SYSOPER*, *SYSASM*, and *DEFAULT*.

    For a user that is granted the *SYSDBA* system privilege, you can specify a connection that includes this privilege. If this item is not defined, you can assign the *DEFAULT* role to the user.

15. For **`Oracle Alert Log File Paths`**, when the alert log file name is included, you can specify the absolute file path of the mapped alert log files for the remote database instances in this database connection. For example, **`KRZ_LOG_PATHS.config1=`**`AlertLogPath`.

    Windows Use a semicolon (;) to separate the multiple files.

    Linux AIX Use a colon (:) to separate the multiple files.

    Each file is matched to a database instance by the `alert_`*`instance`*`.log` file name pattern. Alternatively, it is ignored if it is not matched.

    The `local database instance alert log` files are discovered automatically.

    If **`Oracle Alert Log File Paths`** was not configured, the `Alert Log` is not available.

16. For **`Oracle Alert Log File Charset`**, you can specify the code page of the mapped alert log files. For example, **`KRZ_LOG_CHARSET.config1=`** `CharSet`

    If this field is empty, the system's current locale setting is used as shown here:

    ```
    ISO8859_1: ISO 8859-1 Western European encoding
    UTF-8: UTF-8 encoding of Unicode
    GB18030: Simplified Chinese GB18030 encoding
    CP950: Traditional Chinese encoding
    EUC_JP: Japanese encoding
    ```

17. Save and close the `oracle_database_silent_config.txt` file. Then, enter:
    `install_dir``/bin/oracle_database-agent.sh config` *`instance_name install_dir`*`/`
    `samples/oracle_database_silent_config.txt`
    where *`instance_name`* is the name that you want to give to the instance.

18. To start the agent, enter:
    `install_dir``/bin/oracle_database-agent.sh start` *`instance_name`*.

What to do next

- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Granting privileges to the Oracle Database agent user

After you install the agent, you must grant privileges to the Oracle user account that is used by the Oracle Database agent.

You can grant privileges for the following users:

- Standard file system (non-ASM) instance users
- ASM with RAC instance non-SYS users

Granting privileges to users for standard file system instances

For standard file system instances, the Oracle user ID that the Oracle Database agent uses must have select privileges on the dynamic performance views, tables, and data dictionary views that are required by the agent. It must also have other Oracle object and system privileges that are necessary to run some database commands.

Procedure

1. (Optional) If an Oracle database user ID does not exist, create this ID by using Oracle facilities and running the following command: `create user` *`UserName`* `identified by` *`Password`*
2. Grant select privileges for the dynamic performance views, tables, and data dictionary views to the Oracle user ID that you created by running the **krzgrant.sql** script that is provided with the Oracle Database agent. This step must be done before you configure the agent. For directions about how to customize and run the **krzgrant.sql** script, see "Customizing the krzgrant.sql script" on page 622 and "Running the krzgrant.sql script" on page 623.

   Note: The select privileges for the dynamic performance views, tables, and data dictionary views rely on the capabilities of the Oracle database in specific application environments. You can grant authorized Oracle privileges to the Oracle database user ID only for the dynamic performance views, tables, and data dictionary views that are used by the Oracle Database agent.
3. Grant other Oracle object privileges and system privileges to the Oracle user ID that the Oracle Database agent uses by using Oracle facilities.

Customizing the krzgrant.sql script

If you do not want to allow Oracle authorized select privileges on some dynamic performance views, tables, and data dictionary views in the **krzgrant.sql** script, you can customize the **krzgrant.sql** script before running it.

Note: The agent instance checks all default privileges in the **krzgrant.sql** script and reports an agent event with a lack of privileges when the agent starts. You can disable privilege checking by using the following variable setting: KRZ_CHECK_ORACLE_PRIVILEGE=FALSE. The test connection step of GUI configuration checks all Oracle privileges that are defined in the `krzgrant.sql` file. If you confirm that the Oracle user has the correct privileges, ignore that checking privileges fails in the test connection step.

Edit the `krzgrant.sql` file in a plain text editor to remove or add the '--' prefix at the beginning of grant statements to skip the granting execution for those unauthorized Oracle tables or views.

For example, change the following lines:

```
     execute immediate 'grant select on DBA_HIST_SNAPSHOT to '||userName;
     execute immediate 'grant select on DBA_HIST_SQLSTAT to '||userName;
     execute immediate 'grant select on DBA_HIST_SQLTEXT to '||userName;
     execute immediate 'grant select on DBA_HIST_SQL_PLAN to '||userName;
     execute immediate 'grant select on DBA_HIST_SYSMETRIC_SUMMARY to '||userName;
```

to these lines:

```
--     execute immediate 'grant select on DBA_HIST_SNAPSHOT to '||userName;
--     execute immediate 'grant select on DBA_HIST_SQLSTAT to '||userName;
--     execute immediate 'grant select on DBA_HIST_SQLTEXT to '||userName;
--     execute immediate 'grant select on DBA_HIST_SQL_PLAN to '||userName;
--     execute immediate 'grant select on DBA_HIST_SYSMETRIC_SUMMARY to '||userName;
```

Granting privileges to non-SYS users for ASM instances

You must connect to ASM instances that are using the SYSDBA and SYSASM roles for users. If you do not want to use the SYS account to connect to ASM instances, create a user account and grant the SYSDBA and SYSASM roles to the account.

Procedure

1. Run the following commands to create a user account and grant roles:
   - Log in to the ASM database with the SYSASM role to create a new user for an agent and grant the SYSDBA role or SYSASM role:
     a.
     ```
     create user UserName identified by Password
     ```
     b.
     ```
     grant sysdba to UserName
     ```
     or
     ```
     grant sysasm to UserName
     ```
2. When you create the ASM connection in the configuration window, specify the *UserName* user and the SYSDBA or SYSASM role.

   Note: If you choose the SYSASM role to access the ASM database, you must configure the agent instance by using Oracle home or Oracle instant client to connect to the Oracle database.

Running the `krzgrant.sql` script

Before you begin

- If you do not run the **`krzgrant.sql`** script, an event is raised in the agent event workspace.
- To complete the installation procedure, see Chapter 6, "Installing your agents," on page 121.

After the installation, you can find the **`krzgrant.sql`** script in the following directory:

- Windows `install_dir\TMAITM6_X64`
- Linux AIX `install_dir/architecture/rz/bin`

where:

install_dir
: Installation directory for the Oracle Database agent.

architecture
: The IBM Application Performance Management or Cloud APM system architecture identifier. For example, lx8266 represents Linux Intel v2.6 (64-bit). For a complete list of the architecture codes, see the `install_dir/registry/archdsc.tbl` file.

The **`krzgrant.sql`** script has the following usage: `krzgrant.sql user_ID temporary_directory`

where:

user_ID
: The ID of the Oracle user. This user ID must be created before you run this SQL file. Example value: *tivoli*.

temporary_directory
: The name of the temporary directory that contains the `krzagent.log` output file of the **`krzgrant.sql`** script. This directory must exist before you run this SQL script. Example value: `install_dir/tmp`.

You must have the Oracle database administrator (DBA) authorization role and write permission to the temporary directory to perform the following procedure.

Procedure

1. From the command line, run the commands to set environment variables.

   - Windows

     ```
     SET ORACLE_SID= sid
     SET ORACLE_HOME= home
     ```

   - Linux AIX

     ```
     ORACLE_SID = sid
     export ORACLE_SID
     ORACLE_HOME = home
     export ORACLE_HOME
     ```

   where:

   sid
   Oracle system identifier, which is case-sensitive.

   home
   Home directory for the monitored Oracle instance.

2. From the same command-line window where you set environment variables, start the Oracle SQL Plus or an alternative tool that you use to issue SQL statements.
3. Log on to the Oracle database as a user that has Oracle DBA privileges.
4. Go to the directory that contains the **krzgrant.sql** script and run the following command to grant select privileges:

   ```
   @krzgrant.sql user_ID temporary_directory
   ```

   The output is logged in the krzagent.log file in the temporary directory. This log records the views and tables to which the Oracle Database agent is granted select privileges.

   After the privileges are successfully granted, you can configure and start the Oracle Database agent.

Configuring OS monitoring

The Monitoring Agent for Linux OS, Monitoring Agent for UNIX OS, and Monitoring Agent for Windows OS agents are configured automatically. You can configure log file monitoring for the OS agents so that you can monitor application log files. You can run the OS agents as a non-root user. Also, there are some additional configuration options for the Linux OS agent.

Running the OS agents as a non-root user

You can run the Monitoring Agent for Windows OS, Monitoring Agent for UNIX OS, and Monitoring Agent for Linux OS as a non-root user.

To run the Windows OS agent as a non-root user, see "Running the Monitoring Agent for Windows OS as a non-root user" on page 625.

To run the Monitoring Agent for UNIX OS and Monitoring Agent for Linux OS agents as a non-root user, see "Starting agents as a non-root user" on page 992.

Restriction:

While running as a nonroot user, the agent cannot access /proc/pid/status, and therefore cannot report the following attributes:

- -User CPU Time (UNIXPS.USERTIME)
- -System CPU Time (UNIXPS.SYSTEMTIM)
- -Total CPU Time (UNIXPS.TOTALTIME)
- -Thread Count (UNIXPS.THREADCNT)

- -Child User CPU Time (UNIXPS.CHILDUTIME)
- -Child System CPU Time (UNIXPS.CHILDSTIME)
- -Total Child CPU Time (UNIXPS.CHILDTIME)
- -Wait CPU Time (UNIXPS.WAITCPUTIM)
- -Terminal (UNIXPS.USERTTY)

These attributes are not visible in the Cloud APM console but are available to create thresholds.

Running the Monitoring Agent for Windows OS as a non-root user

You can run the Windows OS agent as a non-root user. However, some functions are unavailable.

When you run the Windows OS agent as a non-root user, some functions are unavailable in the following attribute groups, if they are owned solely by the administrator account:

- Registry
- File Trend
- File Change

Remote deployment of other agents is not available because administrator rights are required to install the new agents.

For Agent Management Services, the watchdog cannot stop or start any agent that it does not have privileges to stop or start.

To create a non-root user, create a new Limited (non-root) user and set up registry permissions for the new user as in the following example:

- Full access to `HKEY_LOCAL_MACHINE\SOFTWARE\Candle`
- Read access to `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib`

The user that starts the Monitoring Agent for Windows OS – Primary service must have rights to manage the Monitoring Agent for Windows OS - Watchdog service. The user that starts the Monitoring Agent for Windows OS - Watchdog service must also have rights to manage any services that are managed by the Agent Management Services, including the Monitoring Agent for Windows OS – Primary service. To grant users the authority to manage system services in Windows, use security templates, group policy, or edit the `Subinacl.exe` file. For more information, see the following Microsoft documentation: http://support.microsoft.com/kb/325349 (http://support.microsoft.com/kb/325349).

The following example shows how to grant users the authority to manage system services by using security templates:

1. Click **Start** > **Run**, enter `mmc` in the Open box, and then click **OK**.
2. On the **File** menu, click **Add/Remove Snap-in**.
3. Click **Add** > **Security Configuration and Analysis**, and then click **Add** again.
4. Click **Close** and then click **OK**.
5. In the console tree, right-click **Security Configuration and Analysis**, and then click **Open Database**.
6. Specify a name and location for the database, and then click **Open**.
7. In the **Import Template** dialog box that is displayed, click the security template that you want to import, and then click **Open**.
8. In the console tree, right-click **Security Configuration and Analysis**, and then click **Analyze Computer Now**.
9. In the **Perform Analysis** dialog box that is displayed, accept the default path for the log file that is displayed in the Error log file path box. Otherwise, specify the location that you want. Click **OK**.
10. After the analysis is complete, configure the service permissions as follows:
    a. In the console tree, click **System Services**.

b. In the right pane, double-click the Monitoring Agent for Windows OS - Primary service.

c. Select the **Define this policy in the database** check box, and then click **Edit Security**.

d. To configure permissions for a new user or group, click **Add**.

e. In the **Select Users, Computers, or Groups** dialog box, type the name of the user or group that you want to set permissions for, and then click **OK**. In the **Permissions for User or Group** list, select the **Allow** check box (next to **Start**). Stop and pause permission is selected by default, so that the user or group can start, stop, or pause the service.

f. Click **OK** twice.

11. Repeat step 10 to configure the service permissions for the Monitoring Agent for Windows OS - Watchdog service.
12. To apply the new security settings to the local computer, right-click **Security Configuration and Analysis**, and then click **Configure Computer Now**.

Note: You can use also the Secedit command line tool to configure and analyze system security. For more information about Secedit, click **Start** > **Run**, enter cmd, and then click **OK**. At the command prompt, type `secedit /?`, and then press **ENTER**. When you use this method to apply settings, all the settings in the template are reapplied. This method might override other previously configured file, registry, or service permissions.

The following example shows how to set the Monitoring Agent for Windows OS and Watchdog services to log on as a non-root user by using the Windows Services console:

1. Click **Start** > **Run**, enter `services.msc`, and then click **OK**.
2. Select **Monitoring Agent for Windows OS - Primary**.
3. Right-click **Properties**.
4. Verify the startup type as being Automatic.
5. Select the **Log On** tab, and then select **Log on as "This account"** and supply the ID and password. Click **OK**.
6. Select **Monitoring Agent for Windows OS - Watchdog**.
7. Right-click **Properties**.
8. Verify the startup type as being Manual.
9. Select the **Log On** tab, and then select **Log on as "This account"** and supply the ID and password. Click **OK**.

Configuring OS agent log file monitoring

The Monitoring Agent for Linux OS, Monitoring Agent for UNIX OS, and Monitoring Agent for Windows OS agents are configured automatically. However, you can configure log file monitoring for the OS agents so that you can monitor application log files.

After the agents filter the log data, the data is sent in the form of a log event to the Cloud APM console.

Adding or removing log file monitoring configuration for the OS agents

You add log file monitoring configuration for the OS agents so the OS agents can filter log file data. Then, subsequently, you can also remove the log file monitoring configuration for the OS agents, if necessary.

Before you begin

The OS agents now include a sample regex1.conf file and a regex1.fmt file that you can view before you configure .conf and .fmt files. The files are located here:

- On UNIX/LINUX: `<install_dir>/samples/logfile-monitoring`
- On windows: `<install_dir\samples\logfile-monitoring`

Use a text editor to create a configuration `.conf` file and a format `.fmt` file. For more information about the content of these files, see "Configuration file" on page 631 and "Format file " on page 639. You must ensure that you save these files on the system where you access the Performance Management console so that you can upload the files to the Cloud APM server.

About this task

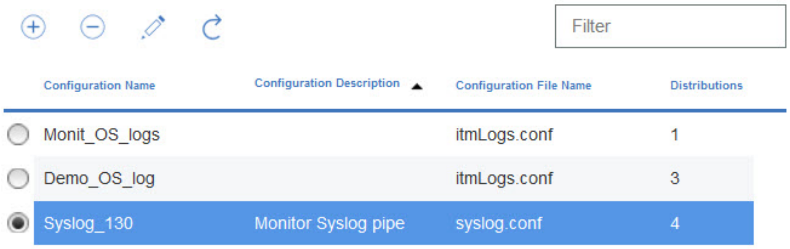
To enable the OS agents to monitor log files, you must upload the configuration file and format file and specify to which OS agent the configuration applies. The OS agent downloads the `.conf` and `.fmt` files and the agent monitors the log files that you specify in the configuration.

Procedure

Adding the log file monitoring configuration for the OS agents

1. Click **System configuration** > **Agent Configuration**.
2. Depending on the system on which you want to monitor the log files, click either the **Unix OS**, **Linux OS**, or **Windows OS** tab.
3. To create a new configuration, click the (+) icon to open the **New Log File Configuration** window. Enter a name for the configuration and a description of the configuration.
4. To view the contents of the `.conf` and the `.fmt` files, click **View**.
5. To upload the configuration by using the Cloud APM server, select the `.conf` file and the `.fmt` file from the same system where you open the Performance Management console and click **Done**.
6. On the OS agent tab, select the configuration that you uploaded.

   Important: The `.conf` and `.fmt` files that are distributed to the agents are renamed to the configuration name that you define.



	Configuration Name	Configuration Description ▲	Configuration File Name	Distributions
○	Monit_OS_logs		itmLogs.conf	1
○	Demo_OS_log		itmLogs.conf	3
●	Syslog_130	Monitor Syslog pipe	syslog.conf	4

7. To deploy the configuration, in the **Log Configuration Distributions List** table, select the agents to which you want to deploy the configuration and click **Apply Changes**.

Removing the log file monitoring configuration for the OS agents

8. Select the configuration name.
9. Clear the manage systems, and click **Apply Changes**.

   Important:

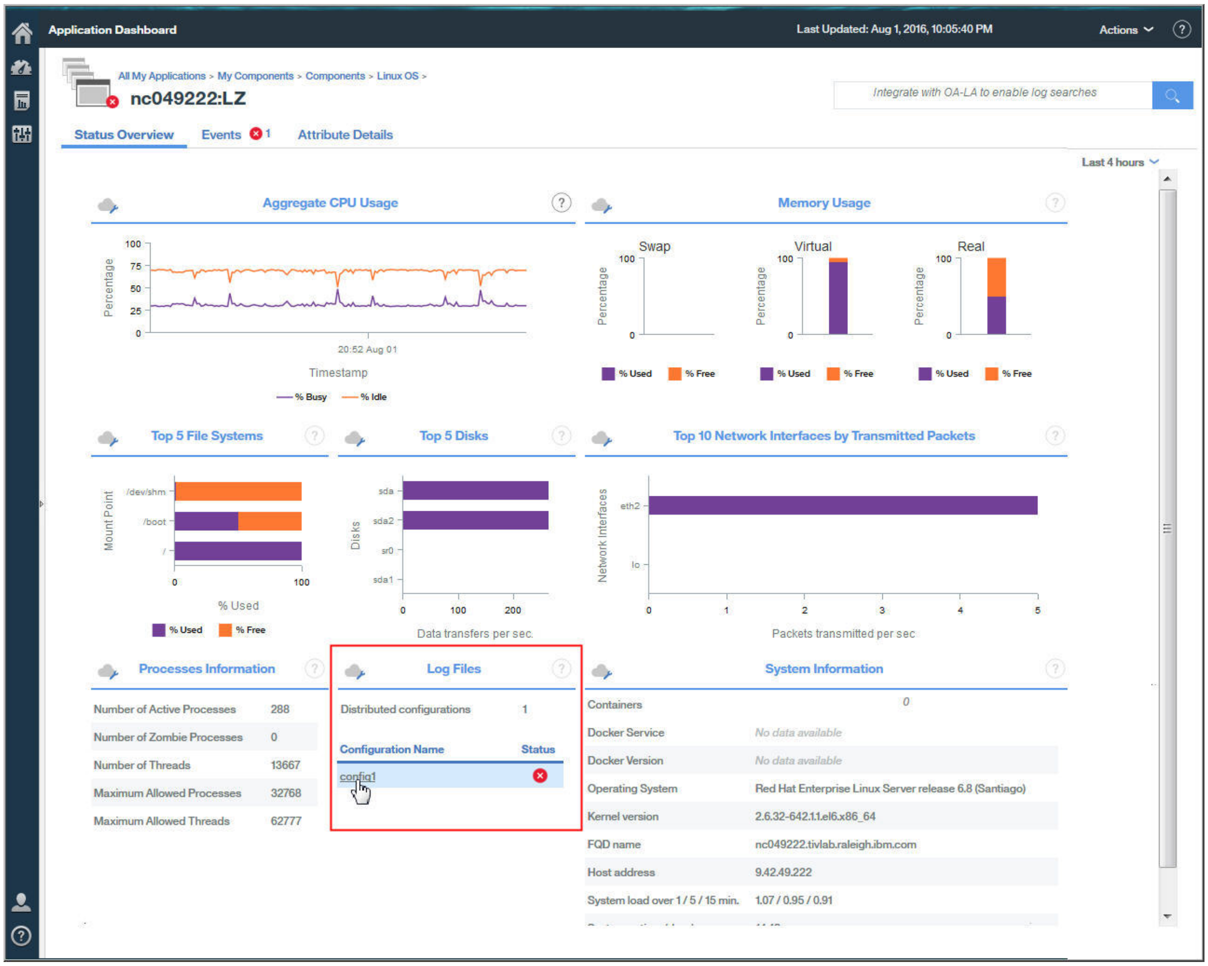
   After you remove the log monitoring configuration, the log file monitoring resource remains and it stays online until you restart the OS agent. The offline log file monitoring resources are cleared after the time that is specified in the **Remove Offline System Delay** option.

Viewing log file monitoring content

You can view the log file monitoring configuration for the OS agents that you deployed to monitor log files.

Procedure

1. Click **Performance** > **Application Performance Dashboard**, and select an application that includes the OS agent where you deployed the log file monitoring configuration.
2. Drill down to the OS agent dashboard, and in the Log Files widget, click the profile to view the log monitoring configurations that are distributed and the monitored logs.



Number of Active Processes	288
Number of Zombie Processes	0
Number of Threads	13667
Maximum Allowed Processes	32768
Maximum Allowed Threads	62777

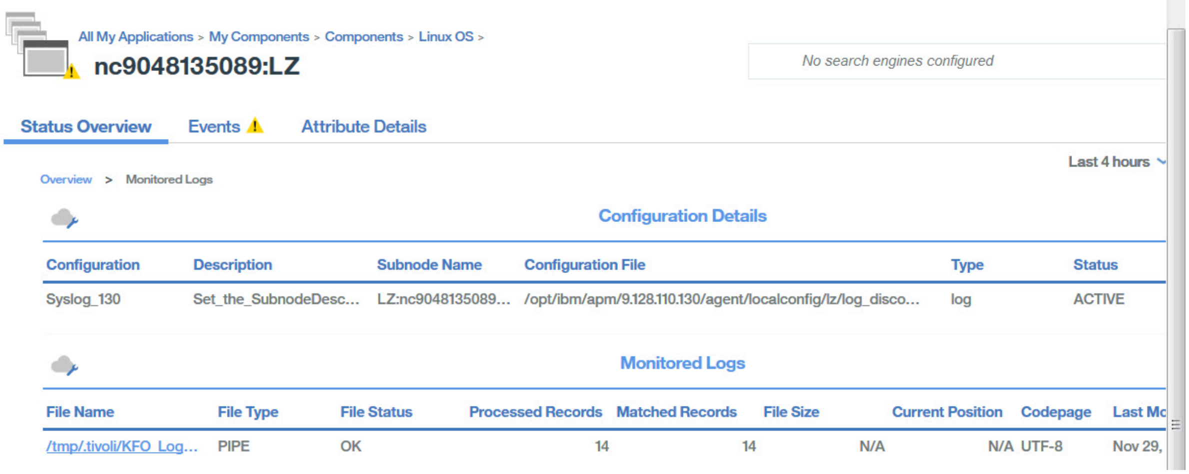
Distributed configurations 1

Configuration Name	Status
config1	✕

Containers	0
Docker Service	No data available
Docker Version	No data available
Operating System	Red Hat Enterprise Linux Server release 6.8 (Santiago)
Kernel version	2.6.32-642.1.1.el6.x86_64
FQD name	nc049222.tivlab.raleigh.ibm.com
Host address	9.42.49.222
System load over 1 / 5 / 15 min.	1.07 / 0.95 / 0.91

The configuration details include the configuration name, description, subnode, configuration file, status, and error code.

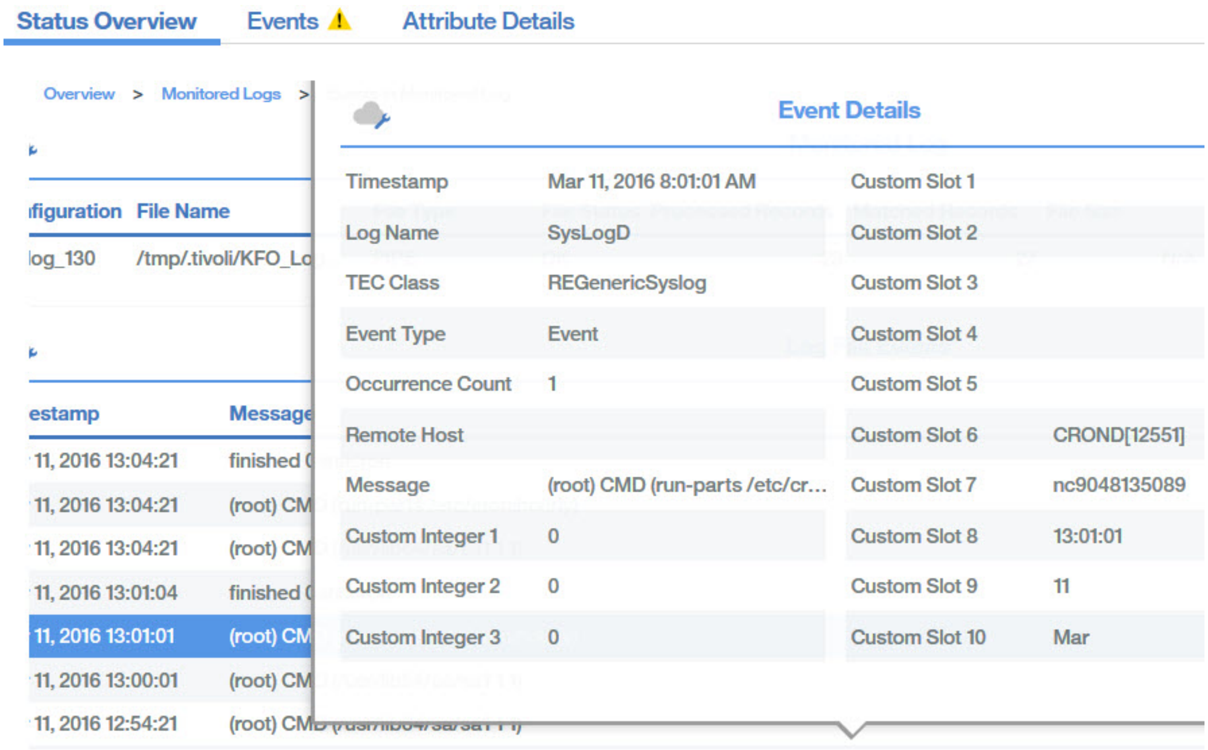
3. Click the log file name to view all the log file events that are associated with the log file.


All My Applications > My Components > Components > Linux OS >
nc9048135089:LZ
No search engines configured
Status Overview
Events
Attribute Details
Last 4 hours
Overview > Monitored Logs
Configuration Details
Monitored Logs

Configuration	Description	Subnode Name	Configuration File	Type	Status
Syslog_130	Set_the_SubnodeDesc...	LZ:nc9048135089...	/opt/ibm/apm/9.128.110.130/agent/localconfig/lz/log_disco...	log	ACTIVE

File Name	File Type	File Status	Processed Records	Matched Records	File Size	Current Position	Codepage	Last Mc
/tmp/.tivoli/KFO_Log...	PIPE	OK	14	14	N/A	N/A	UTF-8	Nov 29,

4. Click the event to view the event details, for example, all the fields that you defined in the Format file.


Status Overview
Events
Attribute Details
Overview > Monitored Logs >
ıfiguration File Name
log_130 /tmp/.tivoli/KFO_Lo
estamp Message
11, 2016 13:04:21 finished
11, 2016 13:04:21 (root) CM
11, 2016 13:04:21 (root) CM
11, 2016 13:01:04 finished
11, 2016 13:01:01 (root) CM
11, 2016 13:00:01 (root) CM
11, 2016 12:54:21 (root) CMD (/usr/lib64/sa/sa1 1 1)

Event Details

Timestamp	Mar 11, 2016 8:01:01 AM	Custom Slot 1	
Log Name	SysLogD	Custom Slot 2	
TEC Class	REGenericSyslog	Custom Slot 3	
Event Type	Event	Custom Slot 4	
Occurrence Count	1	Custom Slot 5	
Remote Host		Custom Slot 6	CROND[12551]
Message	(root) CMD (run-parts /etc/cr...	Custom Slot 7	nc9048135089
Custom Integer 1	0	Custom Slot 8	13:01:01
Custom Integer 2	0	Custom Slot 9	11
Custom Integer 3	0	Custom Slot 10	Mar

Displaying log file monitoring events

After you configure the OS agent to monitor you application log files, you can create thresholds to raise alarms on the log file conditions that you want to be alerted of.

Procedure

1. On the Navigation Bar, click **System Configuration** > **Threshold Manager**.
2. Select the target OS for **Data Source type**.
3. Click ⊕Add to create a new threshold.
4. Set a severity for the event that exceeds this threshold.
5. Select the data set to create a threshold for. The following data sets are eligible for log file monitoring:
   - *Kpp* Log File RegEx Statistics
   - *Kpp* Log File Status
   - *Kpp* LogfileProfileEvents
6. Click ⊕**Add** to add a condition. In the **Add Condition** box, select an attribute and an operator, and then enter a threshold value.

   Repeat this step to add more conditions to your threshold if required.
7. In the Group assignment section, select the resource group that you want to assign your threshold to.
8. Click **Save**.
9. On the Navigation Bar, click **System Configuration** > **Advanced Configuration**.
10. In the **UI Integration** category, set the **Enable Subnode Events** valut to be True.
11. Click **Save**.

Results

When the specified condition becomes true, the log file event that triggers the alert is displayed in the Events tab.

Log file monitoring environment variables

You can set environment variables for log file monitoring in the OS agent environment files.

Set the following environment variables and replace K*PC* with the OS agent code where *PC* is the two character agent code, for example, klz is the code for the Linux OS agent.

K*PC*_FCP_LOG
: This variable is available in the `install_dir/config/.pc.environment` file. The default value is `True` and you use it to enable or disable the log monitoring feature.

K*PC*_FCP_LOG_PROCESS_MAX_CPU_PCT
: This setting is the maximum allowable percentage of all system CPU that the agent uses over a 1-minute interval. Valid values are 5 - 100. The default value is 100. This setting is associated with the CPU throttling feature. If you specify a value less than 5, the minimum value of 5 is used.

K*PC*_FCP_LOG_PROCESS_PRIORITY_CLASS
: This setting is the operating system scheduler priority for the process. A is lowest, C is the operating system default, and F is the highest priority. The setting is one of the following values: `A, B, C, D, E, F`. These values are superseded by any values that you specify in the `.conf` file.

K*PC*_FCP_LOG_SEND_EVENTS
: The default setting is `True` and it is used by the OS agent to send events to the Cloud APM server.

K*PC*_FCP_LOG_SEND_EIF_EVENTS
: The default setting is `True`. If this option is set to `Yes` the agent sends event data to the Cloud APM server or to any EIF receiver such as the OMNIbus EIF probe. If the option is set to `No`, the agent does not send the event data. The setting of this option is global and applies to all monitoring profiles.

Note: The EIF receiver consumes events, otherwise problems might occur when the agent cache fills.

K*PC*_FCP_TRUNCATE_HOSTNAME_IN_LONG_SUBNODE_MSN
: OS agents with log file event monitoring have a subnode limitation. To manage log file events, the subnode MSN has the following structure: `UX:`*`CITRAHOSTNAME_PROFILENAME`*. The maximum size limitation for the subnode name is 32 characters. If the built subnode MSN name is too long and it is more than 32 characters, it is truncated to 32 characters. This name corresponds to the substring that is taken from the Profile Name.

In the OS agent configuration file, use the following variables to manage the profile names that are too long:

- UNIX OS agent: KUX_FCP_TRUNCATE_HOSTNAME_IN_LONG_SUBNODE_MSN=true
- Linux OS agent: KLZ_FCP_TRUNCATE_HOSTNAME_IN_LONG_SUBNODE_MSN=true
- Windows OS agent: KNT_FCP_TRUNCATE_HOSTNAME_IN_LONG_SUBNODE_MSN=true

For example, if you have an agent that is called `aixhost_nc123456789A`, which is 20 characters in length, `CTIRAHOSTNAME=aixhost_nc123456789A` is 20 characters.

and you have two profiles that are called:

```
ProfileLong12A (14 characters)
ProfileLong12B (14 characters)
```

the following related subnode MSNs are expected:

```
UX:aixhost_nc123456789A_ProfileLong12A (38 characters)
UX:aixhost_nc123456789A_ProfileLong12B (38 characters)
```

However, the subnode MSNs are truncated to the 32 character limitation so the resulting names are the same for both:

```
UX:aixhost_nc123456789A_ProfileL
UX:aixhost_nc123456789A_ProfileL
```

To truncate CTIRAHOSTNAME instead of the Profile Name, set the *Kpc_FCP_TRUNCATE_HOSTNAME_IN_LONG_SUBNODE_MSN=true* variable.

For example, if *n* is the length of the Profile Name, such as 14, the substring for the MSN name that relates to *CTIRAHOSTNAME* is truncated to 32-n-3 characters, so the *CTIRAHOSTNAME* variable is: `aixhost_nc1234`. Then, the distinguished subnode MSNs are:

```
UX:aixhost_nc1234_ProfileLong12A
UX:aixhost_nc1234_ProfileLong12B
```

Configuration file

OS agents use a configuration file t that is read by the agent when it starts. The file contains configuration options and filters. You must create this configuration file and configure the agent instance to use it.

The configuration file is monitored for changes to its time stamp every 60 seconds thereafter. If the time stamp of the file changes, the agent reinitializes its configuration dynamically, without requiring a restart. For more information, see "Changing the agent configuration and format files" on page 643.

The `.conf` file for the OS agent accepts these options:

codepage

This parameter is the code page of the monitored file. Use this parameter in the configuration file when the code page of the monitored file is different from the code page of the system. Specify the code page of the monitored file, for example, ibm-5348_P100-1997, UTF-16, or UTF-8.

ConfigFilesAreUTF8=Y

This parameter specifies that the configuration file and format file are in UTF-8. Use this parameter if the encoding of the configuration files is UTF-8 and the system code page is not. The default is that the agent assumes the system encoding.

DupDetectionKeyAttributes

A comma-separated list of Cloud APM attributes that is used to determine which events are duplicates. If all the named attributes are the same in two events, then those two events are considered duplicates. This option applies only to events. For more information, see "Event filtering and summarization" on page 997.

Note:

1. The attribute names are case-sensitive, so you must enter the names exactly as described.
2. If you do not provide a list of attributes, the values default to `Class` and `Logname`.

ENFORCE_STRICT_TEC_COMPATIBILITY

This parameter refers to all white space characters in the log data to ensure that the characters are respected. For example, when you use a format such as `"%s %s"` to extract information from log messages, the OS agent matches not only a literal space but also any other white space characters that are present such as tabs and carriage returns.

When this parameter is not set, the default behavior of the OS agent when it matches a Tivoli Enterprise Console® style format string is to match as much of the input text as it can, while it processes the format from left to right.

For example, for the `%s:%s` format string and the `one:two:three` input string, the OS agent default assigns `one.two` to the first parameter (corresponding to the first `%s`) and it assigns `three` to the second parameter.

Note:

1. This parameter does not apply to format statements that use the regular expression syntax.
2. Setting this parameter has a performance impact. To give greater control over the behavior and performance of matching, avoid setting this parameter and use regular expressions instead.

EventSummaryInterval

Specifies the number of seconds during which the agent searches for duplicate events to suppress. Set this parameter to a positive integer. This option applies only to events. For more information, see "Event filtering and summarization" on page 997.

EventFloodThreshold

Specifies which events are sent when duplicate events are detected. Set this parameter to `send_none`, `send_all`, `send_first`, or a positive integer. This option applies only to events. For more information, see "Event filtering and summarization" on page 997.

EventMaxSize

Specifies in bytes, the maximum size of a generated event. If specified, this parameter is used in two places:

1. The parameter can be used by the agent to set the size of a buffer that is used to process events. If not set, this buffer defaults to a size of 16384 bytes. If the buffer is set too small, events are truncated and can be discarded.
2. The parameter can be used by the EIF sender to set the size of a buffer that is used to send events to an EIF receiver, such as the OMNIbus EIF probe. If not set, this buffer defaults to a size of 4096 bytes. If the buffer is set too small, events are discarded.

FileComparisonMode

Specifies which log files are monitored when more than one file matches a wildcard pattern. The following values are available:

CompareByAllMatches

This value is the default behavior. All files that match the wildcard pattern that is specified in `LogSources` are monitored.

CompareByLastUpdate

Of the files that match the wildcard pattern that is specified in `LogSources`, the file with the most recently updated time stamp is monitored.

CompareBySize

Of the two or more files that match the file name pattern criteria, the larger file is selected for monitoring. Do not use `CompareBySize` with multiple matching files that are being updated at the same time and increasing their file sizes. If the largest file is subject to frequent change, monitoring might continually restart at the beginning of the newly selected file. Instead, use `CompareBySize` for a set of matching files where only one is active and being updated at any specific time.

CompareByCreationTime

Of the files that match the wildcard pattern that is specified in `LogSources`, the file with the most recently created time stamp is monitored. This value has the following restrictions:

- The value is applicable only to Windows operating systems because UNIX and Linux operating systems do not store a true creation time for files.
- The value is not supported for remote files that you monitor by using the Secure Shell (SSH) File Transfer Protocol.

Tip: The `CompareByLastUpdate`, `CompareBySize`, and `CompareByCreationTime` values can all be used for rolling log files. `CompareByLastUpdate` is typically used for these files.

FQDomain

Specifies how and if the agent sets a domain name:

- If set to `yes`, the agent determines the system domain name.
- If set to no, the agent does not set a domain name. The `fqhostname` attribute is assigned a blank string.
- If set so that it does not contain a `yes` or no value, the domain name is accepted as the value and it is appended to the host name.

For more information, see "Format file " on page 639.

IncludeEIFEventAttr

The agent includes a large attribute that is called *EIFEvent*, which is a representation of the event that is sent through the Event Integration Facility if that feature is enabled. The information that is contained in the *EIFEvent* attribute can also be found in other attributes. Its large size made it problematic, thus it was disabled by default. Setting this value to y, reenables the EIFEvent attribute.

Note: Using this attribute might cause thresholds to fail if you have large events. A large event in this context is an event where the total number of bytes that is required to contain all values for all attributes and their names results in a string longer than 3600 bytes.

LognameIsBasename

When set to y, the value of the `Logname` attribute is the base name of the log file in which the event was found. This option applies only to Performance Management events. The path is removed. For example, `/data/logs/mylog.log` becomes `mylog.log`. If this value is set to n, then you get the full path. However, because the attribute is limited to 64 characters, setting it to n means that the name is truncated if it is longer. For this reason, the default value is y. To see the full path name in a longer attribute, you can specify it in the mappings section of a format in the `.fmt` file, for example, `filename FILENAME CustomSlot1`. The mapping completes the slot that is named `filename` with the full path of the file in which the event was found and maps it into `CustomSlot1`, which is 256 characters.

LogSources

Specifies the text log files to poll for messages. The complete path to each file must be specified, and file names must be separated by commas. Within each file name, you can also use an asterisk (*) to represent any sequence of characters, or a question mark (?) to represent any single character. For example, `mylog*` results in polling all log files whose names begin with `mylog`, whereas `mylog???` results in polling all log files whose names consist of `mylog` followed by exactly 3 characters. These wildcard characters are supported only within the file name; the path must be explicitly specified.

If you want to use regular expressions or pattern matching in the path, see the RegexLogSources description.

A log file source is not required to exist when the agent is started; the log file is polled when it is created.

NewFilePollInterval

Specifies the frequency, in seconds, that the agent checks for new files to monitor. For example, if a file name specified by the *LogSources* or *RegexLogSources* configuration file settings does not yet exist when the agent starts, it checks again for the existence of the files after this interval.

NumEventsToCatchUp

Specifies the event in the log that the agent starts with. This option provides some flexibility if the source that is being monitored is new or the agent is stopped for an extended time. The following values are valid:

Note: For text files, values `0` and `-1` apply. For Windows Event Log, values `0`, `-1`, and n apply.

0

Start with the next event in the logs. This value is the default.

-1

When set to `-1`, the agent saves its place in the file that is being monitored. It saves its place so that when the agent is stopped and later restarted, it can process any events that are written to the log while it was stopped. The agent otherwise ignores events that arrived while it was stopped and restarts from the end of the file. This setting does not apply to pipes, or syslog monitoring on UNIX and Linux systems.

n

Set to a positive integer. Starts with the *nth* event from the most current event in the logs; that is, start *n* events back from the most current event in the logs. If *n* is greater than the number of events that are available, all the events that are available are processed.

Note: You can use the n value only for Windows Event Log. The n value is ignored when `UseNewEventLogAPI` is set to *y*.

PollInterval

Specifies the frequency, in seconds, to poll each log file that is listed in the `LogSources` option for new messages. The default value is 5 seconds.

If you upgraded a Windows Event Log adapter from a previous release and you have a value that is set for PollingInterval in the Windows registry, you must specify the PollInterval option in the agent configuration file with the same value that is used in the Windows registry. This rule applies only if you are replacing a Tivoli Enterprise Console OS agent that had values in the registry.

ProcessPriorityClass

Specifies the process priority for the agent. You can adjust this value to improve system performance if the agent processes large volumes of events and is using too many processor resources. The possible values are:

- A - Very low priority
- B - Low priority
- C - Typical priority
- D - Above typical priority
- E - High priority
- F - Very high priority
- USE_CONF_FILE_VALUE - Use the value that is specified in the configuration file. This value is the default.

RegexLogSources

Specifies the text log files to poll for messages. It differs from the LogSources option in that regular expression meta characters can be used in the base name portion of the file name and in one subdirectory of the file name. This difference provides greater flexibility than the LogSources option in describing multiple files to monitor in multiple directories.

For example, specifying /var/log/mylog* for the LogSources statement is identical to using the dot (.) meta character followed by an asterisk (*) meta character to form /var/log/mylog.* in the RegexLogSources statement. This type of qualifier results in polling all log files in the /var/log directory whose base names begin with mylog and are followed by zero or more characters. A /var/log/mylog.+ qualifier results in polling all log files in the /var/log directory whose names begin with mylog and are followed by one or more characters.

Similar to LogSources, the complete path to each file must be specified and the file names must be separated by commas. However, the comma is also a valid character inside a regular expression. To distinguish between a comma that is used as part of a regular expression and one that is used to separate file names, commas that are used as part of a regular expression must be escaped with the backslash (\) character.

For example, if you want to search for logs that match either of the following regular expressions, /logs/.*\.log and /other/logs/[a-z]{0,3}\.log, you must escape the comma in the {0,3} clause of the second expression so the agent does not mistake it for the beginning of a new expression: RegexLogSources=/logs/.*\.log,/other/logs/[a-z]{0\,3}\.log

If meta characters are used in the path name, the meta characters can be used in only one subdirectory of the path. For example, you can specify /var/log/[0-9\.]*/mylog.* to have meta characters in one subdirectory. The [0-9\.]* results in matching any subdirectory of /var/log that consists solely of numbers and dots (.). The mylog.* results in matching any file names in those/var/log subdirectories that begin with mylog and are followed by zero or more characters.

Because some operating systems use the backslash (\) as a directory separator it can be confused with a regular expression escape meta character. Because of this confusion, forward slashes must always be used to indicate directories. For example, Windows files that are specified as C:\temp \mylog.* might mean the \t is a shorthand tab character. Therefore, always use forward slashes (/) for all operating systems directory separators. For example, C:/temp/mylog.* represents all files in the C:/temp directory that start with mylog.

If more than one subdirectory contains meta characters, a trace message is also issued. For example, c:/[0-9\.]*/temp.files/mylog.* has two subdirectories with meta characters. [0-9\.]* is the first subdirectory with meta characters and temp.files is the second subdirectory that used a dot (.) meta character. In this case, the agent assumes that the first subdirectory with the meta character is used and the subsequent directories with meta characters are ignored.

SubnodeName

A string value that can be used to override the default name that is assigned to a monitoring profile subnode. By default the subnode name that is assigned to a monitoring profile corresponds to the base name of the configuration file that is used for that profile. By using this setting, a different subnode name can be assigned.

SubnodeDescription

A string value that can be used to assign a value to the *Subnode Description* attribute of *LFAProfiles*.

UnmatchLog

Specifies a file to log discarded events that cannot be parsed into an event class by the agent. The discarded events can then be analyzed to determine whether modifications to the agent format file are required. Events that match a pattern that uses *DISCARD* do not appear in the unmatch log because they did match a pattern.

This option is used in a test environment to validate the filters in the format file. This option fills up your file system if you leave it on for extended periods.

Options for remote log file monitoring by using SSH

Other than **SshHostList**, which is a list, all options can have only one value, which is applied to all remote hosts that are specified in **SshHostList**.

Only text log files are supported. AIX error report, syslog, and Windows Event Log are not supported.

Tip: You can set up syslog to write its output to a text log file and then remotely monitor that text file with the OS agent.

SshAuthType

Must be set to either *PASSWORD* or *PUBLICKEY*. If set to *PASSWORD*, the value of **SshPassword** is treated as the password to be used for SSH authentication with all remote systems. If set to *PUBLICKEY*, the value of **SshPassword** is treated as the pass phrase that controls access to the private key file. If set to *PUBLICKEY*, **SshPrivKeyfile** and **SshPubKeyfile** must also be specified.

SshHostList

A comma-separated list of remote hosts to monitor. All log files that are specified in the **LogSources** or **RegexLogSources** statements are monitored on each host that is listed here. If *localhost* is one of the specified host names, the agent monitors the same set of files directly on the local system. When you specify *localhost*, SSH is not used to access the files on the local system; the log files are read directly.

SshPassword

When the value of **SshAuthType** is *PASSWORD*, this value is the account password of the user that is specified in **SshUserid**. You can supply the account password in clear text, or you can supply a password that is encrypted with the IBM Tivoli Monitoring CLI **itmpwdsnmp** command. For more information about how to encrypt a password by using the **itmpwdsnmp** command, see "Remote log file monitoring: Encrypting a password or pass phrase" on page 648.

When the value of **SshAuthType** is *PUBLICKEY*, this value is the pass phrase that decrypts the private key that is specified by the **SshPrivKeyfile** parameter. You can supply the pass phrase in clear text, or you can supply a pass phrase that is encrypted with the IBM Tivoli Monitoring CLI **itmpwdsnmp** command. For more information about how to encrypt a password by using the **itmpwdsnmp** command, see "Remote log file monitoring: Encrypting a password or pass phrase" on page 648.

Note: If the value of **SshAuthType** is *PUBLICKEY*, and you configured SSH not to require a pass phrase, **SshPassword** must be set to null. To set **SshPassword** to null, the entry in the configuration file is:

```
SshPassword=
```

SshPort

A TCP port to connect to for SSH. If not set, defaults to *22*.

SshPrivKeyfile

If **SshAuthType** is set to *PUBLICKEY*, this value must be the full path to the file that contains the private key of the user that is specified in **SshUserid**, and **SshPubKeyfile** must also be set. If **SshAuthType** is not set to *PUBLICKEY*, this value is not required and is ignored.

SshPubKeyfile

If **SshAuthType** is set to *PUBLICKEY*, this value must be the full path to the file that contains the public key of the user that is specified in **SshUserid**, and **SshPrivKeyfile** must also be set. If **SshAuthType** is not set to *PUBLICKEY*, this value is not required and is ignored.

SshUserid

The user name on the remote systems, which the agent uses for SSH authentication.

Option that is supported on UNIX and Linux systems only

Linux AIX

AutoInitSyslog

If this option is set to `Yes`, the agent automatically configures the syslog facility to write a standard set of events to a pipe that the agent monitors. By enabling this setting, you can monitor syslog events without maintaining and rolling over log files. If this option is not set in the configuration file, it is the same as being set to `No`.

Restriction: This option is not supported for remote log file monitoring.

Options that are supported on Windows systems only

Windows

NTEventLogMaxReadBytes

If you are using the older NT Event Log interface (`UseNewEventLogAPI` is not set to `y`) to read event log data on a Windows system, the agent reads up to this number of bytes each time it checks the event log for new data. Setting the value to `0` causes the agent to attempt to read all new data, as it did in earlier releases. This activity can occupy the agent for a considerable amount of time on a system with many events. The default value is `655360`. When set, the agent might not stop at exactly the value that is specified, but rather at the nearest multiple of an internal buffer size to this value.

PreFilter

Specifies how events in a Windows Event Log are filtered before agent processing. `PreFilter` statements are used by `PreFilterMode` when the filters determine which events are sent from an event log to the agent. An event matches a `PreFilter` statement when each *attribute=value* specification in the `PreFilter` statement matches an event in the event log. A PreFilter statement must contain at least the log specification and can contain up to three more specifications, which are all optional: event ID, event type, and event source. The order of the attributes in the statement does not matter.

The `PreFilter` statement has the following basic format:

```
PreFilter:Log=log_name;EventId=value; EventType=value;Source=value;
```

You can specify multiple values for each attribute by separating each value with a comma.

Each `PreFilter` statement must be on a single line.

`PreFilter` is not mandatory. All Windows log events are sent to the agent if prefilters are not specified and `PreFilterMode=OUT`.

PreFilterMode

This option applies only to Windows Event Log. The option specifies whether Windows systems log events that match a `PreFilter` statement are sent (`PreFilterMode=IN`) or ignored (`PreFilterMode=OUT`). Valid values are `IN`, `in`, `OUT`, or `out`. The default value is `OUT`.

PreFilterMode is optional; if PreFilterMode is not specified, only events that do not match any PreFilter statements are sent to the agent.

Note: If you set PreFilterMode=IN, you must also define the PreFilter statements.

SpaceReplacement

Set to TRUE by default for Windows Event Log (Windows Server 2008 only) but not for previous versions of Event Log. When SpaceReplacement is TRUE, any spaces in the security ID, subsource, Level, and keywords fields of the event log messages are replaced with underscores (_). When SpaceReplacement is FALSE, any spaces in the security ID, subsource, Level, and keywords fields of the event log messages remain unchanged. For more information about this option, see “Windows Event Log” on page 999.

UseNewEventLogAPI

When set to y on Windows systems, uses the new Windows Event Log interface for event logs. The option is supported only on Windows 2008 and later. The option is needed to access many of the new event logs that debuted in Windows 2008 and the applications that run on it. The option is ignored on earlier versions of Windows and on UNIX and Linux. For more information about this option, see “Windows Event Log” on page 999.

WINEVENTLOGS

Controls which Windows event logs are monitored.

The WINEVENTLOGS statement is a comma-delimited list with no spaces. For more information, see “Windows Event Log” on page 999.

Note: Any carriage returns, tabs, or new lines in Windows events are replaced by spaces.

Option that is supported on AIX systems only

AIX

AIXErrptCmd

An **errpt** (error report) command string that the agent runs can be supplied here. The command output is fed into the stream of log data that is being monitored.

For example, the following command causes the agent to search for the *mmddhhmmyy* string and replace it with the actual date and time on startup. Only the first occurrence of the string is replaced.

```
AIXErrptCmd=errpt -c -smmddhhmmyy
```

Although you can supply your own errpt command, you must use the -c (concurrent mode) option so that the command runs continuously. You cannot use the -t option or the following options that result in detailed output: -a, -A, or -g.

The data stream is the standard output from the errpt command, so regular expressions in the .fmt file must be written to match. For example, the data output might be:

```
IDENTIFIER TIMESTAMP  T C RESOURCE_NAME  DESCRIPTION
F7FA22C9   0723182911 I O SYSJ2      UNABLE TO ALLOCATE SPACE IN FILE SYSTEM
2B4F5CAB   1006152710 U U ffdc       UNDETERMINED ERROR
2B4F5CAB   1006152610 U U ffdc       UNDETERMINED ERROR
```

A sample format that picks up the data rows, but not the header, is:

```
REGEX GenericErrpt
^([A-F0-9]{8}) +([0-9]{10}) ([A-Z]) ([A-Z]) (\S+) +(.*)$
Identifier $1 CustomSlot1
Timestamp  $2 CustomSlot2
T          $3 CustomSlot3
C          $4 CustomSlot4
Resource   $5 CustomSlot5
msg        $6
END
```

For more information, see *Monitoring an AIX Binary Log* in the IBM Agent Builder User's Guide.

Options that apply only when events are being forwarded to EIF

Important: These options apply to EIF events sent directly to Operations Analytics - Log Analysis, OMNIbus, or any other generic EIF receiver. The options are not intended for use with the Cloud APM server.

BufferEvents

Specifies how event buffering is enabled. The possible values are:

- **YES** - Stores events in the file that is specified by the BufEvtPath option (This value is the default).
- **MEMORY_ONLY** - Buffers events in memory.
- **NO** - Does not store or buffer events.

BufEvtPath

Specifies the full path name of the agent cache file. If this path is not rectified the default is:

- AIX `/etc/Tivoli/tec/cache`
- Windows `\etc\Tivoli\tec\cache`

Note: If events are being forwarded to more than one server, a *BufEvtPath* value must be specified for each forwarding channel. An index number is appended to the *BufEvtPath* name for each additional entry. For example, use *BufEvtPath1* to indicate the path name of the agent cache file for forwarding to the first extra server. The value that is set in each *BufEvtPath* must be unique.

BufEvtMaxSize

Specifies the maximum size, in KB, of the agent cache file. The default value is 64. The cache file stores events on disk when the *BufferEvents* option is set to Yes. The minimum size for the file is 8 KB. File sizes specified less than this level are ignored, and 8 KB is used. The value that you specify for the maximum file size does not have an upper limit.

Note: If the cache file exists, you must delete the file for option changes to take effect.

NO_UTF8_CONVERSION

Specifies whether the Event Integration Facility encodes event data in UTF-8. When this option is set to YES, the EIF does not encode event data in UTF-8. The data is assumed to already be in UTF-8 encoding when passed to the EIF. However, a prefix is added to the flag to indicate that the data is in UTF-8 encoding (if the flag does not exist at the beginning of the event data). The default value is `NO`.

MaxEventQueueDepth

This value indicates the maximum number of events that can be queued for forwarding. When the limit is reached, each new event that is placed in the queue bumps the oldest event from the queue. If not specified, the default value is 1000. This setting applies to all forwarding channels if *NumAdditionalServers* is used.

NumAdditionalServers

This entry is required if you want to forward events to more than one Netcool/OMNIbus ObjectServer. Its value is used to indicate the number of servers that events are forwarded to. Valid values are 1 - 8.

ServerLocation

Specifies the name of the host on which the event server is installed. Specify host name or IP address. Use the dotted format for IP address. You can specify failover values such as `ServerLocation1=2.3.4.5,2.3.4.6.` for the server locations if you want to. If you specify failover values for *ServerLocation*, you must also specify an extra *ServerPort* value for each *ServerLocation*.

Note: If events are being forwarded to more than one server, a *ServerLocation* value must be specified for each server. An index number is appended to the *ServerLocation* name for each additional entry. For example, use *ServerLocation1* to specify the name of the host on which the first extra server is installed.

ServerPort

Specifies the port number on which the EIF receiver listens for events. The *ServerPort* option can contain up to eight values, which are separated by commas. If failover values are specified for

ServerLocation, you must set an equivalent *ServerPort* value. The ServerPort is not used when the *TransportList* option is specified.

Note: If events are being forwarded to more than one server, a *ServerPort* value must be specified for each server. An index number is appended to the *ServerPort* name for each additional entry. For example, use *ServerPort1* to specify the port number on which the EIF receiver listens for events for the first extra server.

TransportList
Specifies the user-supplied names of the transport mechanisms, which are separated by commas. When a transport mechanism fails for sender applications, the API uses the following transport mechanisms in the order that is specified in the list. For receiving applications, the API creates and uses all the transport mechanisms. The transport type and channel for each *type_name* must be specified by using the Type and Channels keywords:

***type_name*Type**

Specifies the transport type for the transport mechanism that is specified by the *TransportList* option. SOCKET is the only supported transport type.

The server and port for each channel_name are specified by the *ServerLocation* and *ServerPort* options.

***type_name*Channels**

***channel_name*Port**
Specifies the port number on which the transport mechanisms server listens for the specified channel (set by the *Channel* option). When this keyword is set to zero, the portmapper is used. This keyword is required.

***channel_name*PortMapper**
Enables the portmapper for the specified channel.

***channel_name*PortMapperName**
Specifies the name of the portmapper if the portmapper is enabled.

***channel_name*PortMapperNumber**
Specifies the ID that is registered by the remote procedure call.

***channel_name*PortMapperVersion**
Specifies the version of the portmapper if the portmapper is enabled.

***channel_name*ServerLocation**
Specifies the name of the event server and the region where the server for transport mechanisms is located for the specified channel. The channel is set by the *Channel* option. This keyword is required.

The configuration file accepts generic EIF options when used directly with OMNIbus. These options operate only over an EIF connection to OMNIbus. They do not affect events that are sent to the Cloud APM server. For more information about these EIF options, see EIF keywords.

Format file

OS agents extract information from system log messages and then match different log messages to event classes. A format file serves as a lookup file for matching log messages to event classes, which tells the event class what to read, what to match, and how to format the data.

When the format file is used as a lookup file, all format specifications in the file are compared from the beginning to the end of the file. When two classes match or when a message has multiple matching classes, the first expression from the end that matches is used. If no match is found, the event is discarded. A discarded event is written to the unmatch log if it is defined in the `.conf` file.

The regular expression syntax that you use to create patterns to match log messages and events is described. Regular expression-filtering support is provided by using the International Components for Unicode (ICU) libraries to check whether an attribute value that is examined matches the specified pattern.

For more information about using regular expressions, see Regular Expressions in the *ICU User Guide*.

Format file specifications

The format file describes the patterns that the agent looks for to match events in the monitored logs. The format file consists of one or more format specifications.

You can change the format file while an agent instance is running. The file is read by the agent when it starts, and is monitored for changes to its time stamp every 60 seconds thereafter. If the time stamp of the file changes, the agent reinitializes its configuration dynamically, without requiring a restart. For more information, see "Changing the agent configuration and format files" on page 643.

To create new patterns to match an event, use the new regular expression syntax that consists of the following parts:

- Format header
- Regular expression
- Slot mappings
- End statement

The format header contains the **REGEX** keyword, which informs the agent that you are using a regular expression to match the pattern in the monitored log.

You assign this regular expression to an event class as shown in the following example:

```
REGEX REExample
```

If you use the special predefined event class `*DISCARD*` as your event class, any log records matching the associated pattern are discarded, and no events are generated for them. For example:

```
REGEX *DISCARD*
```

When a pattern is matched, nothing is written to the unmatch log. The log file status records that are matched include these discarded events.

Note: You can assign multiple event definitions to either the same event class or to different event classes. The class name is arbitrary and you can use it to indicate the type of event or to group events in various ways.

After the format header, the format content consists of a regular expression on the first line, followed by mappings. Each mapping is shown on a separate line and these mappings are described in the following example.

All lines that match the regular expressions are selected and sent to the monitoring server as events. The regular expression contains subexpressions. You can use the subexpressions to match specific parts of these lines that are the same to a variable called a *slot* in the Event Integration Facility.

The following monitoring log contains three lines that you might want to monitor:

```
Error:  disk failure
Error: out of memory
WARNING: incorrect login
```

For example, you generate an event for a specific error, such as the lines that begin with `Error` and ignore the line that begins with `Warning`. The regular expression must match the lines that begin with `Error` and also include a subexpression. The subexpression is denoted by parentheses and it must match only the input text that you want to assign to the *msg* slot. The following format definition is a simple regular expression with only one subexpression:

```
REGEX REExample
Error: (.*)
msg $1
END
```

Based on this format specification, and the preceding set of log data, the agent generates two events. Both events are assigned the `REEXample` event class. In the first event, the `disk failure` value is assigned to the *msg* slot. Also, in the second event, the out of memory value is assigned to the *msg* slot. Because the `Warning` line did not match the regular expression, it is ignored and no event is generated.

When you assign the value of $1 to the *msg* slot, you assign it the value of the first subexpression.

If you have log text that contains the following errors, you might want to assign these error messages to their own event class so that you are informed immediately of a disk failure:

```
Error: disk failure on device /dev/sd0: bad sector
Error: disk failure on device /dev/sd1: temperature out of range
```

You can include a description of the disk on which the error occurred, and more specifically the disk error in the event.

The following regular expression contains two subexpressions that identify this information:

```
REGEX DiskFailure
Error: disk failure on device (/dev/sd[0-9]):(.*)
device $1 CustomSlot1
msg $2
END
```

You assign these two subexpressions to event slots. The two events that are generated contain the following values:

```
"device=/dev/sd0" and "msg=bad sector"
"device=/dev/sd1" and "msg=temperature out of range"
```

If you use EIF to generate the first event, it displays as shown in the following example:

```
DiskError;device='/dev/sd0';msg='bad sector';END
```

If the event is sent to the Cloud APM server, the slot that is named *msg* is assigned to the Performance Management agent attribute with the same name. But the *device* slot has no predefined attribute.

If you need to see the value that is assigned to *device* directly on the Cloud APM console, or write thresholds against it, you must assign it to a Performance Management attribute.

The OS agent includes the following 13 predefined attributes:

- Ten string type attributes that range from *CustomSlot1* to *CustomSlot10*
- Three integer type attributes that range from *CustomInteger1* to *CustomInteger3*

Using these attribute names in the format file populates Performance Management attributes with the same name. Using these attributes does not affect the content of the EIF event sent directly to OMNIbus.

Note: The `CustomSlot` and `CustomInteger` attribute names are case-sensitive, so you must enter the names exactly as shown.

You assign a slot from the event definition to one of these custom Performance Management attributes in the format file.

You assign the *device* slot to the Performance Management string type attribute called *CustomSlot1* as shown in the following example:

```
REGEX DiskFailure
Error: disk failure on device (/dev/sd[0-9]):(.*)
device $1 CustomSlot1
msg $2
END
```

When the event is displayed in the Application Performance Dashboard, the value that is assigned to the *device* slot is assigned to the Performance Management `CustomSlot1` attribute. You view this value in the Cloud APM console or use it to define thresholds. You can assign any slot in the event definition to any of the 10 custom agent attributes in the same manner, by using "`CustomSlot`*n*", where *n* is a number from 1 - 10, next to the slot definition.

In this example, the first subexpression is defined specifically as `(/dev/sd[0-9])`, but the second subexpression is defined generally as `(.*)`. In defining the regular expression as specifically as possible, you improve performance. Therefore, if you enter a search for an error on a device that does not match the specific error message that is defined here, the search procedure stops immediately when the error is not found. Time is not wasted looking for a match.

The *END* keyword completes the format specification. The format header, regular expression, and the *END* keyword must each begin on a new line, as shown in the following example:

```
REGEX REExample
Error:
msg $1
END <EOL>
<EOF>
```

Note: For the last format in the file, you must insert a new line after the `END` keyword as shown in the example. Otherwise, you get a parsing error.

CustomInteger1 to *CustomInteger3* are 64-bit custom integer attributes. You can use them in the same manner as the string type `CustomSlot` attributes. You can use these attributes to map individual slots, or subexpressions, from the log file to individual Cloud APM attributes. Because these attributes are numeric, you can use arithmetic comparisons on them, such as < and >, which is not possible with the string attributes.

Note: Although these values are evaluated as integers by the Cloud APM server, for EIF purposes and within the format file, they are still treated as strings. For example, to use an integer slot in a PRINTF statement, you still identify it with “%s”, not “%d”.

The following example illustrates the use of a custom integer attribute. Suppose that a periodic UNIX syslog message is received that reports the percentage of a file system that is free, such as the following hypothetical log record:

```
Oct 24 11:05:10 jimmy fschecker[2165]: Filesystem /usr is 97% full.
```

You can use the following statement in the format file to check for the percentage of the file system that is free:

```
REGEX FileSystemUsage
^([A-Z][a-z]{2}) ([ 0-9][0-9]) ([0-9]{2}:[0-9]{2}:[0-9]{2}) (.*?) (.*?):
Filesystem (.*?) is ([0-9]+)% full\.$
Month   $1 CustomSlot1
Date    $2 CustomSlot2
Time    $3 CustomSlot3
Host    $4 CustomSlot4
Service $5 CustomSlot5
Filesystem     $6 CustomSlot6
PctFull         $7  CustomInteger1
msg          PRINTF("%s: %s% full", Filesystem, PctFull)
END
```

Note: In the preceding statement, everything between the ^ and $ symbols on the second and third lines must be on a single line.

Because you might have other events that put values in *CustomInteger1*, you can avoid confusing the different event types by using the value of the *Class* attribute to limit its effect to the correct type of events. For example, the following threshold formula causes the threshold to fire only when an event of the *FileSystemUsage* event class has a value greater than or equal to 95 in *CustomInteger1*:

```
(  Class == 'FileSystemUsage' AND CustomInteger1 >= 95)
```

A different event can then use *CustomInteger1* for a different purpose and not trigger this threshold accidentally.

In summary, you can now write a threshold in Performance Management that uses arithmetic operators on the `CustomInteger` attributes, which is not possible with the `CustomSlots` attributes.

Note: If you map non-integer data to the CustomInteger attributes, the resulting value might be zero or some unexpected value.

Changing the agent configuration and format files

The OS agent reads its configuration (.conf) and format (.fmt) files when it starts, and monitors their time stamp every 60 seconds thereafter.

If the time stamp of the configuration or format file changes, the agent reinitializes its configuration dynamically, without requiring a restart. During reinitialization, monitoring is interrupted momentarily. When monitoring resumes, the agent must determine the position in the monitored logs from which to restart. As a result, the agent behaves in the same way as a full stop and restart.

Note: Agent reinitialization after a configuration or format file change resets information in the Log File RegEx Statistics, Log File Status, and Log File Event attribute groups.

By default, the agent starts monitoring from the end of the file, when the reinitialization completes. This starting position can cause events that occurred during the interruption of monitoring to be missed. To ensure that such events are picked up when monitoring resumes, use the NumEventsToCatchUp=-1 setting.

Setting NumEventsToCatchUp=-1 causes a position file to be maintained. The position file is updated each time that the agent reads the log file. The update saves the position of the agent in the log file, in case of an agent restart. Maintaining the position file has a small performance impact, so maintain this file only if required. For more information about NumEventsToCatchUp, see "Configuration file" on page 631.

Note: Some configuration values are not present in the configuration file and are set during initial configuration. If you change these values, you must restart the agent.

Inheritance

A format file uses inheritance to derive slot definitions from a previously defined format specification.

Use the FOLLOWS relationship to build specific format specifications from generic format specifications by using inheritance.

First, you define a base class and call it DiskFailure, for example, as shown here:

```
REGEX DiskFailure
Disk Failure on device (.*)
device $1 CustomSlot1
END
```

This regular expression matches the Disk Failure on device/dev/sd0 errors in the monitoring log so that the /dev/sd0 value is assigned to the *device* slot.

However, you can also see an extended version of this error message reported in the monitoring log.

For example, you might see a Disk Failure on device /dev/sd0, error code: 13 error message.

This error message is matched to a slot as shown in the following example:

```
REGEX DiskFailureError FOLLOWS DiskFailure
Disk Failure on device (.*), error code: ([0-9]*)
errcode $2 CustomSlot2
END
```

Now, the event includes the *device* slot and the *errcode* slot. Because the DiskFailure event class defined a slot for the device name already, you allow the subclass to inherit that slot, and this inheritance saves you from declaring it a second time. The slot is defined as $1 so the first subexpression in the regular expression is assigned to that slot.

However, the DiskFailureError class also defines a second subexpression. You can assign this subexpression to a new slot called errcode and define it as $2 to refer to the second subexpression in the regular expression. This type of assignment is shown in the previous example that displays the log text.

The event now contains the `device` slot that is assigned the `/dev/sd0` value and the `errcode` slot that is assigned a value of 13. CustomSlot1 is assigned the device, and CustomSlot2 is assigned the error code.

Performance Management custom attribute mappings are also inherited. For more information about Performance Management custom attribute mappings, see "Format file specifications" on page 640.

Multi-line

Use the multi-line syntax to match records that span more than one line to patterns in the log that you are monitoring.

Specify the `\n` new line character as part of the regular expression to indicate where the line breaks occur in the monitoring log. See this type of syntax in the following example:

```
REGEX REMultiLine
Line1:(.*)\nLine2(.*)
msg $1
second_msg $2
END
```

Note: Windows Specify a `\r\n` carriage return and new line combination.

If the following error messages are reported in the log text, the REMultiLine event is created:

```
Line1: An error occurred
Line2: The error was "disk error"
```

The `msg` slot is assigned the value of An `error occurred` and the `second_msg` slot is assigned the value of `The error was "disk error"`.

Mappings

The OS agent uses mappings to determine the event class for a system log message. The agent determines the event class by matching the message to a pattern in the format file.

The agent converts log messages to event class instances that contain attribute `name=value` pairs. The event is then sent to the event server.

The agent determines the event class for a system log message at the source. The agent determines the event class by matching a system log message to a pattern in the format file. After you use this matching procedure to determine a class, you must assign values to the attributes.

Attribute values come from various sources, such as:

- Default values that are provided by the agent
- Log text that matches specific subexpressions in regular expressions

A map statement is included in the format file and consists of the following syntax:

```
name    value CustomSlotn
```

Here, you specify any identifier to describe the name of a slot (also known as a variable, attribute, or value identifier). Then, you specify a value to assign to this slot by applying any of the values that are described in "Value specifiers" on page 645.

Use custom slots to view data in the Performance Management console and to define thresholds. When you create thresholds, all custom slot values are strings. Custom slots are also required for duplicate detection to work because you must identify the slots that are used to determine duplicates. For more information about filtering events, see "Event filtering and summarization" on page 997. `msg` is a special slot name, with its own attribute in the event table. You do not need to use a custom slot for the `msg`.

You can limit the scope of a slot so that it exists only within the format definition. When you define the slot, you precede the slot name with a dash, for example:

```
-name    value
```

Any slot that you define in this way is not included in the final event. However, you can reference the slot elsewhere in the format definition, specifically within a PRINTF statement. In the `REGenericSyslog` example that follows, the `service` slot is not included if you generate but you can reference it in the PRINTF statement. It retains the same value that was applied to the original slot when it was defined without the dash. By using this procedure, you can use temporary variables from the format definition that are not included in the final event. For example, you can define an event class, `REGenericSyslog`, to match generic UNIX syslog events in the following way:

```
REGEX REGenericSyslog
^([A-Z][a-z]{2}) ([ 0-9][0-9]) ([0-9]{2}:[0-9]{2}:[0-9]{2}) (.*?) (.*?): (.*)$
month $1
date $2
time $3
host $4
-service $5
msg $6
syslog_msg PRINTF("service %s reports %s", service, msg)
END
```

Value specifiers
The mappings in a format specification assign values to attributes.

The mapping part of a format specification consists of the following types of value specifiers:

- `$i`
- String constant
- PRINTF statement

`$i`

The `i` indicates the position of a subexpression in a format string. Each subexpression is numbered from 1 to the maximum number of subexpressions in the format string.

The value of a `$i` value specifier (also known as a variable, slot, or attribute) is the portion of the system log message that is matched by the corresponding subexpression.

In the following example, the log agent translates any log message from the UNIX syslog facility into a syslog event with values assigned to it:

```
REGEX REGenericSyslog
^([A-Z][a-z]{2}) ([ 0-9][0-9]) ([0-9]{2}:[0-9]{2}:[0-9]{2})
 (.*?) (.*?): (.*)$
month   $1
date    $2
time    $3
host    $4
service $5
msg     $6
END
```

Each subexpression numbered from $1 to $6 matches an item in parentheses in the regular expression.

Therefore, the following syslog event:

```
Apr  6 10:03:20 jimmy syslogd 1.4.1: restart.
```

is assigned the following values:

```
month=Apr
date=6
time=10:03:20
host=jimmy
service=syslogd 1.4.1
msg=restart.
```

For example, in the syslog event, the `10:03:20` value matches the third item in parentheses in the regular expression, so the value is assigned to the *$3* time value. Similarly, the `jimmy` value matches the fourth item in parentheses in the regular expression, so the value is assigned to the *$4* host value.

string constant

The string constant declares that the value of the attribute is the specified string. If the attribute value is a single constant without any spaces, you specify it without surrounding double quotation marks (" ") as shown in the following example:

```
severity WARNING
```

Otherwise, if there are spaces in the attribute value, double quotation marks must be used as shown in the following example:

```
component "Web Server"
```

PRINTF statement

The PRINTF statement creates more complex attribute values from other attribute values. The PRINTF statement consists of the keyword `PRINTF` followed by a printf() C-style format string and one or more attribute names.

The format string supports only the `%s` component specifier. The values of the attributes that are used in the PRINTF statement must be derived from either a *$i* value specification or a constant string value specification (you cannot derive them from another PRINTF statement).

Use the value of the argument attributes to compose a new constant string according to the format string. This new constant string becomes the value of the attribute.

Based on the previous example where you defined the `REGenericSyslog` base class, and the *service* and *msg* slots, you can define an attribute called *syslog_msg* by using the `PRINTF` keyword.

```
syslog_msg PRINTF("service %s reports %s", service, msg)
```

If the following log message is reported:

```
Apr  6 10:03:20 jimmy syslogd 1.4.1: restart.
```

a new constant string is composed that contains the attribute values from the format string:

```
syslog_msg="service syslogd 1.4.1 reports restart."
```

Keywords

In the format file, use keywords to assign values that expand at run time.

The following keywords expand at run time:

- DEFAULT
- FILENAME
- LABEL
- REGEX

DEFAULT

Use the `DEFAULT` keyword to assign a DEFAULT value to a specific slot or attribute. The OS agent assigns an internal default value to slots that are described in the following table:

Table 194. Slots and the DEFAULT value

Slots	Description
hostname	*hostname* is the short host name of the system where the agent is running. It does not include the domain name of the system.
origin	*origin* is the IP address of the system where the agent is running.
fqhostname	*fqhostname* is the fully qualified host name of the system where the agent is running. It includes the domain name of the system.
RemoteHost	When an event originates on the local system, this attribute is empty. If an event originates on a remote system, *RemoteHost* contains a string of the form *user@host:port*, which indicates the remote host name on which the event occurred, and the user and port on that host that are used to connect.

The value that is assigned to *fqhostname* is influenced by the following FQDomain (optional) settings in the `.conf` file:

- If you set FQDomain to `yes`, the agent determines the system domain name itself.
- If you do not set a value for FQDomain or if you set the value to `no`, the agent does not set a domain name, and the *fqhostname* attribute is assigned a blank string.
- If you set FQDomain so that it does not contain a `yes` or `no` value, the domain name is accepted as the value and it is appended to the host name.

In the following example, the format definition contains three attributes or slots:

- *hostname* DEFAULT
- *origin* DEFAULT
- *fqhostname* DEFAULT

If you set the `FQDomain` to `yes` in the `.conf` file and you run it on a computer with the following properties:

- *hostname*: `myhost`
- *IP address*: `192.168.1.100`
- *domainname*: `mycompany.com`

an event is created and the three slots are assigned the following values:

```
"hostname=myhost", "origin=192.168.1.100", "fqhostname=myhost.mycompany.com"
```

FILENAME

The `FILENAME` keyword indicates the fully qualified file name (including the path) of the log file that contains the message. If you use a single agent to monitor multiple log files and you need to identify the source of the event, use this keyword to populate an event attribute with the file name. If the message comes from the system log, mapping is set to `EventLog` for Windows OS agents and `SysLogD` for UNIX OS agents.

Note: The path includes an attribute for this keyword.

LABEL

The `LABEL` keyword specifies the host name of the system where the agent is running.

REGEX
The REGEX keyword expands to the regular expression that matched the message and caused the event.

Maximum message length
This value is the maximum message length that the OS agent can receive without truncating the message.

The maximum message length is different for Performance Management and Tivoli Netcool/OMNIbus.

Performance Management
For events sent to Performance Management, the `msg` attribute is limited to 2048 bytes. Messages that are greater in length are truncated.

Tivoli Netcool/OMNIbus
For events sent through the Probe for Tivoli EIF to Netcool/OMNIbus, the total size of the event, including the class name and all slots and their values cannot exceed 4096 bytes. For example, in the following sample EIF event, `;END` does not count against the 4096-byte limit. However, everything else does count against the limit, including the syntactic elements such as the semicolons, quotation marks, and equal signs.

```
Class;attr1=value1;attr2=value2;msg='Hello, world';END
```

Remote log file monitoring: Encrypting a password or pass phrase

For increased security, you can encrypt passwords and pass phrases that are transmitted to remote systems when you use Remote log file monitoring.

About this task

The encrypted password and pass phrases are stored in the configuration (`.conf`) file. For more information about the configuration file, see "Configuration file" on page 631.

Procedure

- Run the **itmpwdsnmp** command and supply the password or pass phrase that is to be encrypted:
  - Linux AIX The command is run from the Cloud APM installation directory. The default installation path is `opt/ibm/apm/agent` and *install_dir* is where you installed the agent.
  - Windows The default installation path is `C:\IBM\APM`.

  Linux Example of the command when it is run on a Linux system:

  ```
  $ export install_dir=/opt/ibm/apm/agent/bin
  $ /opt/ibm/apm/agent/bin

  Enter string to be encrypted:
  mypassword

  Confirm string:
  mypassword

  {AES256:keyfile:a}Z7BS23aupYqwlXb1Gh+weg==
  $
  ```

  In the example, the entire output from the `{AES256:keyfile:a}Z7BS23aupYqwlXb1Gh+weg==` command is used to set **SshPassword** in the agent configuration file. The `{AES256:keyfile:a}` prefix tells the agent that the password is encrypted.

  To encrypt a pass phrase for a private key file, follow the same procedure.

Configuring OS agent custom scripting

The Monitoring Agent for Linux OS, Monitoring Agent for UNIX OS, and Monitoring Agent for Windows OS agents are configured automatically. This feature allows users to define scripts to run at OS agents at a defined frequency.

The custom scripting feature is enabled by default. The administrator can enable/disable it by setting a new environment variable *KXX_FCP_SCRIPT*=true/false (the default is true) in the agent configuration file, where XX can be:

- LZ for Monitoring Agent for Linux OS

- UX for Monitoring Agent for UNIX OS

- NT for Monitoring Agent for Windows OS

The details are provided in the following sections.

Custom scripting Quick Start

Add custom scripting for the OS agents to define scripts to run at OS agents at a defined frequency.

The feature is enabled with default values as soon as the OS agent is started. The only action to start the scripting feature is:

Create a property file under default directory (on Linux™ or UNIX™ it is `install_dir/localconfig/`*`product code`*`/scripts_definitions`, on Windows™ it is `install_dir\localconfig\nt\scripts_definitions` by using as an example the provided template `script_property.txt`.

Only two properties are required:

ATTRIBUTE_NAME
Any name used to uniquely identify the script definition inside the property file.

SCRIPT_PATH_WITH_PARMS
The fully qualified path of the script with arguments.

Not only shell scripts but also perl and other types of scripts can be used. Specify the full command to run in the SCRIPT_PATH_WITH_PARMS property.

For example, `perl C:\IBM\scripts\Custom_Scripts\date.pl`. In this example, make sure that the location of `perl` can be resolved by the agent through the PATH variable in its environment. Specify the full path where `perl` is installed otherwise.

Parameters in OS agent environment files

You can set the parameters for custom scripting in the OS agent environment files.

It is possible to customize the scripting feature by setting parameters in the OS agent environment files:

***install dir*/config/lz.environment**
The environment file for the Monitoring Agent for Linux OS.

***install dir*/config/ux.environment**
The environment file for the Monitoring Agent for UNIX OS.

***install dir*\TMAITM6_x64\KNTENV**
The environment file for the 64 bit Monitoring Agent for Windows OS.

***install dir*\TMAITM6\KNTENV**
The environment file for the 32 bit Monitoring Agent for Windows OS.

KXX_FCP_SCRIPT
The scripting feature is enabled by default. To disable it set: `KXX_FCP_SCRIPT=false`

Other parameters can be defined inside the agent environment files based on specific needs:

KXX_FCP_SCRIPT_DEFINITIONS
The location where property files are stored.

The default location on Linux™ or UNIX™ is *install dir*/localconfig/*PC*/scripts_definitions, on Windows™ it is *install dir*\localconfig\nt\scripts_definitions

KXX_FCP_SCRIPT_INTERVAL

OS agent uses the value of this variable as loop interval in seconds to check execution of running scripts and it sends events if the filter condition is satisfied. The minimum value is 30 seconds and the maximum value is 300 seconds. Invalid values are reset to the default. The default value is 60 seconds.

Note: This parameter is ignored if KXX_FCP_SCRIPT_SYNC_INTERVALS is set to USE_SCRIPT (see definition for KXX_FCP_SCRIPT_SYNC_INTERVALS).

KXX_FCP_SCRIPT_SYNC_INTERVALS

If the agent looping interval defined by KXX_FCP_SCRIPT_INTERVAL is larger than the script execution frequency, it can happen that data produced by some of the script execution loops is lost. In order to avoid this behavior, the script execution frequency can be synchronized with the agent looping interval by setting the KXX_FCP_SCRIPT_SYNC_INTERVALS to:

- USE_AGENT - The value of each script execution frequency is forced to be the maximum between KXX_FCP_SCRIPT_INTERVAL and EXECUTION_FREQUENCY defined in its property file.
- USE_SCRIPT - The agent looping interval is dynamically set to the minimum frequency value (EXECUTION_FREQUENCY in property file) between all of the defined scripts. The value set by KXX_FCP_SCRIPT_INTERVAL is ignored. The frequency of the scripts remains as defined in the property files. When you set USE_SCRIPT, the agent looping interval can change every time a script definition is added, changed, or removed. In any case, it cannot be lower than the value set by KXX_FCP_OVERRIDE_MIN_FREQUENCY_LIMIT or bigger than 300 seconds.
- NO - No synchronization is done and some execution results might be lost.

KXX_FCP_SCRIPT_DEFINITIONS_CHECK_INTERVAL

At startup and at every interval that is defined by this variable, the OS agent checks for any changes in scripts or property files. Note if KXX_FCP_SCRIPT_DEFINITIONS_CHECK_INTERVAL is less than the agent looping interval it is reset to the agent looping interval. The maximum allowed value is the default, 300 seconds.

KXX_FCP_USER

This parameter is valid only on Linux™ or UNIX™ OS agents. It defines the user used to create the fcp_deamon process if different from OS agent process user; all the scripts are run by this user. Be advised the owner of the OS agent must have correct permission to create the fcp_daemon process. On Windows™, a different user must be defined as login of the service Monitoring Agent for Windows OS "FCProvider". The user must have "Full Control" permission to agent installation directory and scripts repository directories. For more information, see:

https://www.ibm.com/support/knowledgecenter/SSHLNR_8.1.4/com.ibm.pm.doc/install/install_linuxaix_agent_nonroot.html

KXX_FCP_MAX_CDP_DP_THREAD_POOL_SIZE

It defines the maximum concurrent number of scripts to be run. Maximum value is 32.

KXX_FCP_MAX_DAEMON_RESTARTS

The OS agent watches the fcp_daemon: If an abnormal exit of process occurs, the OS agent restarts it. The default value is 4. The restart is done for the KXX_FCP_MAX_DAEMON_RESTARTS (times at a day). The value 0 must be used to avoid the restart; if -1 is set, the OS agent retries to restart fcp_daemon forever. The restart counter is reset at OS agent restart.

KXX_FCP_SEND_SCRIPT_RUNTIME_EVENTS

The default value is true. If set to false, the OS agent stops sending events for each row of script standard output. In this case script outputs are visible on console workspaces but no situations are displayed and no historical data is collected.

KXX_FCP_OVERRIDE_MIN_FREQUENCY_LIMIT

It is used when KXX_FCP_SCRIPT_SYNC_INTERVALS is set to USE_SCRIPT. In this condition, it sets the minimum value of the OS agent looping interval.

Using low values for the OS agent looping interval (less than 5 seconds) is highly invasive and can impact OS agent performances. If frequent data collection is needed (for example, every second), it is suggested to customize a script. The script caches data at the needed frequency and returns the collected data to the OS agent at a higher interval (for example, every 60 seconds).

The following Agent Builder (`CDP`) variables can also be used to control the behavior of the fcp_daemon:

CDP_DP_REFRESH_INTERVAL
(default 60 sec) Global script scheduled start time. Used if the frequency is not passed in the script property file.

CDP_DP_SCRIPT_TIMEOUT
(default 30 sec) Global script execution maximum time. When the execution time of a script exceeds this limit, its Status_Code is set to TIMEOUT

CDP_DP_KILL_ORPHAN_SCRIPTS
(Y|N - default N) Global behavior used by fcp_daemon process for timing out scripts. When set to 'Y', the scripts are ended, otherwise they are abandoned. This value is ignored for a specific script if the `KILL_AFTER_TIMEOUT` key is set in the script property file

CDP_MAXIMUM_ROW_COUNT_FOR_CPCI_DATA_RESPONSES
(default 1000) Global value is added for performance reasons to limit the maximum number of output rows returned by the scripts. Extra rows after this limit are ignored. Allowed values are positive integer. Invalid values are changed to no limit.

The fcp_daemon also supports the other environment variables that are used to control Agent Builder agents. For a complete list, see the *IBM Agent Builder User's Guide* here: Chapter 2, “PDF documentation,” on page 45.

Parameters in property files

You can set the parameters for custom scripting in the property files.

The KXX_FCP_SCRIPT_DEFINITIONS directory contains a list of *.properties files. Each property file contains a list of scripts to run with respective properties in the form of `key=value`. The properties that can be defined (case-insensitive) are:

ATTRIBUTE_NAME

Required - string max 256 characters. A name of your choice that defines a specific script and its attributes. The characters that can be used for the `ATTRIBUTE NAME` name can be alphanumeric and only the underscore can be used as a special character. If other special characters (a space or blank) are used, they are converted to underscore (_). When multiple scripts are listed inside the same property file, more, different `ATTRIBUTE_NAME` must be defined (one for each script). It must be the first value that is specified for each defined script and delimits the start of the properties set for the specific script until the next `ATTRIBUTE_NAME`.

SCRIPT_PATH_WITH_PARMS
Required - string max 512 characters. This parameter defines the full path to the script with parameters, which are separated by a blank. No special characters can be used in the script path name. Values containing blanks must be enclosed in single (') or double quotation marks ("). Environment variables can be passed, but only enclosed in ${...} for all the operating systems. Environment variables must be available in the OS agent process context.

EXECUTION_FREQUENCY
Optional - default value is 60 seconds. This parameter defines the script execution frequency.

CUSTOM_NAME
Optional - string max 256 characters. This parameter can be used for a description of the script.

IS_ACTIVE

Optional - true|false The default value is true. It activates the script. If false, the script is not run.

DISABLE_USE_AGENT_SYNC

Optional - true|false The default value is false. If true, the EXECUTION_FREQUENCY of the script is respected also if the global variable KXX_FCP_SCRIPT_SYNC_INTERVALS is set to USE_AGENT.

KILL_AFTER_TIMEOUT

Optional - true|false The default value is defined by the CDP_DP_KILL_ORPHAN_SCRIPTS variable. When true, the script is ended after timeout. A timeout occurs when script execution is greater than the value specified by CDP_DP_SCRIPT_TIMEOUT parameter in OS agent configuration file. Otherwise, it is ignored. In both cases, no data is collected. Note when KILL_AFTER_TIMEOUT is set, only the script that is defined in property file is ended and not child processes (if any) created by the script. This feature is not supported by Solaris™ and Windows™ 32-bit OS agents and any timing out scripts are abandoned.

Output rows that are returned by a script are parsed.

The script returns a standard output (called as first token). When the script returns more values in the output row, they are added as more tokens. A maximum of five strings, five integers and five floats following a predefined syntax.

OUTPUT_TYPE

STRING|INTEGER|FLOAT - Optional - default value is string. It defines the type of the first token that is returned by each row of the script; OUTPUT_TYPE can be:

- STRING (default) - strings up to 2048 characters. When used, the "Standard_Output_String" attribute of KXX_Custom_Scripts_Rtm_Smp is completed by the first token.
- INTEGER - allows getting numeric values between -9223372036854775806 and 9223372036854775806. When used, the "Standard_Output_Integer" attribute of KXX_Custom_Scripts_Rtm_Smp is completed by the first token.
- FLOAT - allows getting numeric values between -92233720368547758.06 and 92233720368547758.06, with two decimal precisions. When used, the "Standard_Output_Float" attribute of KXX_Custom_Scripts_Rtm_Smp is completed by the first token.

TOKEN_TYPES

STRING|INTEGER|FLOAT - Optional - It defines the output type of more tokens after the first one. The user can define a maximum of five strings, five integers and 5 floats. It is a list of types that are separated by commas: <token_type>,<token_type>,... token_type can be empty or one from (case-insensitive):

- - STRING or S
- - INTEGER or I
- - FLOAT or F
- If *TOKEN_TYPES* is empty, the corresponding token is skipped.

  Examples of the same valid layouts:
- - TOKEN_TYPES=S,I,S,,,F,,F,F
- - TOKEN_TYPES=String,integer,S,,,Float,,f,FLOAT

TOKEN_LABELS

STRING - Optional - Maximum 16 characters each label. It defines the labels of the tokens that are defined in TOKEN_TYPES. This value is a list of token labels that are separated by commas, and must correspond to the tokens defined by TOKEN_TYPES. For example:

- TOKEN_TYPES=S,I,S,,,F,,F,F
- TOKEN_LABELS=Cpu Name,Cpu number,Description,,,value 1,,value 2,value 3
- TOKEN_LABELS is ignored if TOKEN_TYPES is not set.

TOKEN_SEPARATOR

Optional - default semicolon ";" It sets the string to be used as separator to split the output row in tokens. It is ignored if TOKEN_TYPES is not set. Empty value (blank) is accepted as separator and multiple consecutive blanks in output rows are considered as a single one.

The following two parameters allow filtering the rows output of a script. They are applied by the OS agent only to the first token and they must be used together:

FILTER_VALUE

Optional. The value used for comparison. It is required if FILTER_OPERATOR is defined. If the OUTPUT_TYPE is a string, the filter value must reflect exactly the string value that is returned by the script that is intended to be filtered, without any additional quotation marks (no wildcards allowed).

FILTER_OPERATOR

Optional. The operator used for the comparison. It is required if FILTER_VALUE is defined. Accepted FILTER_OPERATOR values include:

- = (equal to)
- != (not equal to)
- > (larger than) only for numeric type
- >= (not lower than) only for numeric type
- < (less than) only for numeric type
- <= (not bigger than) only for numeric type

Examples of property file

Examples of setting parameters in the property files.

#First script definition: script ex_script1.sh is started every 150 seconds. It returns float values and only the output rows equal to 0.5 are considered by the agent.

```
ATTRIBUTE_NAME=sample1
SCRIPT_PATH_WITH_PARMS=/opt/ibm/apm/agent/localconfig/lz/scripts_definitions/ex_script1.sh
EXECUTION_FREQUENCY=150
OUTPUT_TYPE=FLOAT
FILTER_VALUE=0.5
FILTER_OPERATOR==
```

#Second script definition: script ex_script2 is started every 60 seconds. It returns integer values and only the rows different from 0 are considered by the agent.

```
ATTRIBUTE_NAME=ex_script2
SCRIPT_PATH_WITH_PARMS=${CANDLE_HOME}/tmp/check_out.sh
EXECUTION_FREQUENCY=60
OUTPUT_TYPE=INTEGER
FILTER_VALUE=0
FILTER_OPERATOR=!=
```

#Third script definition: script ex_script3.sh is started every 120 seconds with three input parameters (the first input parameter is an integer, the second and third are string). It is ended if it hangs or if the execution time is greater than the timeout value.

```
ATTRIBUTE_NAME=ex_script3
SCRIPT_PATH_WITH_PARMS=/opt/scripts/ex_script3.sh 1 "second input parameter" "third input
parameter"
EXECUTION_FREQUENCY=120
OUTPUT_TYPE=STRING
KILL_AFTER_TIMEOUT=TRUE
```

#Fourth script definition: script cpu_mem_percentage.sh is started every 50 seconds and returns the cpuid as standard output string and two float values for Idle and Used CPU percentage and two integers for Memory and Virtual Memory usage. The pipe is used as separator to parse the output. An example of row that must be returned by the script is:

cpu2|35,5|65,5|3443|123800

```
ATTRIBUTE_NAME=cpu and mem Usage
SCRIPT_PATH_WITH_PARMS=${SCRIPT_HOME}/cpu_mem_percentage.sh
OUTPUT_TYPE=STRING
TOKEN_TYPES=F,F,I,I
TOKEN_LABELS= Idle CPU %, Used CPU %, Virt MEM used MB, MEM used MB
TOKEN_SEPARATOR=|
EXECUTION_FREQUENCY=50
```

Known problems and limitations

Known problems and limitations

- The Scripting Feature is not supported on Windows™ 2003 64-bit systems.
- Kill after timeout does not work on Solaris™ and Windows™ 32-bit OS agents.
- The fcp_daemon can stop running scripts in Windows™ 32 bit if some scripts do not complete within the timeout period and the user enabled intensive tracing. If fcp_daemon stops running scripts, the data reported on the console reflects the last time that the script was run. It is also possible that the OS agent stops returning data. Stopping the fcp_daemon process allows the agent to resume proper operation.
- SCRIPT_NONZERO_RETURN is returned instead of SCRIPT_NOT_FOUND or SCRIPT_LAUNCH_ERROR on Solaris™.
- The scripting feature does not provide full globalization; some issues can be found by using Nationalized characters in property files or script outputs.
- On Windows™ OS agent, there is no possibility of running scripts residing on a mapped network drive.
- When the Windows™ OS agent is upgraded, the scripting feature is not enabled by default. Edit the KNTENV and change `KNT_FCP_SCRIPT=FALSE` to `KNT_FCP_SCRIPT=TRUE`

Troubleshooting custom scripting

Troubleshooting custom scripting

Standard *KBB_RAS1* variable applies to the OS agent and to the fcp_daemon processes. To apply a specific trace setting to fcp_daemon only, use the *KXX_FCP_KBB_RAS1* variable; when *KXX_FCP_KBB_RAS1* is set, the value that is specified by *KBB_RAS1* is ignored by fcp_daemon.

To trace the operations logged by the OS agent core threads of the feature:

```
KBB_RAS1=ERROR (UNIT:factory ALL)
```

To trace scripting queries from the APM server and events sent to the server, add the entries:

On the Monitoring Agent for Linux OS

```
(UNIT:klz34 ALL) (UNIT:klz36 ALL)
```

On the Monitoring Agent for UNIX OS

```
(UNIT:kux48 ALL) (UNIT:kux50 ALL)
```

On the Monitoring Agent for Windows OS

```
(UNIT:knt84 ALL) (UNIT:knt86 ALL)
```

To view TEMA traces to verify private situation execution, add the entries:

```
(UNIT:kraavp all) (UNIT:kraapv all)
```

To see the execution of the scripts and how the data from the scripts is being parsed set:

```
KXX_FCP_KBB_RAS1=Error (UNIT:command ALL)
```

To troubleshoot problems in the communication between the os agent and fcp_daemon, add this trace level to both *KBB_RAS1* and *KXX_FCP_KBB_RAS1*:

```
(UNIT:cps_socket FLOW) (UNIT:cpci FLOW)
```

To see the interaction between the OS agent process and the fcp_daemon in detail add to both *KBB_RAS1* and *KXX_FCP_KBB_RAS1*:

```
(UNIT:cps_socket ALL) (UNIT:cpci ALL)
```

Quick Start Scenario
This section describes the minimum steps that are needed to configure custom scripting for an example scenario.

The following section describes the minimum steps that are needed to configure a Monitoring Agent for Linux OS to run two custom scripts.

Custom Scripts descriptions

In this example, the user has two scripts under a directory /scripts_repo:

checkDIRsize.sh – This script checks the size of a specified directory that is passed as input parameter. The output is an integer like: `4594740`

cpu_mem_usage.sh – This script checks the used CPU percentages and used Swap Memory megabytes. The output is returned in the form: `cpu1|96,5|23800`

Where the first token is the CPU ID, the second token is the used CPU percentage, the third token is the used swap memory in megabytes.

The customization needed to have the Monitoring Agent for Linux OS run these scripts.

The feature is enabled with default values as soon as the OS agent is started:

You create property files `AnyName.properties` under the default directory `install dir/localconfig/lz/scripts_definitions`. In this example, create two property files, one for each script named checkDIRsize.properties and cpu_mem_usage.properties:

```
#CheckDIRsize.properties
ATTRIBUTE_NAME=OPT_DIR_SIZE
SCRIPT_PATH_WITH_PARMS=/scripts_repo/checkDIRsize.sh /opt
EXECUTION_FREQUENCY=20
OUTPUT_TYPE=INTEGER
```

```
#cpu_mem_usage.properties
ATTRIBUTE_NAME=cpu_mem_usage
SCRIPT_PATH_WITH_PARMS=/scripts_repo/cpu_mem_percentage.sh
OUTPUT_TYPE=string
TOKEN_TYPES=F,I
TOKEN_LABELS= Used CPU %, Swap MEM used MB
TOKEN_SEPARATOR=|
EXECUTION_FREQUENCY=10
```

There is no need to restart the OS agent after you add (or change) the two property files. The OS agent checks script definition directory with a specified time interval (default value 300 seconds). Open the console and under the "Custom Scripts" workspace the scripts details and results are shown.

Configuring local environment variables

You can configure the local environment to change behavior of the Monitoring Agent for Linux OS.

About this task

Complete these steps to configure the Monitoring Agent for Linux OS.

Procedure

1. Create the `install_dir/config/lz.environment` file if the file doesn't exist.
2. Edit the `install_dir/config/lz.environment` file and enter values for the environment variables. For more information about the environment variables that you can configure, see Local environment variables.
3. Restart the Monitoring Agent for Linux OS.

   Note: Do not modify the `.lz.environment` file since it contains the default settings that are provided by IBM and changes to this file are not preserved when the agent is updated.

   Local environment variables

   KBB_SHOW_MTAB_FS
   This variable controls the Monitoring Agent for Linux OS data collection for file system. When `KBB_SHOW_MTAB_FS=false`, the agent monitors only the file systems that are listed in `/etc/fstab`. When `KBB_SHOW_MTAB_FS=true`, the agent monitors file systems that are listed in both `/etc/fstab` and `/etc/mtab`.

   Starting with APM 8.1.4.013 (LZ agent version 06.35.14.17), the default value is true.

   KBB_SHOW_CIFS
   Specifies whether CIFS monitoring is enabled. The default value is false. Possible values are 'true' and 'false'.

   KBB_SHOW_NFS
   Specifies whether NFS monitoring is enabled. The default value is false. Possible values are 'true' and 'false'.

   Note: Even if the value is set to 'true', the UI is configured to not displayed NFS file system information.

   KLZ_EXCLUDE_DOCKER_FS_STR
   Exclude Docker mounts that contain the string defined. This variable is a comma-separated list of substring part of a mount point to exclude. The default value is set as follows: `KLZ_EXCLUDE_DOCKER_FS_STR=/docker/,/var/lib/kubelet/,/var/lib/docker/,/pods/`

   Note: This variable works together with `KBB_SHOW_MTAB_FS=true`.

   KLZ_EXCLUDE_FSTYPE
   Exclude file system to monitor defined mount point. This variable is a comma-separated list of file system type. The names must match those names that are shown by the command "`mount -l`".

   Example: `KLZ_EXCLUDE_FSTYPE=mvfs,ext3`

   Note: In a conflict with `KBB_SHOW_NFS` or `KBB_SHOW_CIFS` settings, `KLZ_EXCLUDE_FSTYPE` takes priority.

Configuring Monitoring Agent for Windows OS file monitoring threshold

For any threshold created against the File Change or File Trend data sets, the following guidelines must be followed.

File Change attributes

Use File Change attributes to monitor changes to your file system and to request notification when resources change. File Change is a multiple-instance data set. You cannot mix these attributes with any other multiple-instance data set.

Note: When you define a threshold, the function REGEX is not supported in any attributes that contain the name or the path of the files.

When you create thresholds that use the File Change data set, you must supply values for the following attributes to restrict the monitoring for the threshold:

- Watch Directory (Unicode)
- Watch File (Unicode) optional

The use of this data set requires that you set one or more filter conditions. The following attributes can be used and the values for these filters are single characters, **y** or **n**, where y=yes, and n=no:

- Change File Name
- Change Directory Name
- Change Attributes
- Change Size
- Change Last Write
- Change Last Access
- Change Create
- Change Security
- Monitor all Conditions (activates all of the above)

Any query and threshold must contain only one row of filter data in the threshold formula. If more than one row is provided the results are undefined.

Note: The product provided File Change query does not produce data unless a threshold is defined against this data set.

Action The type of change that occurred most recently to the directory or to the file. Valid values are positive integers in the range 0 to 5 and can include the use of the *AVG, *MIN, *MAX, or *SUM functions. The numbers 1 through 5 determine the type of change that recently occurred.

1. File/directory was added
2. File/directory was removed
3. File/directory changed (in size, in attribute, or in the security for the file/directory)
4. File/directory was renamed - the old name displays
5. File/directory was renamed - the new name displays

For example, to determine whether a file or directory was removed, enter 2.

File Trend attributes

Use File Trend attributes to monitor the growth rate in file space usage, by both change and absolute size change, over various monitoring periods. File Trend is a multiple-instance data set. You cannot mix these attributes with those of any other multiple-instance data set.

Note: When you define a threshold, the function REGEX is not supported in any attributes that contain the name or the path of the files.

The File Trend attributes monitor the discrete files only and not subdirectories.

When you create a threshold that uses the File Trend data set, you must supply values for the following attributes to restrict the monitoring for the threshold:

- Watch Directory (Unicode)
- Watch File (Unicode) optional

Thresholds must contain only one row of filter data in the threshold formula. If more than one row is provided the results are undefined.

Note: The product provided File Trend query does not produce data unless a threshold is defined against this data set.

Configuring PHP monitoring

You must configure the Monitoring Agent for PHP so that the agent can collect data from the PHP application that is being monitored.

Before you begin

1. Ensure that you install the php-process package. If you use the `yum install` command to install PHP, run the `yum install php-process` command to install the php-process package.
2. Ensure that the Apache HTTPD server is started before you configure the agent.

   Open the Apache HTTP Server `httpd.conf` configuration file and ensure that both the `mod_status` and `ExtendedStatus On` options are enabled. For example:

   ```
   ExtendedStatus On
   <Location /server-status>
           SetHandler server-status
           Order deny,allow
           Allow from all
           Allow from 127.0.0.1
   </Location>
   ```

   In the given example, `http://127.0.0.1/server-status` must work fine for the agent to work properly.

   Note: You must have Lynx or Links installed on Linux for the agent to get monitoring data.

   Make sure that the command `apachectl status` works fine in the monitored Apache server with no code changes to the `apachectl` command. Lynx must be installed for the command `apachectl status` to work properly.

About this task

To avoid permission issues when you configure the agent, be sure to use the same root user or non-root user ID that was used for installing the agent. If you installed your agent as a selected user and want to configure the agent as a different user, see "Configuring agents as a non-root user" on page 183. If you installed and configured your agent as a selected user and want to start the agent as a different user, see "Starting agents as a non-root user" on page 992.

The PHP agent is a multiple instance agent; you must create the first instance and start the agent manually. The Managed System Name includes the instance name that you specify, for example, *`instance_name`*`:`*`host_name`*`:`*`pc`*, where *pc* is your two character product code. The Managed System Name is limited to 32 characters. The instance name that you specify is limited to 28 characters, minus the length of your host name. For example, if you specify PHP2 as your instance name, your managed system name is PHP2`:hostname:PJ`.

Important: If you specify a long instance name, the Managed System name is truncated and the agent code does not display correctly.

Procedure

- If your environment is the same as the default settings, you can use the default execution binary path, default `php.ini` file path, and default port to configure the agent:

  a) Enter:
  `install_dir``/bin/php-agent.sh config` *`instance_name install_dir`*`/samples/php_silent_config.txt`
  Where *instance_name* is the name you want to give to the instance, and *install_dir* is the PHP agent installation directory. The default installation directory is `/opt/ibm/apm/agent`.

  b) To start the agent, enter:
  `install_dir``/bin/php-agent.sh start` *`instance_name`*

- To configure the agent by editing the silent response file and running the script with no interaction, complete the following steps:

a) Open *install_dir*/samples/php_silent_config.txt in a text editor.

b) For **Location of PHP execution binary**, you can specify the directory where the PHP execution is located. The default location is /usr/local/bin.

c) For **Location of PHP INI file**, you can specify the directory where the php.ini file is located. The default location is /etc.

d) For **Web server port**, you can specify the port number of the web server that is running WordPress. The default is 80.

e) For **Application DocumentRoot**, you can specify the DocumentRoot of the PHP WordPress application. Use a colon to separate multiple records. To allow the agent to find all the records for you, use the default value of ALL.

f) Save and close the php_silent_config.txt file, then enter:
install_dir/bin/php-agent.sh config *instance_name* *install_dir*/samples/php_silent_config.txt
Where *instance_name* is the name that you want to give to the instance, and *install_dir* is the PHP agent installation directory. The default installation directory is /opt/ibm/apm/agent.

g) To start the agent, enter:
install_dir/bin/php-agent.sh start *instance_name*

- To configure the agent by running the script and responding to prompts, complete the following steps:

a) Enter:
install_dir/bin/php-agent.sh config *instance_name*
Where *instance_name* is the name you want to give to the instance, and *install_dir* is the PHP agent installation directory.

b) When prompted to Edit Monitoring Agent for PHP settings, enter 1 to continue.

c) When prompted for Location of PHP execution binary, press Enter to accept the default location or specify your own location.

d) When prompted for Location of PHP INI file, press Enter to accept the default location or specify your own location.

e) When prompted for Web server port, press Enter to accept the default port or specify a different port number.

f) When prompted for Application DocumentRoot, press Enter to accept the default or specify the DocumentRoot of the PHP WordPress application. You can use a colon to separate multiple records.

g) To start the agent, enter:
install_dir/bin/php-agent.sh start *instance_name*

Results

The agent evaluates only the performance of PHP requests in WordPress applications. CSS and JS loading are not evaluated. The agent does not use URL arguments to identify URLs.

What to do next

You can verify the PHP agent data is displayed in the Cloud APM console.

You must ensure that the WordPress plugin-in for the agent is activated. To ensure activation, complete the following steps:

1. In a web browser, enter the following URL http://*hostname*:*port*/wp-admin/.
2. Access the administrative page by navigating to **Plugins** > **Installed Plugins**.
3. Ensure that the PHP agent plug-in is activated. The PHP agent plug-in is listed as **WordPress Agent**. Typically, the plug-in is already activated. If it is not already activated, click on **Activate**.

Configuring PostgreSQL monitoring

You must configure the Monitoring Agent for PostgreSQL so that the agent can collect data from the PostgreSQL database that is being monitored.

Before you begin

You must install the PostgreSQL JDBC driver before you install this agent. The path to this driver is required at the time of agent configuration.

JDBC type 4 driver is the new version and hence preferable. User can install the subtype of JDBC 4 version according to the JDK version the agent uses. For mapping of JDBC version to JDK version get more information at https://jdbc.postgresql.org/download.html.

The `pg_hba.conf` file is the PostgreSQL database file that contains authentication settings. When the `auth-method` parameter value is set to `ident` in the `pg_hba.conf` file, the PostgreSQL agent cannot connect to the PostgreSQL database. Ensure that the authentication settings for the `auth-method` parameter are correct. For example, you can set these values for `auth-method` parameter: md5, `trust`, or `password`.

For remote monitoring, `pg_hba.conf` file should be updated according to SSL configuration done in `postgresql.conf` file.
If SSL is off then **`host`** entry should be added:

Table 195.

host	<DB name>	<user name>	<Agent machine IP>/32	<auth-method>

If SSL is on then **`hostssl`** entry should be added

Table 196.

hostssl	<DB name>	<user name>	<TEMA IP>/32	cert

Here, DB name is specific database name or you can enter all, user name is default user name postgres or you can enter all, auth-method can be md5 or password or trust.

Note: `host` or **`hostssl`** entry is not required for PostgreSQL server local monitoring.

A few of the attributes collected by the agent rely on the `pg_stat_statements` extension. To add `pg_stat_statements` first install the package `postgresql-contrib`. You must modify the `postgresql.conf` configuration file in order for the PostgreSQL server to load the `pg_stat_statements` extension.

1. Open the `postgresql.conf` file in a text editor and update the `shared_preload_libraries` line:

   ```
   shared_preload_libraries = 'pg_stat_statements'
   pg_stat_statements.track_utility = false
   ```

   These changes are required to monitor SQL statements, except utility commands.

   Note: The status of `pg_stat_statements.track_utility` is set or modified by a superuser only.
2. Update the `connection settings` section in `postgresql.conf` file `listen_addresses = <Agent machine IP>` here, mention specific agent IP or * to listen all IP addresses.

   Note: This setting is required for remote monitoring.
3. Restart the PostgreSQL server after you update and save the `postgresql.conf`.
4. Run the following SQL command by using psql, that should be connected to the same database that would be provided later in the agent configuration for JDBC connectivity:

   ```
   create extension pg_stat_statements;
   select pg_stat_statements_reset();
   ```

Note: The command `create extension` and function `pg_stat_statements_reset()` are run by a superuser only.

The view `pg_stat_statements` needs to be enabled for specific database, for more details refer https://www.postgresql.org/docs/9.6/static/pgstatstatements.html.

Review hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the PostgreSQL agent.

About this task

The PostgreSQL agent is a multiple instance agent; you must create the first instance and start the agent manually. The managed system name includes the instance name that you specify, for example, *`instance_name`*`:`*`host_name`*`:`*`pc`*, where *pc* is your two character product code. The managed system name is limited to 32 characters. The instance name that you specify is limited to 28 characters, minus the length of your host name. For example, if you specify `PostgreSQL2` as your instance name, your managed system name is `PostgreSQL2:hostname:PN`.

Important: If you specify a long instance name, the managed system name is truncated and the agent code is not displayed completely.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the "Change history" on page 54.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring the agent on Windows systems

You can use the IBM Cloud Application Performance Management window to configure the agent on Windows systems.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Monitoring Agent for PostgreSQL**, and then click **Configure agent**.
3. In the **Enter a unique instance name** field, type the agent instance name and click **OK**.
4. In the **Monitoring Agent for PostgreSQL** window, complete these steps:
   a. In the **IP Address** field, enter the IP address of a PostgreSQL server that you want to monitor remotely. If the agent is installed on the server to be monitored, retain the default value.
   b. In the **JDBC database name** field, enter a database name to change the default database name of `postgres`.
   c. In the **JDBC user name** field, enter a user name to change the default name of `postgres`.
   d. In the **JDBC password** field, enter the JDBC user password.
   e. In the **Confirm JDBC password** field, re-enter the password.
   f. In the **JDBC port number** field, enter a port number to change the default port number of 5432.
   g. In the **JDBC JAR file** field, enter the path for the PostgreSQL connector for the Java JAR file and click **Next**.
   h. In the **Java trace level** field, enter the trace level according to the IBM support instructions. The default trace level is `Error`.
   i. Click **OK**. The agent instance is displayed in the IBM Performance Management window.
5. Right-click the **Monitoring Agent for PostgreSQL** instance, and click **Start**.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the console, see "Starting the Cloud APM console" on page 957.

For help with troubleshooting, see the Troubleshooting section.

Configuring the agent on Linux systems

To configure the agent on Linux operating systems, you must run the script and respond to prompts.

Procedure

1. On the command line, enter the following command:
   install_dir/bin/postgresql-agent.sh config *instance_name*
2. When you are prompted to edit the agent for PostgreSQL settings, enter 1 to continue.
3. When you are prompted to enter a value for the following parameters, press Enter to accept the default value or specify a different value and press Enter:
   - **PostgreSQL Server IP Address**: Enter the IP address of a PostgreSQL server that you want to monitor remotely. If the agent is installed on the server to be monitored, retain the default value.
   - **JDBC database name**: Enter a database name to change the default database name of `postgres`.
   - **JDBC user name**: Enter a user name to change the default name of `postgres`.
   - **Enter JDBC password**: Enter the JDBC user password.
   - **Re-type JDBC password**: re-enter the password.
   - **JDBC port number**: Enter a port number to change the default port number of 5432.
   - **JDBC JAR file**: Enter the path for the PostgreSQL connector for the Java JAR file.

   Important: The version of the JDBC JAR file must be compatible with the version of the PostgreSQL database that is being monitored.
4. When you are prompted to enter a value for the `Java trace level` parameter, enter 2 to accept the default value or specify the trace level according to the IBM support instructions.
5. Run the following command to start the agent:

   ```
   install_dir/bin/postgresql-agent.sh start instance_name
   ```

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

For help with troubleshooting, see the Troubleshootingsection.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

You can use the silent response file to configure the PostgreSQL agent on Linux and Windows systems. After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

- To configure the agent by editing the silent response file and running the script without responding to prompts, complete the following steps:

1. In a text editor, open the silent response file that is available in this path: *install_dir*/samples/postgresql_silent_config.txt where *install_dir* is the installation directory of PostgreSQL agent. The default installation directory is /opt/ibm/apm/agent.
2. To edit the silent configuration file, complete the following steps:
   a. For the **PostgreSQL Server IP Address** parameter, specify the IP address of a PostgreSQL server that you want to monitor remotely. If the agent is installed on the server to be monitored, retain the default value.
   b. For the **JDBC database name** parameter, specify a database name to change the default database name of postgres.
   c. For the **JDBC user name** parameter, specify a user name to change the default name of postgres.
   d. For the **JDBC password** parameter, enter the JDBC user password.
   e. For the **JDBC port number** parameter, specify a port number to change the default port number of 5432.
   f. For the **JDBC JAR file** parameter, specify the path for the PostgreSQL connector for the Java JAR file

      Important: The version of the JDBC JAR file must be compatible with the version of the PostgreSQL database that is being monitored.
   g. For the **Java trace level** parameter, specify the trace level according to the IBM support instructions. The default trace level is Error.
3. Save and close the silent response file, and run the following command:

   ```
   install_dir/bin/postgresql-agent.sh config
   instance_name
   install_dir/samples/postgresql_silent_config.txt
   ```

   Where *instance_name* is the name that you want to give to the instance.
4. To start the agent, enter the following command:

   ```
   install_dir/bin/postgresql-agent.sh start instance_name
   ```

What to do next

Log in to the Cloud APM console to view data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

For help with troubleshooting, see the Troubleshooting section.

Configuring Python monitoring

Both on-premises and IBM Cloud Python applications can be monitored. Complete the corresponding configuration steps based on your application type.

About this task

Configure the Python data collector to monitor your on-premises and IBM Cloud Python applications.

Procedure

- Configure the data collector to monitor IBM Cloud applications.
  a) Configure the Python data collector to collect and send data for IBM Cloud applications. For instructions, see "Configuring the Python data collector for IBM Cloud applications" on page 664.
  b) Optional: Customize the monitoring capabilities of the Python data collector. For more information, see "Customizing the Python data collector for IBM Cloud applications" on page 665.
- Configure the data collector to monitor on-premises applications.

a) Configure the data collector to collect and send data to the Cloud APM server. For instructions, see "Configuring the Python data collector for on-premises applications" on page 669.

b) Optional: Customize the monitoring capabilities of the Python data collector. For more information, see "Customizing the Python data collector for on-premises applications" on page 670.

Configuring the Python data collector for IBM Cloud applications

To collect information about Python applications on IBM Cloud, you must configure the Python data collector.

Before you begin

1. Make sure the Python applications that you want to monitor have unique names. The Python data collector handles two different applications with the same name as one application, which might cause data display issues in the Cloud APM console.
2. Download the data collector package from IBM Marketplace. For detailed instructions, see "Downloading your agents and data collectors" on page 105.

About this task

To configure the data collector, you first deploy a pypi package server, and then install the data collector to a Python Django application.

Procedure

1. Extract files from the data collector package. The `python_datacollector_8.1.4.0.tgz` package is included in the extracted directory.
2. Extract the `python_datacollector_8.1.4.0.tgz` package, for example, by running the following command:

   ```
   tar -zxf python_datacollector_8.1.4.0.tgz
   ```

3. Find the `manifest.yml` file of the package server in the extracted directory and define the domain, host, and name in this file as is shown in the following example:

   ```
   domain: mybluemix.net
   name: pythondc
   host: pythondc
   ```

   Remember: The *host* and *name* values must be the same and unique.
4. From the `python_dc` directory, push the `pythondc` application to IBM Cloud by running the following command:

   ```
   cf push
   ```

5. In the `requirements.txt` file of your Python application, add the following lines:

   ```
   cryptography==1.9.0
   --extra-index-url https://<your_host_name_and_domain>/python-dc-repos/simple/
   ibm_python_dc
   ```

6. In the `settings.py` file of your Python application, add `ibm_python_dc.kpg_plugin.ResourceMiddleware` to the beginning of the `MIDDLEWARE_CLASSES` section, for example:

   ```
   MIDDLEWARE_CLASSES = (
       "ibm_python_dc.kpg_plugin.ResourceMiddleware",
       "mezzanine.core.middleware.UpdateCacheMiddleware",
       'django.contrib.sessions.middleware.SessionMiddleware',
       'django.middleware.common.CommonMiddleware',
   ```

7. From the directory that contains the `manifest.yml` file of your Python application, run the following command:

```
cf push
```

Tip: For a sample `manifest.yml` file, see “Sample manifest.yml file” on page 188.

Results

The data collector is configured and is connected to the Cloud APM server.

What to do next

You can verify the monitoring data for your IBM Cloud application is displayed in the Cloud APM console. For instructions on how to start the Cloud APM console, see Starting the Cloud APM console. For information about using the Applications editor, see Managing applications.

Customizing the Python data collector for IBM Cloud applications

You can add environment variables in the IBM Cloud user interface (UI) to customize the monitoring for your IBM Cloud application. Use the following information to add the variables according to your needs.

User-defined environment variables for the Python data collector

You can use the information in the following table to customize Python monitoring on IBM Cloud.

Table 197. Supported user-defined environment variables for Python monitoring on IBM Cloud

Variable name	Importance	Value	Description
APM_BM_GATEWAY_URL	Optional	• https://*<server ip or hostname>*:443 • http://*<server ip or hostname>*:80	The target on-premises server gateway URL.
APM_KEYFILE_PSWD	Optional	Encrypted password of the key file	The encrypted key file password that is paired with the key file. If you are a Linux user, you can use the `echo -n` *`<keyfile password>`* `| base64` command to encrypt your password. **Note:** Set this variable only when you configured the Gateway to use HTTPS.
APM_KEYFILE_URL	Optional	http://*<hosted http server>*:*<port>*/keyfile.p12	The URL to download the key file. **Note:** Set this variable only when you configured the Gateway to use HTTPS.
KPG_ENABLE_DEEPDIVE	Optional	• False • True	Enables or disables the collection of diagnostics data. • True: The default value. If you set this variable to `True`, diagnostics data are collected. • False: If you set this variable to `False`, diagnostics data are not collected. If you do not set this variable, diagnostics data are collected.

Table 197. Supported user-defined environment variables for Python monitoring on IBM Cloud (continued)

Variable name	Importance	Value	Description
KPG_DD_CONFIG_FILE	Optional	File name of the diagnostics monitoring configuration file.	File name of the diagnostics monitoring configuration file. The default file name is `kpg_dd_config.xml`. **Note:** After you customize the settings in this file, you must put it in the application root directory. If you do not set this variable, the default configuration file `kpg_dd_config.xml` in the data collector package is used.
KPG_DD_APP_PATH	Optional	Path to the Python application.	The path to the Python application or the module for which the data collector collects diagnostics data. Separate the paths of different Python applications and modules that you want to monitor with a semi-colon `;`. If you do not set this variable, the data collector collects data for requests and the modules that your application use. Data of requests in Python lib are not collected.
KPG_DD_SECURITY_FILTER	Optional	• True • False	• True: The default value. If you set this variable to `True`, values (such as the passwords) are masked in SQL statements and parameters are not displayed in group widget **Request Context**. • False: If you set this variable to `False`, values in SQL statements are not masked and parameters are displayed in group widget **Request Context**. If you do not set this variable, values (such as the passwords) are masked in SQL statements and parameters are not displayed in group widget **Request Context**.

Table 197. Supported user-defined environment variables for Python monitoring on IBM Cloud (continued)			
Variable name	**Importance**	**Value**	**Description**
KPG_GC_STATS	Optional	True	All statistical functions of python garbage collection are enabled. When you set this value to `True`, it equals running the following command: `gc.set_debug(gc.DEBUG_STATS | gc.DEBUG_COLLECTABLE | gc.DEBUG_UNCOLLECTABLE | gc.DEBUG_INSTANCES | gc.DEBUG_OBJECTS )` To disable KPG_GC_STATS, delete this environment variable. Do not set it to `False`. **Note:** Never set `KPG_SAVE_ALL=True` in your formal production environment. It is only for the debug mode. Make sure that enough memory is assigned to the application.
KPG_LOG_LEVEL	Optional	• DEBUG • ERROR • INFO	• DEBUG: Only useful debug information is printed in the log, for example, collected data, data that are sent to server, and server response. • ERROR: Only information about exceptions and unexpected situations is printed in the log. • INFO: The summary information about the data collector for the user to know what it is doing is printed in the log.
KPG_LOG_TOCONSOLE	Optional	• Y • True • Any other value that is not `False`	The log is printed to console and you can see the log by running the command **`cf logs <appname>`**.

Table 197. Supported user-defined environment variables for Python monitoring on IBM Cloud (continued)

Variable name	Importance	Value	Description
KPG_SAVE_ALL	Optional	True	All unreferenced objects are saved into `gc.garbage`, and you need to clear `gc.garbage` every minute (the data collector clears it for you). When the value is set to `True`, it equals running the following command: `gc.set_debug(gc.SAVE_ALL)` To disable KPG_SAVE_ALL, delete this environment variable. Do not set it to `False`. **Note:** Never set `KPG_SAVE_ALL=True` in your formal production environment. It is only for the debug mode. Make sure that enough memory is assigned to the application.

Unconfiguring the Python data collector for IBM Cloud applications

If you do not need to monitor your Python environment or if you want to upgrade the Python data collector, you must first unconfigure previous settings for the Python data collector.

Procedure

1. Go to the home directory of your Python application.
2. Remove the following lines from the `requirements.txt` file for the application:

   ```
   --extra-index-url https://<your_host_name_and_domain>/python_dc/static/python-dc-repos/
   simple/
   ibm-python-dc
   ```

3. In the `settings.py` file, remove the following line from the `MIDDLEWARE_CLASSES` section:

   ```
   ibm_python_dc.kpg_plugin.ResourceMiddleware
   ```

4. Run the following command to re-push the application for the changes to take effect:

   ```
   cf push
   ```

Results

You have successfully unconfigured the Python data collector.

What to do next

After you unconfigure the data collector, the Cloud APM console continues to display the data collector in any applications that you added the data collector to. The Cloud APM console will show that no data is available for the application and will not indicate that the data collector is offline. For information about how to remove the data collector from applications and from resource groups, see “Removing data collectors from Cloud APM console” on page 188.

Configuring the Python data collector for on-premises applications

To collect information about Python applications that run in your local environment, you must configure the Python data collector.

Before you begin

1. Make sure the Python applications that you want to monitor have unique names. The Python data collector handles two different applications with the same name as one application, which might cause data display issues in the Cloud APM console.
2. Download the data collector package from IBM Marketplace. For detailed instructions, see “Downloading your agents and data collectors” on page 105.

About this task

The data collector package is a preconfigured one with a preconfigured `global.environment` file and a `keyfile.p12` that is copied to the `etc` folder. As a result, the data collector automatically connects to the Cloud APM server.

The following procedure configures the data collector within your Python application with default settings. To customize data collector configuration, use the environment variables in the data collector configuration files. For more information, see “Customizing the Python data collector for on-premises applications” on page 670.

Procedure

1. Extract files from the data collector package. The `python_datacollector_8.1.4.0.tgz` package is included in the extracted directory.
2. Extract files from the data collector package, for example, by running the following command:

   ```
   tar -zxf python_datacollector_8.1.4.0.tgz
   ```

3. From the `python_dc` directory, run the following command:

   ```
   python server.py
   ```

4. Run the following command:

   ```
   pip install ibm_python_dc --extra-index-url http://host name or ip:8000/
   python-dc-repos/simple/ --trusted-host host name or ip
   ```

   where *host name or ip* is the name or IP address of the host to run your Python data collector repository.

   Important: Use either the name or the IP address to specify the host for both the URL and the trusted host in this command. For example, if you specify the host by using the IP address and the IP address is `9.42.36.180`, the command is as follows:

   ```
   pip install ibm_python_dc --extra-index-url http://9.42.36.180:8000/
   python-dc-repos/simple/ --trusted-host 9.42.36.180
   ```

5. In the `settings.py` file of your Python application, add `ibm_python_dc.kpg_plugin.ResourceMiddleware` to the `MIDDLEWARE_CLASSES` section to the format of the following example:

   ```
   MIDDLEWARE_CLASSES = (
       "ibm_python_dc.kpg_plugin.ResourceMiddleware",
       "mezzanine.core.middleware.UpdateCacheMiddleware",
       'django.contrib.sessions.middleware.SessionMiddleware',
       'django.middleware.common.CommonMiddleware',
   ```

Results

The data collector is configured with default settings and connected to the Cloud APM server.

What to do next

You can now log in to the Cloud APM server to view the monitoring data.

Remember: After you add your Python application to the Cloud APM console, you can view its monitoring data in the component named Python Runtime application.

For instructions on how to start the Cloud APM server, see Starting the Cloud APM console. For information about using the Applications editor, see Managing applications.

Customizing the Python data collector for on-premises applications

By modifying files in the data collector package, you can set the environment variables to customize the monitoring for your Python application.

Two files are provided to customize data collector settings, `global.environment` and `config.properties`. After you change the settings in these files, restart the Python application for the change to take effect.

By modifying the `global.environment` file, you can customize the connection between the data collector and the Cloud APM server. If you want to use another Cloud APM server instead of the default one, or the key file or its password is changed, modify the Cloud APM server to reconnect the data collector to the Cloud APM server.

By modifying the `config.properties` file, you can customize data collector behaviors according to your needs, such as enabling or disabling method trace.

The `global.environment` configuration file

Table 198 on page 670 shows the environment variables that you can set in the `global.environment` configuration file and the correlated descriptions. You can find the `global.environment` file in the `etc` folder where your Python data collector is installed, for example, `/root/.pyenv/versions/3.5.2/lib/python3.5/site-packages/ibm_python_dc/etc` directory.

Table 198. Supported environment variables in the `global.environment` file

Variable name	Importance	Value	Description
APM_BM_GATEWAY_URL	Optional	• https://*<server ip or hostname>*:443 • http://*<server ip or hostname>*:80	The target on-premises server gateway URL.
APM_KEYFILE_PSWD	Optional	Password of the key file	The key file password that is paired with the key file. **Note:** Set this variable only when you configured the Gateway to use HTTPS.
APM_KEYFILE_URL	Optional	http://*<hosted http server>*:*<port>*/keyfile.p12	The URL to download the key file. **Note:** Set this variable only when you configured the Gateway to use HTTPS.

The `config.properties` file

Table 199 on page 671 shows the environment variables you can set in the `config.properties` configuration files and the correlated description. You can find the `config.properties` file in the installation directory of your Python data collector, for example, `/root/.pyenv/versions/3.5.2/lib/python3.5/site-packages/ibm_python_dc` directory.

Table 199. Supported environment variables in the `config.properties` file

Variable name	Importance	Value	Description
KPG_ENABLE_DEEPDIVE	Optional	• False • True	• False: The default value. If you set this variable to `False`, diagnostics data will not be collected. • True: If you set this variable to `True`, diagnostics data will be collected. The default level is `True`. If you do not set this variable, diagnostics data will not be collected.
KPG_DD_CONFIG_FILE	Optional	File name of the diagnostics monitoring configuration file.	File name of the diagnostics monitoring configuration file. The default file name is `kpg_dd_config.xml`. **Note:** After you customize the settings in this file, you must put it in the application root directory. If you do not set this variable, the default configuration file `kpg_dd_config.xml` in the data collector package will be used.
KPG_DD_APP_PATH	Optional	Path to the Python application.	The path to the Python application or the module for which the data collector collects diagnostics data. Separate the paths of different Python applications and modules that you want to monitor with a semi-colon `;`. If you do not set this variable, the data collector will collect diagnostics data for requests and the modules that your application use. Data of requests in Python lib will not be collected.
KPG_DD_SECURITY_FILTER	Optional	• True • False	• True: The default value. If you set this variable to `True`, values (such as the passwords) will be masked in SQL statements and parameters will not be displayed in group widget **Request Context**. • False: If you set this variable to `False`, values in SQL statements will not be masked and parameters will be displayed in group widget **Request Context**. If you do not set this variable, values (such as the passwords) will be masked in SQL statements and parameters will not be displayed in group widget **Request Context**.

Table 199. Supported environment variables in the `config.properties` file (continued)

Variable name	Importance	Value	Description
KPG_GC_STATS	Optional	True	All statistical functions of python garbage collection are enabled. When you set this value to `True`, it equals running the following command: `gc.set_debug(gc.DEBUG_STATS | gc.DEBUG_COLLECTABLE | gc.DEBUG_UNCOLLECTABLE | gc.DEBUG_INSTANCES | gc.DEBUG_OBJECTS )` To disable KPG_GC_STATS, delete this environment variable. Do not set it to `False`. The default value is `True`. **Note:** Never set `KPG_GC_STATS=True` in your formal product environment. It is only for the debug mode. And make sure enough memory is assigned to the application.
KPG_LOG_LEVEL	Optional	• DEBUG • ERROR • INFO	• DEBUG: Only useful debug information will be printed in the log, for example, collected data, data that is sent to server, and server response. • ERROR: Only information about exceptions and very unexpected situations will be printed in the log. • INFO: The summary information about the data collector for the user to know what it is doing will be printed in the log. The default value is `ERROR`.
KPG_LOG_TOCONSOLE	Optional	• Y • True • Any other value that is not `False`	The log will be printed to console and you can see the log by running the command **`cf logs <appname>`**. The default value is `True`.

Table 199. Supported environment variables in the `config.properties` file (continued)

Variable name	Importance	Value	Description
KPG_SAVE_ALL	Optional	True	All unreferenced objects will be saved into `gc.garbage`, and you need to clear `gc.garbage` every minute (the data collector does this for you). When the value is set to `True`, it equals running the following command: `gc.set_debug(gc.SAVE_ALL)` To disable KPG_SAVE_ALL, delete this environment variable. Do not set it to `False`. The default value is `True`. **Note:** Never set `KPG_SAVE_ALL=True` in your formal product environment. It is only for the Debug mode. And make sure enough memory is assigned to the application.
APM_GW_PROXY_CONNECTION	Optional	http://*<server ip or hostname>*:port	The HTTP or HTTPS proxy that the Python data collector uses to send monitoring data.

Unconfiguring the Python data collector for on-premises applications

If you do not need to monitor your Python environment or if you want to upgrade the Python data collector, you must first unconfigure previous settings for the Python data collector.

Procedure

1. Go to the home directory of your Python application.
2. Remove the following lines from the `requirements.txt` file for the application:

```
--extra-index-url https://<your_host_name_and_domain>/python_dc/static/python-dc-repos/
simple/
ibm-python-dc
```

3. In the `settings.py` file, remove the following line from the `MIDDLEWARE_CLASSES` section:

```
ibm_python_dc.kpg_plugin.ResourceMiddleware
```

4. Run the `pip uninstall ibm_python_dc` command to uninstall the Python data collector from the Python run time.

Results

You have successfully unconfigured the Python data collector.

What to do next

After you unconfigure the data collector, the Cloud APM console continues to display the data collector in any applications that you added the data collector to. The Cloud APM console will show that no data is available for the application and will not indicate that the data collector is offline. For information about how to remove the data collector from applications and from resource groups, see "Removing data collectors from Cloud APM console" on page 188.

Configuring RabbitMQ monitoring

The Monitoring Agent for RabbitMQ monitors the health and performance of the RabbitMQ cluster resources, such as the nodes, queues, and channels of the cluster. You must configure the RabbitMQ agent so that the agent can collect the RabbitMQ data.

Before you begin

- Review the hardware and software prerequisites.
- Ensure that the RabbitMQ user, who connects to the node, has read permission and either the monitoring, administrator, or management tag is enabled for this user.
- If the agent is configured to use HTTPS protocol for monitoring, ensure following prerequisites are met:
  - Ensure that the RabbitMQ management plug-in is enabled on all nodes of the cluster, because if one node of the cluster fails, the RabbitMQ agent connects to a peer node that is available in the cluster.
  - Ensure that user has a CA certificate issued by a trusted vendor and a server certificate signed using this CA.
  - Ensure that user has truststore, in `jks` format, containing the CA certificate. Path to this truststore must be provided during agent configuration.
  - Ensure that the management plug-in is configured to use SSL. To configure, provide a port that is configured to use SSL, an accessible path, preferably local, to the CA certificate, a server certificate and a private key for the server certificate.

Review hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the RabbitMQ agent.

About this task

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the "Change history" on page 54.

The RabbitMQ agent is a multiple instance agent. You must create the first instance, and start the agent manually.

- To configure the agent on Windows systems, you can use the IBM Cloud Application Performance Management window or the silent response file.
- To configure the agent on Linux systems, you can run the script and respond to prompts, or use the silent response file.

Configuring the agent on Windows systems

You can use the IBM Cloud Application Performance Management window to configure the agent on Windows systems.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Monitoring Agent for RabbitMQ**, and then click **Configure agent**.

   Remember: After you configure the agent for the first time, the **Configure agent** option is disabled. To configure the agent again, click **Reconfigure**.
3. In the **Enter a unique instance name** field, type the agent instance name and click **OK**.
4. In the **Monitoring Agent for RabbitMQ** window, specify values for the configuration parameters, and then click **Next**.

For information about the configuration parameters, see the following topic: “Configuration parameters for the agent” on page 676

5. Right-click the **Monitoring Agent for RabbitMQ** instance, and click **Start**.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the console, see “Starting the Cloud APM console” on page 957.

For help with troubleshooting, see the Troubleshooting section.

Configuring the agent on Linux systems

To configure the agent on Linux operating systems, you must run the script and respond to prompts.

Procedure

1. On the command line, enter the following command: `*install_dir*/bin/rabbitmq.sh config *instance_name*` where *instance_name* is the name that you want to give to the instance:
2. When you are prompted to provide a value for the following parameters, press Enter to accept the default value, or specify a value and then press Enter:

   - IP Address
   - User Name
   - Password
   - Port Number
   - Java home
   - Java trace level

   For information about the configuration parameters, see the following topic: “Configuration parameters for the agent” on page 676
3. Run the following command to start the agent:

   ```
   install_dir/bin/rabbitmq.sh start instance_name
   ```

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see “Starting the Cloud APM console” on page 957.

For help with troubleshooting, see the Troubleshooting section.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

You can use the silent response file to configure the RabbitMQ agent on Linux and Windows systems. After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

1. Open the silent response file that is available at this path: `*install_dir*\samples\rabbitmq_silent_config.txt`
2. In the `rabbitmq_silent_config.txt` file, specify values for all mandatory parameters. You can also modify the default values of other parameters.

For information about the configuration parameters, see the following topic: “Configuration parameters for the agent” on page 676

3. Save the response file, and run the following command:

   Linux AIX `install_dir/bin/rabbitmq-agent.sh config install_dir/samples/rabbitmq_silent_config.txt`

   Windows `install_dir/bin/rabbitmq-agent.bat config install_dir/samples/rabbitmq_silent_config.txt`

4. Start the agent:

   Linux AIX Run the following command: `install_dir\bin\rabbitmq-agent.sh start`

   Windows Right-click **Monitoring Agent for RabbitMQ** and then click **Start**.

What to do next

Log in to the Cloud APM console to view data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see “Starting the Cloud APM console” on page 957.

For help with troubleshooting, see the Troubleshooting section.

Configuration parameters for the agent

When you configure the RabbitMQ agent, you can change the default values of the parameters, such as the instance name and the SSL validation certificates.

The following table contains detailed descriptions of the configuration parameters for the RabbitMQ agent.

Table 200. Names and descriptions of the configuration parameters for the RabbitMQ agent

Parameter name	**Description**	**Mandatory field**
IP Address	The IP address of the node where the RabbitMQ application is installed.	Yes
Username	The user name of the RabbitMQ user.	Yes
Password	The password to connect to the RabbitMQ management user interface.	Yes
Confirm Password	The same password that you entered in the **Password** field.	Yes
Port Number	The port number where the RabbitMQ management plugin is enabled. Use the default port number 15672, or specify another port number.	No
Java home	The path where the java plugin is installed. Use the default path `C:\Program Files\IBM\Java50`, or the directory path where java plugin is installed.	No
Java trace level	The trace level of the Java provider. The valid trace level values are as follows: • OFF • ERROR • WARN • INFO • DEBUG_MAX • ALL	No

Table 200. Names and descriptions of the configuration parameters for the RabbitMQ agent (continued)

Parameter name	Description	Mandatory field
Path to Truststore	The path to the truststore that is in the `jks` format and that contains the trusted CA certificate.	No
Protocol	The protocol that is used to create the connection.	No

Configuring Response Time Monitoring

The Response Time Monitoring agent monitors HTTP and HTTPS transactions on your HTTP server. Real browser-based user transactions (browser timings) are also monitored.

The Response Time Monitoring agent can be used to view the following levels of monitoring information:

HTTP and HTTPS Transaction Monitoring

HTTP transaction monitoring is automatically available when you install the Response Time Monitoring agent.

Depending on the type of HTTP server you are monitoring, HTTPS transaction monitoring might be automatically available or it might need to be manually configured. For more information, see "Response Time Monitoring Components" on page 678.

The Response Time Monitoring also monitors data relating to user counts, session counts, and devices.

Data is presented in the End User Transactions dashboard in the local time of the user and also used in the Requests and Response Time widget.

Real End user Transaction Monitoring (Browser timings)

Depending on the type of HTTP server you are monitoring, browser based timings might be automatically available or might need to be configured. For more information, see "Response Time Monitoring Components" on page 678.

Browser based timings are made possible with JavaScript Injection.

With JavaScript Injection, you can see further widgets and detail within the End User Transaction dashboards. JavaScript Injection ensures that the real end-user response time is collected from within the browser. It monitors the performance of HTTP pages and embedded objects for web pages that are served by the HTTP Server. The following additional real end-user transaction details are available:

- Client Total Time in the Transaction Requests and Response Time widget
- Response Time for Client Time transactions in the Transactions - Top 10 widget
- Render Time Breakdown

For information on how to configure JavaScript injection, see "JavaScript Injection" on page 681

Viewing transaction dashboards

View transaction data in the Application Performance Dashboard.

A number of widgets are available in the Application Performance Dashboard that give contextual detail about transactions.

Good requests have a response time less than 10 seconds. *Slow requests* have a response time greater than 10 seconds. The 10 second value used to determine good vs slow response time is not configurable. The following widgets are available:

- Worst by User - Top 5 group widget
- Worst by Device - Top 5 group widget
- Requests and Response Time group widget

- Transactions - Top 10 group widget
- Transaction Requests and Response Time group widget
- Runs On group widget
- Subtransactions group widget
- Transaction Instances group widget
- Users by Location group widget
- Users at Selected Location group widget
- User Sessions at Selected Location - Top 10 group widget
- User Request and Response Time group widget
- User Sessions - Top 10 group widget
- Device Request and Response Time group widget
- Session group widget
- Session Requests group widget
- Session Instances group widget
- Transaction Instances group widget
- Middleware Transactions Summary dashboard
- Middleware Transactions Details dashboard
- Event thresholds for Transaction Monitoring
- Interaction Aggregate Data
- Transactions Aggregate Data
- WRT Transaction Status

Response Time Monitoring Components

The base functionality of the Response Time Monitoring agent is:

- HTTP transaction monitoring
- HTTPS transaction monitoring
- Browser-based timings (by using JavaScript Injection) monitoring

To see more detailed descriptions of this functionality, see "Configuring Response Time Monitoring" on page 677.

Depending on the type of HTTP Server you are monitoring, the base functionality of the Response Time Monitoring agent is provided by using one of the following components:

IBM HTTP Server Response Time module

The IBM HTTP Server Response Time module can monitor only http content type of: text/html, application/xml or application/json. (without Javascript Injection)

The IBM HTTP Server Response Time module can currently not monitor javascript instrument compressed requests.

The IBM HTTP Server Response Time module Javascript Injection can currently monitor only http content type of text/html.

Packet Analyzer

The Packet Analyzer can only monitor content type of text/html. The Packet Analyzer can monitor gzip compressed requests.

If you are monitoring an IBM HTTP Server or Apache HTTP Server on AIX or Linux, use the IBM HTTP Server Response Time module. It is possible but not recommended to use the Packet Analyzer. IBM HTTP Server Response Time module is not supported on Windows. Use Packet Analyzer in a Windows environment.

If you are monitoring any other HTTP server, use Packet Analyzer. The Packet Analyzer is supported on Window, Linux, and AIX.

Planning the installation

Plan your Response Time Monitoring agent installation based on your operating system and type of HTTP server.

The base functionality of the Response Time Monitoring can be provided by using one of the following components:

- Packet Analyzer
- IBM HTTP Server Response Time module

You determine which component to use based on:

- The type of HTTP server you are installing the Response Time Monitoring agent on.
- Which operating system the HTTP server is installed on.

The considerations for installing the Response Time Monitoring agent with Packet Analyzer are:

- Packet Analyzer is supported on all operating systems (Windows, Linux, and AIX).
- Packet Analyzer monitors HTTP transactions on port 80 for all operating systems.
- HTTPS transaction monitoring is not automatic, and must be configured manually. The Response Time Monitoring agent requires access to the SSL Certificates so that it can decrypt SSL traffic from HTTP servers. For more information, see “Monitoring HTTPS transactions” on page 697.
- Packet Analyzer is supported on all HTTP servers but it is only recommend for Sun Java System Web Server and Microsoft Internet Information Services.
- AIX Linux Windows To install the Response Time Monitoring agent to work with Packet Analyzer on IBM HTTP Server or Apache HTTP Server, the HTTP server must be stopped. When you install the Response Time Monitoring agent and the HTTP server is stopped Packet Analyzer is automatically enabled.
- AIX Linux Windows Although Packet Analyzer can be configured for IBM HTTP Server or Apache HTTP Server, it is not recommended; IBM HTTP Server Response Time module is recommended.
- Windows WinPcap 4.1.3 is required before you install the Response Time Monitoring agent.
- Windows AIX Linux If you install the Response Time Monitoring agent on Sun Java System Web Server or Microsoft Internet Information Services, Packet Analyzer is automatically configured.

The considerations for installing the Response Time Monitoring agent with IBM HTTP Server Response Time module are:

- The IBM HTTP Server Response Time module is a component of the HTTP Server agent. You must install the HTTP Server agent either before the Response Time Monitoring agent or else install at the same time. For more information, see “IBM HTTP Server Response Time module” on page 687.
- IBM HTTP Server Response Time module is supported on all operating systems (Windows, Linux, and AIX). IBM HTTP Server Response Time module supports IBM HTTP Server version 7, 8, and 9.
- Install the Response Time Monitoring agent and the HTTP Server agent on the same machine.
- IBM HTTP Server Response Time module monitors all ports for HTTP and HTTPS request on AIX, Linux, and Windows.
- IBM HTTP Server Response Time module is only supported for IBM HTTP Server or Apache HTTP Server.
- Both agents are automatically started but you must restart the HTTP server.

The following table describes the different combinations for automatically configuring the Response Time Monitoring agent.

Table 201. Scenarios for automatically configuring the Response Time Monitoring agent

HTTP server and OS combinations	Packet Analyzer	IBM HTTP Server Response Time module
Sun Java System Web Server or Microsoft Internet Information Services on AIX or Linux	Automatic	Not supported
Sun Java System Web Server or Microsoft Internet Information Services on Windows	Automatic	Not supported
IBM HTTP Server or Apache HTTP Server on AIX, Linux, or Windows	Automatic if HTTP server is stopped.	Automatic if HTTP server is present and configured

Planning the configuration

HTTP Monitoring is automatically enabled when you install the Response Time Monitoring agent. Depending on your environment, HTTPS and JavaScript Injection might need to be manually configured.

HTTP Monitoring

HTTP transaction monitoring is configured automatically for Packet Analyzer and IBM HTTP Server Response Time module, if you follow the installation guidelines. See "Planning the installation " on page 679.

HTTPS Monitoring

HTTPS transaction monitoring is configured automatically for IBM HTTP Server Response Time module if you follow the installation guidelines. See "Planning the installation " on page 679.

HTTPS transaction monitoring needs to be manually configured for Packet Analyzer. For more information, see " Packet Analyzer roadmap" on page 695.

Browser-based timings (using JavaScript Injection)

Browser-based timings (using JavaScript Injection) are configured automatically for IBM HTTP Server Response Time module.

Browser-based timings (using JavaScript Injection) needs to be manually configured for Packet Analyzer. For more information, see " Packet Analyzer roadmap" on page 695.

The following table describes how the base functionality is configured for each component:

Table 202. Base functionality configuration

	Packet Analyzer on Sun Java System Web Server or Microsoft Internet Information Services (Windows, Linux, or AIX)	IBM HTTP Server Response Time module on IBM HTTP Server or Apache HTTP Server (Windows, Linux, or AIX)
HTTP transaction monitoring	Enabled automatically.	Enabled automatically
HTTPS transaction monitoring	Must be manually configured.	Enabled automatically

Table 202. Base functionality configuration (continued)

	Packet Analyzer on Sun Java System Web Server or Microsoft Internet Information Services (Windows, Linux, or AIX)	**IBM HTTP Server Response Time module on IBM HTTP Server or Apache HTTP Server (Windows, Linux, or AIX)**
Real end user transaction monitoring (Browser Timings) using JavaScript Instrumentation	Must be manually configured.	Enabled automatically

JavaScript Injection

You can customize the data that is collected by the Response Time Monitoring agent for display in the End User Transactions dashboards.

To ensure a good user experience for a web-based application, you must monitor the performance that is perceived by the actual users. This means monitoring at the browser level.

To be able to monitor at the browser level, you need to inject JavaScript Monitoring Code into the pages that you want to monitor. This code then collects data for particular browser timings.

This is done using JavaScript Injection in the web pages and objects that you want to monitor. Depending on the type of HTTP server that you installed your Response Time Monitoring agent on, there are two methods you can use to collect real end-user transaction response time information.

- If you are using an IBM HTTP Server or an Apache HTTP server, use IBM HTTP Server Response Time module. The IBM HTTP Server Response Time module automatically does JavaScript Injection. The IBM HTTP Server Response Time module is a component of the HTTP Server agent. It is installed and configured as part of the HTTP Server agent. For more information, see "IBM HTTP Server Response Time module" on page 687.
- If you are using any other supported HTTP server, use Packet Analyzer. With Packet Analyzer, you must manually instrument your web pages to collect browser timings. For more information, see "Adding the JavaScript monitoring component to your application" on page 696.

The following table shows the features that are available in the Application Performance Dashboard if you configure your environment for Packet Analyzer or IBM HTTP Server Response Time module:

	Packet Analyzer	**IBM HTTP Server Response Time module**
Transactions Top 10	✓	✓
Server Time	✓	✓
Render Time Breakdown	—	✓
AJAX Subtransactions	✓	✓
Resource Timing data in Subtransactions table	—	✓
Transaction Instances (Top 10)	✓	✓
Transaction Instance Topology	✓	✓
Application Topology	✓	✓
Automatic instrumentation of JavaScript Injection	N/A	✓

Reconfiguring the Response Time Monitoring on Windows

Use the `rt-agent` interactive configuration command or IBM Cloud Application Performance Management utility to configure or reconfigure the agent.

Before you begin

If you are enabling HTTPS transaction monitoring, make sure that the Monitoring Agent for HTTP Server is not installed on the same machine. Otherwise the Response Time Monitoring configuration does not change the HTTPS setting for Packet Analyzer.

About this task

The Response Time Monitoring agent is configured automatically following installation. Follow the installation guidelines: "Planning the installation " on page 679. You might need to reconfigure, for example, if you want to monitor a different port or monitor HTTPS transactions.

The installation directory is referred to as *install_dir*. The default installation directory is: `C:\IBM\APM\`

As an alternative to using the `rt-agent` interactive configuration command, you can configure the agent in the IBM Cloud Application Performance Management utility. For more information, see "Using the IBM Cloud Application Performance Management window on Windows systems" on page 182.

Procedure

To customize your data settings, complete the following steps:

1. On the computer where the Response Time Monitoring agent is installed, stop the agent:

   `install_dir\BIN\rt-agent.bat stop`
2. Use silent configuration to configure the agent:

   ```
   install_dir\BIN\rt-agent.bat config install_dir\samples\rt_silent_config.txt
   ```

   If you want to enable HTTPS transaction monitoring, uncomment the following lines in the silent configuration file. The sample of `rt_silent_config.txt` to configure the Response Time Monitoring agent to monitor HTTPS on Windows should look like this:

   ```
   # Monitor HTTPS transactions
   KT5MONITORHTTPSAPP=YES

   # HTTPS keystore (e.g. - /tmp/keys.kdb)
   KT5KEYSTORE=C:\keys\key.kdb

   # HTTPS server certificate map (eg - certAlias,9.48.152.1,443;...)
   KT5SERVERMAP=certalias,9.48.152.1,443

   # Monitor network traffic for the NIC hosts this IP address
   #KT5MONITORIP=9.48.152.1
   ```
3. Restart the Response Time Monitoring agent for the changes to take effect:

   `install_dir\BIN\rt-agent.bat start`

Results

Data from the new source is displayed in the dashboards that are associated with Response Time Monitoring.

Reconfiguring the Response Time Monitoring on AIX and Linux

Use the `rt-agent` configuration command to configure or reconfigure the Response Time Monitoring agent.

About this task

The Response Time Monitoring agent is configured automatically following installation. Follow the installation guidelines: "Planning the installation " on page 679. You might need to reconfigure, for example, if you want to monitor a different port.

The installation directory is referred to as *`install_dir`*. The default installation directory is : `/opt/ibm/apm/agent`.

Use the same root user that you used to install the agent to start, stop, and configure the agent.

Procedure

To reconfigure, complete the following steps:

1. On the computer on which the Response Time Monitoring agent is installed, stop the agent:

   ```
   install_dir/bin/rt-agent.sh stop
   ```

2. Use either interactive or silent configuration:

   a) Interactive configuration:

   ```
   install_dir/bin/rt-agent.sh config
   ```

   b) Silent configuration:

   ```
   install_dir/bin/rt-agent.sh config  install_dir/samples/rt_silent_config.txt
   ```

   If you want to enable HTTPS transaction monitoring, uncomment the following lines in the silent configuration file. The sample of `rt_silent_config.txt` to configure the Response Time Monitoring agent to monitor HTTPS on AIX and Linux should look like this:

   ```
   # Monitor HTTPS transactions
   KT5MONITORHTTPSAPP=YES

   # HTTPS keystore (e.g. - /tmp/keys.kdb)
   KT5KEYSTORE=/tmp/keys.kdb

   # HTTPS server certificate map (eg - certAlias,9.48.152.1,443;...)
   KT5SERVERMAP=certalias,9.48.152.1,443

   # Monitor network traffic for the NIC hosts this IP address
   #KT5MONITORIP=9.48.152.1
   ```

3. Restart the Response Time Monitoring agent for the changes to take effect:

   `install_dir/bin/rt-agent.sh start`

Results

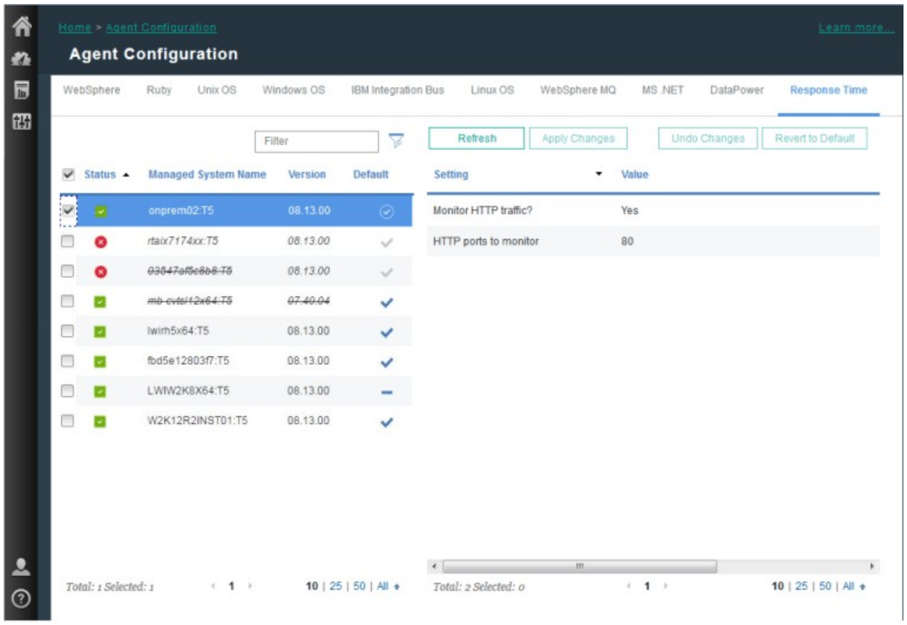
Data from the new source is displayed in the dashboards that are associated with Response Time Monitoring.

Configuring using the Agent Configuration Page

You can use the **Agent Configuration** page in the Cloud APM console to see what agents are installed. Where applicable, you can disable or enable HTTP transaction monitoring and set the ports that are monitored by Response Time Monitoring agents.

Agent Configuration

To access the Response Time Monitoring **Agent Configuration** page, in the Cloud APM console, select **System Configuration** > **Agent Configuration**, then select the **Response Time** tab.



☑	Status ▲	Managed System Name	Version	Default
☑	✓	onprem02:T5	08.13.00	✓
□	✗	*rtaix7174xx:T5*	*08.13.00*	✓
□	✗	~~*03547af5c8b8:T5*~~	*08.13.00*	✓
□	✓	~~*mb-cvtsl12x64:T5*~~	~~*07.40.04*~~	✓
□	✓	lwirh5x64:T5	08.13.00	✓
□	✓	fbd5e12803f7:T5	08.13.00	✓
□	✓	LWIW2K8X64:T5	08.13.00	–
□	✓	W2K12R2INST01:T5	08.13.00	✓

Total: 1 Selected: 1 ‹ 1 › 10 | 25 | 50 | All

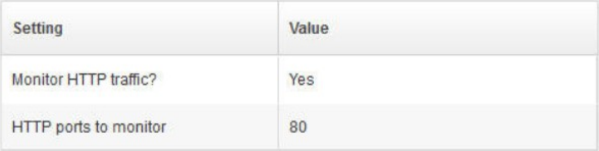
Setting	Value
Monitor HTTP traffic?	Yes
HTTP ports to monitor	80

Total: 2 Selected: 0 ‹ 1 › 10 | 25 | 50 | All

The **Agent Configuration** page lists the systems in your environment on which Response Time Monitoring is installed.

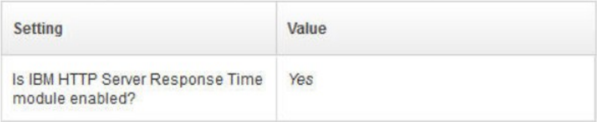
For each system with a Response Time Monitoring agent installed, the **Agent Configuration** page shows:

- Whether the system is online (check mark with green background) or offline (cross with red background).
- The version of the Response Time Monitoring agent that is installed.
- If central configuration cannot determine the type of the agent (that is, whether the Packet Analyzer or IBM HTTP Server Response Time module is being used to monitor HTTP transactions), only the agent is struck through. Generally, the type of agent can't be determined where the agent is not sending agent details through ASF activity.
- Whether the system uses the default configuration values or has some custom values set.
- The ports that are monitored if the Response Time Monitoring agent is using the Packet Analyzer to monitor HTTP transactions.



Setting	Value
Monitor HTTP traffic?	Yes
HTTP ports to monitor	80

- Whether IBM HTTP Server Response Time module, together with HTTP Server agent, is being used to monitor HTTP transactions.



Setting	Value
Is IBM HTTP Server Response Time module enabled?	*Yes*

Tip: IBM HTTP Server Response Time module monitors HTTP and HTTPS transactions automatically. No further configuration of the Response Time Monitoring agent is required.

Select an agent to display its configuration settings. To find a particular agent, enter part or all of the name of the system on which it is installed in the **Filter** field.

Customizations that are made on the **Agent Configuration** page take precedence over any other customization and over default values.

If you change your mind about the settings you changed, click **Undo Changes** to revert to the last settings that were saved, or click **Revert to Default** to revert to the default values.

The new configuration values are sent to Central Configuration Services and online agents are then automatically reconfigured without having to be restarted. If the agent is offline, it downloads the new configuration settings when it comes online. Data from the new ports is displayed in the dashboards that are associated with Response Time Monitoring when the data is refreshed.

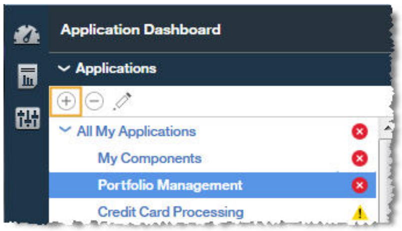
Adding Applications

After you install the Response Time Monitoring agent, you might need to add the applications that you want to monitor to the Application Performance Dashboard.

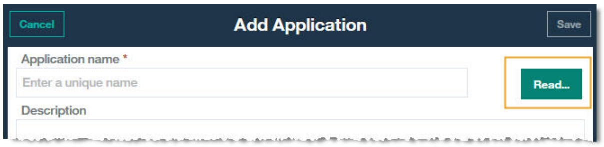
Procedure

To add applications to the Application Performance Dashboard:

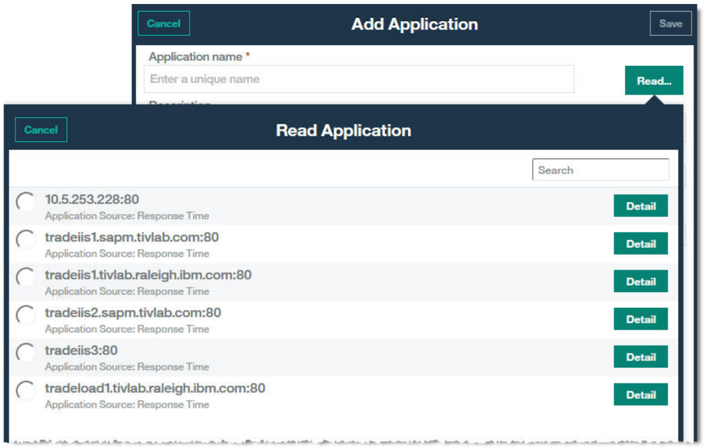
1. In the Application Performance Dashboard, click **Add Application**.



2. Select **Read** to open a list of discovered applications.



3. Select the application that you want to monitor.



○ 10.5.253.228:80 — Application Source: Response Time

○ tradeiis1.sapm.tivlab.com:80 — Application Source: Response Time

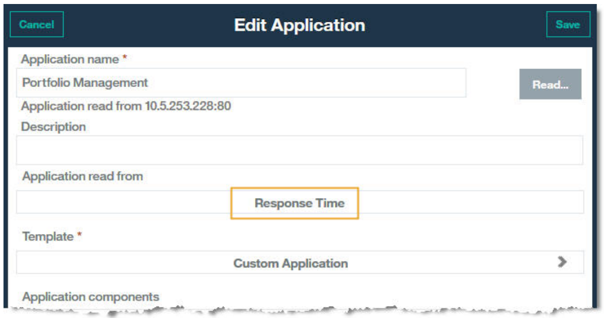
○ tradeiis1.tivlab.raleigh.ibm.com:80 — Application Source: Response Time

○ tradeiis2.sapm.tivlab.com:80 — Application Source: Response Time

○ tradeiis3:80 — Application Source: Response Time

○ tradeload1.tivlab.raleigh.ibm.com:80 — Application Source: Response Time

Response Time is displayed as the source repository in the **Application read from** field, and any components are listed in **Application components**.



Edit Application

Application name *

Portfolio Management

Application read from 10.5.253.228:80

Description

Application read from

Response Time

Template *

Custom Application

Application components

4. No further configuration is needed to display applications that are monitored by the Response Time Monitoring agent in the Application Performance Dashboard. Click **Save** in the **Add Application** window.

Results

Applications that are detected by the Response Time Monitoring agent are listed in **All My Applications** in the Application Performance Dashboard.

Configuring the IBM HTTP Server Response Time module

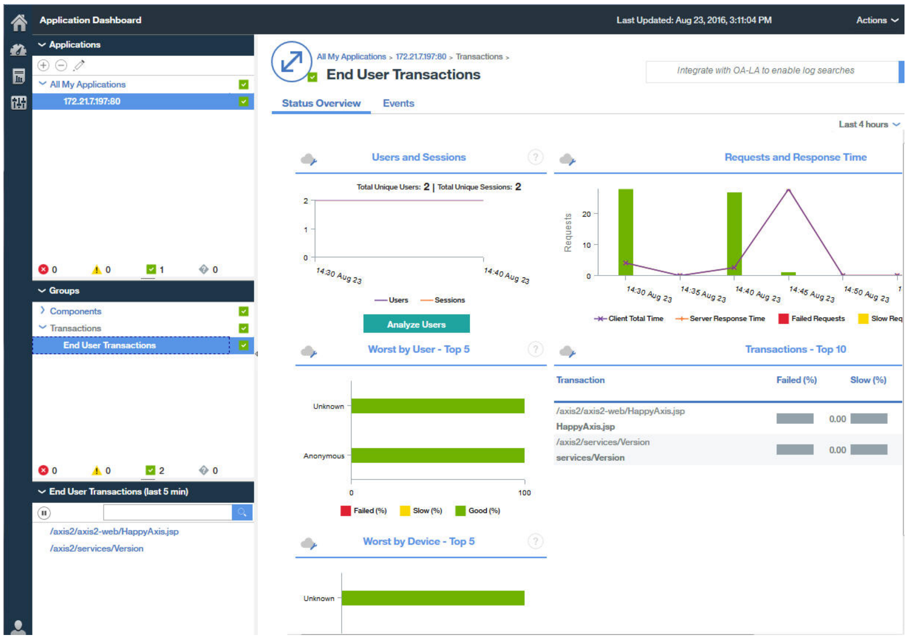
For IBM HTTP Server and Apache HTTP Server, use the IBM HTTP Server Response Time module to view real-end user response time monitoring metrics for HTTP pages.

The IBM HTTP Server Response Time module is installed and configured as part of the HTTP Server agent. The IBM HTTP Server Response Time module only works in conjunction with IBM HTTP Server and Apache HTTP Server on AIX, Linux, and Windows. The IBM HTTP Server Response Time module monitors all ports for HTTP and HTTPS requests.

Using JavaScript, IBM HTTP Server Response Time module inserts a header into web pages that are served by an IBM HTTP Server so that the Response Time Monitoring agent can monitor those pages. Embedded objects that are loaded by the page are tracked by using cookies. Transaction information from web pages that are served by IBM HTTP Server or Apache is then included in the End User Transactions dashboards.

For example:

End User Transactions workspace that shows data that is collected from the IBM HTTP Server Response Time module in the Transactions - Top 10:



Application Dashboard

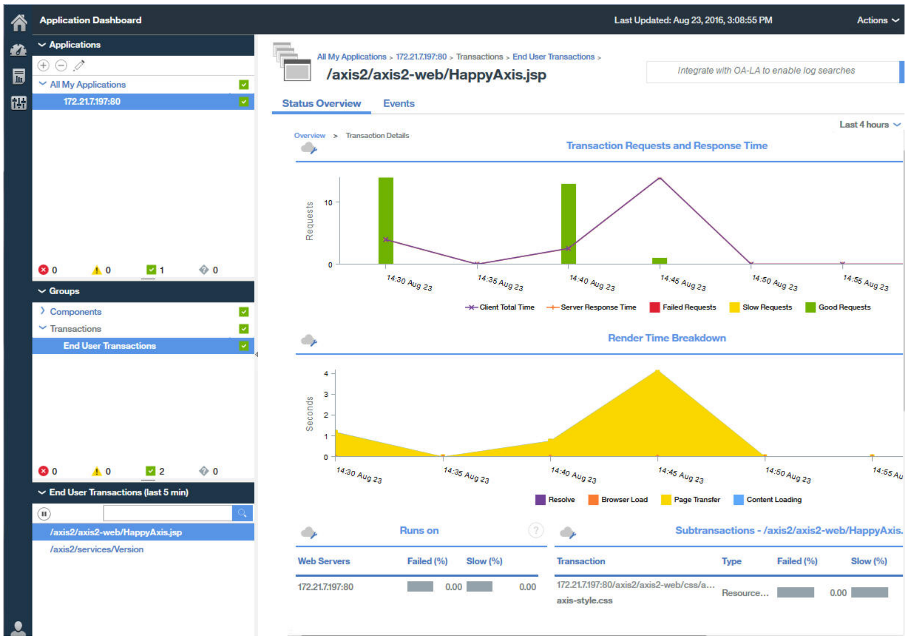
Last Updated: Aug 23, 2016, 3:11:04 PM

End User Transactions

Transactions - Top 10

Transaction	Failed (%)	Slow (%)
/axis2/axis2-web/HappyAxis.jsp HappyAxis.jsp		0.00
/axis2/services/Version services/Version		0.00

End User Transactions workspace that shows data that is collected from the IBM HTTP Server Response Time module in the **Subtransactions** table



Web Servers	Failed (%)	Slow (%)
172.21.7.197:80	0.00	0.00

Transaction	Type	Failed (%)	Slow (%)
172.21.7.197:80/axis2/axis2-web/css/a... axis-style.css	Resource...	0.00	

IBM HTTP Server Response Time module

The IBM HTTP Server Response Time module is a part of the HTTP Server agent. But it works in conjunction with the Response Time Monitoring agent to monitor application transactions on supported HTTP servers.

When you install the Response Time Monitoring agent to work with IBM HTTP Server Response Time module, it monitors all ports for HTTP and HTTPS requests.

The IBM HTTP Server Response Time module is a part of the HTTP Server agent. You must install the HTTP Server agent either before the Response Time Monitoring agent or else install at the same time.

The HTTP Server agent, is composed of two plug-ins:

1. `khu_module` - this is the HTTP Server agent. This plug-in is responsible for all the dashboards associated with the HTTP Server agent. For more information, see the HTTP Server agent Reference.
2. `wrt_module` - this is the IBM HTTP Server Response Time module.

These two plug-ins are indicated in the HTTP Server agent configuration file. The HTTP Server agent configuration file is as follows for Apache HTTP Server:

```
khu.usr.local.apache24.conf.httpd.conf
```

The file is as follows for IBM HTTP Server:

```
khu.opt.IBM.HTTPServer.conf.httpd.conf
```

The naming rule of this file is: `khu.(full path of the http server conf file, change the / to .).conf`

```
LoadModule khu_module

LoadModule wrt_module
```

For the IBM HTTP Server Response Time module to work, the HTTP server configuration file must have an include statement referencing the HTTP Server agent configuration file. For example:

```
include /opt/ibm/apm/agent/tmp/khu/khu.opt.IBM.HTTPServer.conf.httpd.conf
```

This include statement, enables both plugins at the same time. For more information, see “Configuring HTTP Server monitoring” on page 265.

Installing and configuring the IBM HTTP Server Response Time module

Configuration is automatic on the Response Time Monitoring agent side. The IBM HTTP Server Response Time module needs to be installed and configured as part of the HTTP Server agent. The Response Time Monitoring agent automatically detects the IBM HTTP Server Response Time module and enables it.

About this task

Procedure

1. Install HTTP Server agent which automatically installs the IBM HTTP Server Response Time module.
2. Configure the HTTP Server agent. This enables the IBM HTTP Server Response Time module. For more information, see “Configuring HTTP Server monitoring” on page 265.
3. Install Response Time Monitoring agent agent as `root` or `Administrator`, depending on your operating system. For detailed instructions, see Chapter 6, “Installing your agents,” on page 121.
4. Restart IBM HTTP Server. When the Response Time Monitoring installer detects the HTTP Server agent, the Response Time Monitoring agent enables the IBM HTTP Server Response Time module automatically.

Enabling IBM HTTP Server Response Time module manually

You can enable IBM HTTP Server Response Time module manually to monitor the performance of HTTP pages and embedded objects for web pages that are served by IBM HTTP Server.

About this task

The IBM HTTP Server Response Time module is enabled automatically when the HTTP Server agent is installed and configured. However, you might want to manually enable the IBM HTTP Server Response Time module.

Procedure

To enable IBM HTTP Server Response Time module manually on Linux, AIX, or Windows, complete the following steps:

1. Linux AIX

   To enable the IBM HTTP Server Response Time module manually on Linux or AIX, complete the following steps:

   a) Stop the Response Time Monitoring agent.

      Run

      ```
      $AGENT_HOME/bin/rt-agent.sh stop
      ```

      where *$AGENT_HOME* might be `/opt/ibm/apm/agent` on a Linux system, or `/opt/ibm/ccm/agent` on an AIX system.

   b) Run a configuration command and use *$AGENT_HOME*`/samples/rt_silent_config_ihs.txt` to add the load modules to the web server configuration file and set the configuration parameters for the IBM HTTP Server Response Time module.

   c) Restart the Response Time Monitoring agent.

2. Windows

   To enable IBM HTTP Server Response Time module manually on Windows, complete the following steps:

   a) Stop the Response Time Monitoring agent.

      Run

```
AGENT_HOME/bin/rt-agent.bat stop
```

where *AGENT_HOME* might be `C:\IBM\APM` on a Windows system.

b) Run a configuration command and use *AGENT_HOME*`\samples\rt_silent_config_ihs.txt` to add the load modules to the web server configuration file and set the configuration parameters for the IBM HTTP Server Response Time module.

For example,

```
AGENT_HOME\bin\rt_agent.bat config AGENT_HOME\samples\rt_silent_config_ihs.txt
```

c) Restart the Response Time Monitoring agent.

Disabling the IBM HTTP Server Response Time module manually

To disable the IBM HTTP Server Response Time module and use the Packet Analyzer again, reconfigure the agent and turn off monitoring by the IBM HTTP Server Response Time module.

About this task

Use the following procedures to disable the IBM HTTP Server Response Time module.

Procedure

To reconfigure the agent interactively on Linux, AIX, or Windows, complete the following steps:

1. Linux AIX

   To reconfigure the agent interactively on Linux and AIX, complete the following steps:

   a) Run *`install_dir`*`/bin/rt-agent.sh config`

   Where *`install_dir`* is `/opt/ibm/apm/agent` on Linux and AIX.

   b) Restart the Response Time Monitoring agent.

   Alternatively, to set the parameters manually:

   a) Open *`install_dir`*`/config/`*`hostname`*`_t5.cfg` in a text editor.

   where *install_dir* is `/opt/ibm/apm/agent` on Linux and AIX.

   b) Set the following parameters:

   ```
   KT5DISABLEANALYZER=NO
   KT5ENABLEWEBPLUGIN=NO
   ```

   c) Restart the Response Time Monitoring agent.

2. Windows

   To reconfigure the agent interactively on Windows, complete the following steps:

   a) Set the related parameters manually, open the *install_dir*`\TMAITM6_x64\`*`hostname`*`_t5.cfg` in a text editor.

   where *install_dir* is `C:\IBM\APM` on Windows.

   b) Set the following parameters:

   ```
   KT5DISABLEANALYZER=NO
   KT5ENABLEWEBPLUGIN=NO
   ```

   c) Restart the Response Time Monitoring agent.

Advanced Configuration of IBM HTTP Server Response Time module

There are a number of advanced configuration options for the IBM HTTP Server Response Time module.

The IBM HTTP Server Response Time module is configured automatically, but there are a number of advanced configuration tasks you can perform to fine tune the performance and features.

Disabling resource timing monitoring

Resource timing monitoring is enabled for all IBM HTTP Server Response Time module instances installed by HTTP Server agent.

About this task

If you want to reduce the number of resources that are monitored by IBM HTTP Server Response Time module, or disable resource timing monitoring to reduce the processing load that is required for monitoring a particular IBM HTTP Server, complete the following steps:

Procedure

To edit the generated HTTP Server agent configuration file, complete the following steps:

1. At the end of the HTTP server configuration file (httpd.conf), append

   ```
   WrtMaxPostResourcesSize
   ```

2. Set one of the following values:
   - `WrtMaxPostResourcesSize -1`, to monitor all resources
   - `WrtMaxPostResourcesSize 0`, to turn off resource monitoring
   - `WrtMaxPostResourcesSize` *n*, to monitor a particular number of resources, `10` by default. For example, set `WrtMaxPostResourcesSize 2` to set a maximum of two resources to post to the server.
3. Restart the HTTP server.

Disabling ARM Correlator generation

By default, ARM Correlator generation is enabled which allows the IBM HTTP Server Response Time module to link to any back-end servers in the topology. You can disable ARM Correlator generation if required.

About this task

Restriction: If you disable ARM Correlator generation, IBM HTTP Server Response Time module cannot link to back-end servers such as WebSphere Application Server. Disable ARM Correlator generation only under advice from IBM Software Support.

Procedure

To disable ARM Correlator generation, complete the following steps:

1. At the end of the HTTP server configuration file (httpd.conf), append

   ```
   WrtDisableArmCorr
   ```

2. Restart the HTTP server

Disabling Response Time Monitoring with Client Time (JavaScript Instrumentation)

In IBM Application Performance Management, Response Time Monitoring with Client Time (JavaScript Instrumentation) is enabled for all IBM HTTP Server installations running on all computers.

About this task

Procedure

To disable JavaScript Injection manually, complete the following steps:

1. Open the HTTP server configuration file located here: *`HTTP_Server_root`*`/conf/httpd.conf`
2. Navigate to the line added for HTTP Server agent, and append the following line after it:

```
WrtDisableJSI
```

For example,

```
include /opt/ibm/apm/agent/tmp/khu/khu.opt.IBM.HTTPServer.conf.httpd.conf
WrtDisableJSI
```

3. Save `httpd.conf` file, and recycle the HTTP server.

Bypassing the WRTCorrelator cookie

When JavaScript injection is enabled for the Response Time Monitoring agent on the WebSphere Portal, cookies such as the WRTCorrelator cookie might cause issues while you are using the WebSphere Portal. To prevent these issues, you can set the WRTCorrelator cookie to ignore.

Procedure

1. If necessary, start the `WebSphere_Portal` server.
2. Log in to the WebSphere® Integrated Solutions Console.
3. Go to **Resources** > **Resource Environment** > **Resource Environment Providers**.
4. Select **WP ConfigService**.
5. Under Additional Properties, select **Custom Properties**.
6. Click **New**.
7. Specify the name of the property, the WRTCorrelator cookie in this case, and set the value of the property to ignore.

   To set the WRTCorrelator cookie to ignore, enter the following:

   ```
   cookie.ignore.regex =
   digest\.ignore.*|LTPAToken|LTPAToken2|JSESSIONID|WASReqURL|WRTCorrelator
   |PD_STATEFUL.*
   ```

8. Click **Apply** and save your changes.
9. Log out of the WebSphere® Integrated Solutions Console.

Results

The WRTCorrelator cookie is set to ignore and issues such as looping are prevented.

Excluding pages from client time reporting

You might want to exclude particular pages from client time reporting from the browser.

About this task

You can add a parameter to the configuration file to stop the IBM HTTP Server Response Time module from injecting JavaScript into any file that matches the patterns you specify. This in turn stops client time reporting from the browser for those pages.

Procedure

To exclude particular pages from client time reporting from the browser, complete the following steps:

1. Open the following file in a text editor:

   - Linux AIX *install_dir*`/config/`*hostname*`_t5.cfg` where *install_dir* is `/opt/ibm/apm/agent`
   - Windows *install_dir*`\TMAITM6_x64\`*hostname*`_t5.cfg` where *install_dir* is `C:\IBM\APM`

2. In the **advconfig** section, add

   ```
   {KT5WEBPLUGIN_JSI_EXCLUDE_URI_WITH_PATTERNS=URL_path_pattern to_exclude}
   ```

   For example,

```
{KT5WEBPLUGIN_JSI_EXCLUDE_URI_WITH_PATTERNS=*/DoNotJSIMe.jsp,/absolutePath/index.jsp,/
skipThisDir/*}
```

The URL path is limited to 256 characters.

Tip: Use an asterisk (*) as a prefix or suffix to match patterns. Add multiple values separated by a comma to the parameter if required.

3. Save and close *hostname*_t5.cfg.
4. Restart the Response Time Monitoring agent:

```
rt-agent.sh stop
rt-agent.sh start
```

Using the IBM HTTP Server Response Time module as a non-root user

With some careful setup, you can use the IBM HTTP Server Response Time module with a user ID other than root. Remember that if you are using the Network Packet Analyzer, you must use the root user.

To use a user ID other than root, follow these guidelines.

For the Response Time Monitoring agent

Install the Response Time Monitoring agent by using the user ID you are going to run it with, to a directory that has write access.

- For this procedure, the Response Time Monitoring agent user ID is *agentuser*. The directory to which the agent is installed is *$AGENT_HOME*.
- If you install the Response Time Monitoring agent as root, and then run the agent as a different user, you will not be able to create files.

For the IBM HTTP Server Response Time module

ServerRoot must be owned by the same user ID that is used to run apache start|stop.

- During apache start, a wrt tracking directory is created under ServerRoot. Therefore, the user needs sufficient permissions to create files and directories under ServerRoot.
- If the IBM HTTP Server Response Time module user, *ihsuser* is different from *agentuser*, it requires write access to $AGENT_HOME/tmp.

  $AGENT_HOME/tmp is created during the installation of agents. The *ihsuser* needs permission to create a kt5 directory in $AGENT_HOME/tmp.
- There can be multiple versions of ServerRoot, each administered by different users.

For both the Response Time Monitoring agent and IBM HTTP Server Response Time module

Both *agentuser* and *ihsuser* need read/write access to the following directories:

- $AGENT_HOME/tmp/kt5
- ServerRoot/wrt

Normally the IBM HTTP Server Response Time module is started first and the wrt directories are created automatically by the Response Time Monitoring when it first starts up, with read/write access for all.

ServerRoot/wrt is also used by camconfig to push configuration. The *ihsuser* creates shared queue ID files that are picked up by the *agentuser*; the *agentuser* reads the queue ID from the directory and pushes configuration.

If root is used to run apache start initially and *ihsuser* is not root, complete the following steps:

1. Stop the Response Time Monitoring agent.
2. Using root, run apachectl stop.
3. Delete the following directories:

- $AGENT_HOME/tmp/kt5
- ServerRoot/wrt

4. Using *ihsuser*, run apachectl start to re-create the directories with the correct permissions.

For resetting the IBM HTTP Server Response Time module permission to the wrt directory

You can reset the permission of the IBM HTTP Server Response Time module wrt directory for more security. The update involves adding a parameter in the configuration file to limit permission to the wrt directory that was created by the non-root user during IBM HTTP Server installation.

Complete this procedure on the system where the IBM HTTP Server Response Time module is installed to change the permission of the wrt directory for the non-root user from 777 to 700:

1. Open IHS_HOME/conf/httpd.conf file in a text editor.
2. Add property **WrtDisableDirPermNonRoot** to the end of the file:
   - When the property is enabled, only the user ID that started the httpd and matches the user ID that created the directory is used to create the wrt directory with permission 700. All other users are denied access to work with this directory.
   - When the property is not enabled, the wrt directory is created with the default permission 777.
3. Restart the Response Time Monitoring agent:

```
rt-agent.sh stop
rt-agent.sh.start
```

The Response Time Monitoring configuration data and some persistent files are stored in the wrt directory, which is used during the communication process, and each connection process creates a wrt configuration file in the directory. After you add **WrtDisableDirPermNonRoot** to the httpd.conf file, only the limited certain user can communicate with Response Time Monitoring monitoring agent successfully.

Using load balancers

If you are using load balancers in your environment, some additional customization is required.

Procedure

If you are using a load balancer, follow these guidelines:

1. Turn off URL Rewrite on the load balancer.
2. Install a Response Time Monitoring agent on each web server that you want to monitor.

   Do not install Response Time Monitoring on the load balancer.

What to do next

If you are running the Response Time Monitoring agent behind a load balancer, you can configure the load balancer to forward the IP address of the client to optimize monitoring performance. Use the following steps as an example:

1. In the HTTP header, set the IP address of the client in the **X-Forwarded-For** field.
2. In the $AGENT_HOME/config/*hostname*_t5.cfg file, add {KT5WEBPLUGIN_OVERRIDE_SOURCE_ADDR_HEADERS=X-Forwarded-For} to the SECTION=advconfig section.

   Tip: Add multiple values to the parameter if required. For example, {KT5WEBPLUGIN_OVERRIDE_SOURCE_ADDR_HEADERS=x-forwarded-for, iv-remote-address}
3. Restart the Response Time Monitoring agent. Run the following commands:

```
rt-agent.sh stop
rt-agent.sh start
```

Limiting the CPU used to monitor IBM HTTP Server

In saturated environments, you might want to limit the percentage of CPU used by IBM HTTP Server instrumentation.

About this task

Specify the percentage of CPU that the IBM HTTP Server Response Time module can use. The percentage of CPU used is not limited by default. Configure the CPU percentage on the server where the agent is installed.

Procedure

To configure the percentage of CPU used, complete the following steps:

1. Reconfigure the agent interactively or manually:
   - Run the agent script for configuring interactively:

     Linux AIX

     ```
     $AGENT_HOME/bin/rt-agent.sh config
     ```

     Windows

     ```
     install_dir\BIN\rt-agent.bat config
     ```

   - Open the following file in a text editor:

     Linux AIX

     ```
     /opt/ibm/apm/agent/config/hostname_t5.cfg
     ```

     Windows

     ```
     C:\IBM\APM\TMAITM6_x64\hostname_T5.cfg
     ```

2. In the **advconfig** section, add the following parameter and set a value from 0 to 100:

   ```
   KT5WEBPLUGIN_TARGET_CPU_PERCENTAGE=10
   ```

   where the value you specify is the percentage limit of CPU use. The default value of *0* means that the CPU use is not limited.

3. You can also set the following parameters:

Option	Description
KT5WEBPLUGINCONFIGPOSTURL	List of URLs corresponding to an IBM HTTP Server installation. Default: `http://localhost/WrtUpdateConfig.dat`
KT5WEBPLUGIN_MAX_REQUESTS_PER_SECOND	Number of requests per second monitored by each IBM HTTP Server installation. If the number of requests exceeds this number, the subsequent requests are not monitored. If the limit is reached, JavaScript insertion stops, and no data is sent back to the Response Time Monitoring agent. Default: `0` (no maximum)

Option	Description
KT5WEBPLUGIN_CPUMAN_PERIOD_IN_SEC	The period, in seconds, at which CPU use is checked to determine whether the target has been exceeded. Default: 60 seconds
KT5WEBPLUGIN_CATCHUP_PERIOD_COUNT	Number of periods that are allowed at the same state before the CPU is scaled back. For example, by using the default, if the CPU use is high and 4 cycles later, it remains high, CPU use is scaled back. Default: 3

Results
The percentage CPU available to the IBM HTTP Server Response Time module is set at a fixed percentage.

Packet Analyzer roadmap

Use the Packet Analyzer to monitor HTTP transactions. You need to manually configure HTTPS monitoring. You need to manually instrument your web pages to collect browser timings.

To determine the environments in which you can use Packet Analyzer, see "Response Time Monitoring Components" on page 678 and "Planning the installation " on page 679.

Packet Analyzer is automatically enabled when you install the Response Time Monitoring agent, but there are a number of further steps and customizations that you might need to perform.

1. You can customize Packet Analyzer settings, for example, port number in the Agent Configuration window. For more information, see "Configuring Packet Analyzer using Agent Configuration window" on page 695.
2. To monitor HTTPS transactions, manually instrument your web pages to collect browser timings. For more information, see "Monitoring HTTPS transactions" on page 697.
3. To enable browser timings, add the JavaScript Injection component to your application and associate the JavaScript monitoring component with your application, for more information, see "Adding the JavaScript monitoring component to your application" on page 696.
4. If you are operating a high transaction load environment, there are some advanced tuning steps that you might need perform. For more information, see "Advanced Configuration of the Packet Analyzer" on page 702

Configuring Packet Analyzer using Agent Configuration window

You can use the Agent Configuration window to configure Packet Analyzer.

To monitor HTTP traffic for a particular system by using the Packet Analyzer, complete the following steps:

1. To access the Agent Configuration page, in the APM UI select **System Configuration** > **Agent Configuration**, then select the **Response Time** tab.
2. Select the system or systems that you want to update. Select multiple systems if you want to use the same HTTP settings for each of those systems.

   If the systems you select have different HTTP values set, `Multiple Values` or `Multiple Lists` is displayed instead of individual values. You cannot update systems with different values at the same time.
3. In the **Monitor HTTP traffic?** field, double-click the value and select **Yes** from the list.

4. In the **HTTP Ports to monitor** field, double-click the value and enter any additional ports that you want to monitor, other than the default port 80 and any other ports already listed.

   To stop monitoring a port, select the port that you no longer want to monitor and click **Remove**.

5. Click **Apply Changes**.

Adding the JavaScript monitoring component to your application

To help you understand the performance of your web pages in a browser and any errors, the Response Time Monitoring agent needs to be able to collect timing data from the browser. To enable this feature, you must configure the application that you want to monitor.

About this task

Before you can monitor interactions within your web pages, you need to add the JavaScript monitoring component to each web page for your application. The JavaScript monitoring component captures the state of each web page and associated JavaScript interactions. Add the JavaScript monitoring component to the application that you want to monitor. The relevant content and actions are automatically captured and sent to the Cloud APM server for analysis and correlation.

Procedure

Complete the following steps to enable collection of real user monitoring data from the browser. These steps need to be completed only once, unless the application configuration changes.

1. Add the JavaScript monitoring component to the application. The procedure that you use depends on the application type:

   a) For Java EE applications, extract *install_dir*/clienttime/ClientTime.war from the installation package to a directory accessible to the HTTP Server.

   b) For non-Java EE applications, such as Ruby, .NET, Python, and Node.js, save *install_dir*/clienttime/wrtInstrumentation.js from the installation package to a directory accessible to the HTTP Server.

   Extract the *install_dir*/clienttime/ClientTime.war file to a temporary path. You must copy the extracted wrtTimingTarget.dat file to the document root. Document root is a setting on the HTTP Server (Apache, IIS, and so on). It is a directory to store your documents. By default, all requests are taken from this directory but symbolic links and aliases might be used to point to other locations. For example, the document root for Apache is /opt/IBM/HTTPServer/htdocs.

   The wrtInstrumentation.js file can be placed in any directory. Ensure you update the path location to the wrtInstrumentation.js file in the HTML header.

2. Associate the JavaScript monitoring component with the application.

   This association can normally be done by modifying an application header script. Typically, only one header script needs to be modified for each component or application that is to be monitored.

   For both Java EE applications and non-Java EE applications, add the following JavaScript to the application header before any other JavaScript:

   ```
   <script language="JavaScript" src="path/wrtInstrumentation.js"
   type="text/JavaScript"></script>
   ```

   where *path* is the relative path to the JavaScript monitoring component

   For example:

   ```
   <script language="JavaScript" src="/ClientTime/wrtInstrumentation.js"
   type="text/JavaScript"></script>
   ```

Results

Pages that are instrumented with the JavaScript monitoring component are monitored, and data from the pages is analyzed and displayed in End User Transactions dashboards.

Enabling browser timing

By enabling resource timing monitoring, the Response Time Monitoring agent processes the W3C Resource Timing data by using the Packet Analyzer. With this function enabled, you can view detailed performance information on front-end elements.

About this task

Before you can monitor resource timing data, you must add the resource timing monitoring component to your application and associate the resource timing monitoring component with the application. The resource timing monitoring component automatically captures the state and interactions of the front-end elements, and sends the data to the Cloud APM server for analysis. The results of this analysis are displayed in the **Subtransactions** dashboard.

Procedure

Complete the following steps to enable the resource timing monitoring function. Complete these steps only once, unless the application configuration changes.

1. Add the resource timing monitoring component to the application.
   a) Extract *`install_dir`*`/clienttime/wrtInstrumentation.js` file from the installation package.
   b) Add the `wrtInstrumentation.js` file to the JavaScript directory of your application.
2. Append the following line to the application header:

```
<script> var wrt_enableResourceTiming=true; </script>
```

   For example,

```
<script language="JavaScript" src="path/wrtInstrumentation.js"
type="text/JavaScript"></script>
<script> var wrt_enableResourceTiming=true; </script>
```

Results

The pages are instrumented with the resource timing monitoring component. This component is enabled by default. The resource timing data on pages that are instrumented with the resource timing monitoring component is analyzed and displayed in the **Subtransactions** dashboards.

What to do next

If you want to disable the resource timing monitoring component, set the **`wrt_enableResourceTiming`** parameter to `false`.

Monitoring HTTPS transactions

Response Time Monitoring monitors HTTP transactions by default. To monitor HTTPS transactions, Response Time Monitoring requires access to the SSL Certificates so that it can decrypt SSL traffic from your local web servers.

Before you begin

Identify the HTTPS web servers that you want to monitor, including their IP addresses and configured ports. For example, `192.168.1.23`, port 443. For each HTTPS web server, check that Response Time Monitoring can read its ciphers. Response Time Monitoring supports the ciphers supported by IBM Java, including the following ciphers.

- RSA_WITH_RC4_40_MD5
- RSA_WITH_RC4_128_MD5
- RSA_WITH_RC4_128_SHA
- RSA_WITH_RC4_40_SHA

- RSA_WITH_DES40_CBC_SHA
- RSA_WITH_DESC_CBC_SHA
- RSA_WITH_3DES_EDE_CBC_SHA
- RSA_WITH_AES_128_CBC_SHA
- RSA_WITH_AES_256_CBC_SHA
- RSA_EXPORT1024_WITH_RC4_56_MD5
- RSA_EXPORT1024_WITH_RC2_CBC_56_MD5
- RSA_EXPORT1024_WITH_DES_CBC_SHA
- RSA_EXPORT1024_WITH_RC4_56_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256

Restriction: Response Time Monitoring cannot decrypt traffic that uses Diffie-Hellman key exchange.

Procedure

To enable HTTPS transaction monitoring, complete the following steps:

1. Set up the keystore. For more information, see "Setting up the keystore" on page 699.
2. Configure the Response Time Monitoring agent by running one of the following commands and providing values when prompted:

   For example:

```
Configuring Response Time Monitoring Agent
Edit 'Response Time Monitoring Agent' settings? [1=Yes,2=No](default is: 1): 1

Basic Configuration :   Specify basic monitoring configuration. Note: HTTP is
now configured centrally using the Response Time tab under Agent Configuration.

Specifies if HTTPS transactions should be monitored
Monitor HTTPS transactions [ 1=Yes, 2=No ] (default is:2): 1

This keystore contains the certificates of the HTTPS websites being monitored
HTTPS keystore (e.g. - /tmp/keys.kdb) (default is: ): /tmp/keys.kdb

This table maps HTTPS servers to the appropriate certificates (e.g. cert1,
server ip,server port; cert2,server2 ip,server2 port);...
HTTPS server certificate map (eg - certAlias,9.48.152.1,443;...)(default is: ):
 label1,10.0.0.1,9443;label1,9.185.150.71,443

Advanced Configuration :
Specify advanced monitoring configuration

The NIC card which has the selected IP address will be monitored.
IP address of the NIC to be monitored (default is: ): 10.0.0.1

Data Collection and Analysis Configuration :
Specify Configuration Information on how Data is Analyzed.

Configuration completed successfully.
Agent restart required to apply configuration changes.
```

   where:

   - HTTPS keystore is the keystore configured in step 1
   - HTTPS server certificate map, specify:
     - `label 1` - the key label configured in step 1
     - `server ip` - the IP address of the server, which must match the Source/Destination attribute in the IPV4 header of the packets
     - `server port` - server port number, which must match the Source/Destination port attribute in the TCP header of the packets

     Add multiple entries for multiple possibilities of the server IP of the same key label.

   - IP address of the NIC to be monitored, the interface that can see the packets and is mapped to eth0, en0, and so on. The name does not need to match any attributes of IPV4 or the TCP headers of the packets. If `10.0.0.1` corresponds to eth0, use `tcpdump -s0 -i eth0 ...` to see all the packets that need to be analyzed by the Packet Analyzer.

3. Restart the Response Time Monitoring agent.

Setting up the keystore

To monitor HTTPS transactions, import keys into the KT5Keystore for all web servers that you want to monitor.

About this task

You can either export the SSL certificates from the web servers that you are monitoring and import them into the HTTPS Keystore by using IBM Key Management (iKeyman), or specify the web server's keystore stash file (`.kdb`) in the HTTPS Keystore. When you install or configure Response Time Monitoring, you are prompted for the location of the `keys.kdb` file.

If you do not have keystore stash files (`.kdb` and `.sth`), check that the CMS Provider is enabled in your Java version so that you can use iKeyman to set up the key database:

1. Go to the *`install_dir`*`/ibm-jre/jre/lib/security` directory. For example:
   - Linux `/opt/ibm/apm/agent/JRE/lx8266/lib/security`
   - Windows `C:\Program Files\IBM\APM\ibm-jre\jre\lib\security`
2. In the `java.security` file, add the following statement to the list of security providers as shown, where *number* is the last sequence number in the list.

   ```
   security.provider.number=com.ibm.security.cmskeystore.CMSProvider
   ```

   The list of providers looks like the following example:

   ```
   ## List of providers and their preference orders #
   security.provider.1=com.ibm.jsse.IBMJSSEProvider
   security.provider.2=com.ibm.crypto.provider.IBMJCE
   security.provider.3=com.ibm.security.jgss.IBMJGSSProvider
   security.provider.4=com.ibm.security.cert.IBMCertPath
   security.provider.5=com.ibm.security.cmskeystore.CMSProvider
   ...
   #
   ```
3. Save and close the file.

Restriction: Response Time Monitoring cannot decrypt traffic by using Diffie-Hellman key exchange.

Procedure

To enable HTTPS transaction monitoring, collect the SSL certificates from the web servers that you want to monitor and import the certificates and keystore stash files into the HTTPS Keystore by using iKeyman. The following example uses iKeyman to export the certificates from an IBM HTTP Server, and import them to HTTPS Keystore:

1. Install a Response Time Monitoring agent on each HTTPS web server that you want to monitor.
2. Run **IBM Key Management** (iKeyman) from within the IBM Java `bin` directory by running one of the following commands, depending on your operating system.
   - AIX Linux `/opt/ibm/apm/agent/JRE/lx8266/bin/ikeyman`

     Note: You must have X-Window on the environment for iKeyman to work properly.
   - Windows `c:\IBM\APM\java\java80_x64\jre\bin\ikeyman`
3. Create a new Keystore database. In the **New** dialog box, complete the following steps:

   a) From the **Key database type** list, select **CMS**.

If CMS is not available in the list, the CMS Provider might not be enabled. Enable the CMS Provider in the Java security file.

b) In the **File Name** field, enter the name of the HTTPS Keystore file and click **OK**.
For example, `keys.kdb`.

4. In the **Password Prompt** dialog box, complete the following steps:

   a) In the **Password** and **Confirm Password** fields, enter and confirm the password to access `keys.kdb`.

   Do not set an expiration time unless you want to re-create the keystore database and restart the Response Time Monitoring agent periodically.

   b) Select **Stash the password to a file?** to store the password for `keys.`kdb in an encrypted form in a stash file, `keys.sth`.

   Note: The Response Time agent supports stashed password version 1 only. After APM 8.1.4, run following command to store the password for keys.kdb in an encrypted stash file, keys.sth.

   On Linux:

   ```
   cp keyfile.sth keyfile.sth.new-format
   ```

   ```
   cd /opt/IBM/ccm/agent/lx8266/gs/bin
   ```

   ```
   #export LD_LIBRARY_PATH=/opt/ibm/apm/agent/lx8266/gs/lib64:$LD_LIBRARY_PATH
   ```

   ```
   ./gsk8capicmd_64 -keydb -stashpw
   -db /opt/IBM/ccm/agent/keyfiles/keyfile.kdb -v1stash
   ```

   On Windows:

   ```
   copy server.sth server.sth.backup
   ```

   ```
   set PATH=c:\IBM\APM\GSK8_x64\lib64;%PATH%
   ```

   ```
   C:\IBM\APM\GSK8_x64\bin\gsk8capicmd_64 -keydb
   -stashpw -db .\server.kdb -pw passw0rd -v1stash
   ```

5. In the **Key database content** section of the iKeyman window, complete the following steps:

   a) Select **Personal Certificates**.

   b) Click **Import**.

   c) In the **Import Key** dialog box, from the **Keyfile type** list, select **CMS**.

   d) Browse to the keystore file and click **Open**, and then click **OK**.

   e) In the **Password Prompt** dialog box, enter the keystore password.

   f) Select the key from the list and click **OK**.

   g) In the **Change Labels** dialog box, select the key label name. In the **Enter a new label** field, specify the host name of the server and click **Apply**.

   Note: You will need this value when you configure Response Time Monitoring, so make a note of it.

   h) Click **OK**.

6. Save the HTTPS Keystore.

Importing keys from Internet Information Services

To extract keys from Internet Information Services and import them into the KT5Keystore, complete the following steps:

1. Install a Response Time Monitoring agent on each HTTPS web server that you want to monitor.
2. Export a .pfx file from Internet Information Services:

    a. From the Windows **Start** menu, select **Administrative Tools** > **Internet Information Services (IIS) Manager**.
    b. Select the web server and site whose private key you want to export, then right-click and select **Properties** from the context menu.
    c. Select the **Directory Security** tab, then select **Server Certificate** in the **Secure communications** section.
    d. In the **IIS Certificate Wizard**, click **Next**.
    e. Select **Export the current certificate to a .pfx file** and click **Next**.
    f. Enter the path and file name and click **Next**.
    g. Enter an export password for the key and click **Next**.
    h. Click **Next** on all subsequent pages, then click **Finish**.

3. Extract Personal and Signer Certificates from the .pfx file:

    a. Run **IBM Key Management** (iKeyman) from within the IBM Java `bin` directory using the command `c:\IBM\APM\java\java80_x64\jre\bin\ikeyman`. Ensure that the environment variable `JAVA_HOME` is set.
    b. In the Keystore database, select **File** > **Open**.
    c. From the **Key database type** list, select **PKCS12**.
    d. Enter the name and path for the .pfx file you created above, then click **OK**. When prompted, enter the password, then click **OK**.
    e. Select **Key Database Content** > **Personal Certificates**, then click **Export/Import**.
    f. Select an Action Type of **Export Key** and a Key File Type of **PKCS12**. Enter a file name and location for the exported key and click **OK**. When prompted, enter an export password, then click **OK** again.
    g. If the Personal Certificate was signed by a Certificate Authority, select **Key Database Content** > **Signer Certificates** and click **Extract**. Select the default file type, and enter a file name and location for the exported certificate, then click **OK**.

4. Extract Signer `.cer` files (if needed):

    a. If a Signer Certificates file was extracted from the .pfx file, navigate to the directory where it was saved, and make a new copy with the extension `.cer`. Double-click the new copy to open it using the Windows Certificate viewer.
    b. On the **Certification Path** tab, you can view the signer certificate chain. The lowest item in the chain should be the Personal Certificate. For all certificates above it, do the following:

        1) Select a certificate and click **View Certificate**.
        2) Select **Details** and click **Copy to File**.
        3) Accept all defaults in the Certificate Export Wizard and enter a filename with the `.cer` extension.

5. Create a new Keystore database. In the **New** dialog box, complete the following steps:

    a. From the **Key database type** list, select **CMS**, and enter a filename and location. When prompted, enter a password for the new keystore.

       Note: Ensure you select **Stash the password to a file**.

    b. If Signer Certificates were extracted from the .pfx file, do the following:

        1) Select **Key Database Content** > **Signer Certificates**.
        2) For each signer certificate, click **Add** and add the `.cer` file.

    c. Select **Key Database Content** > **Personal Certificates** and click **Import**.
    d. Select the key file type **PKCS12**, and the name and location of the **.p12** file. When prompted, enter the password.
    e. Save the keystore and exit the key management utility.

f. Copy the `.kdb` and `.sth` files to the KT5Keystore on the Response Time Monitoring appliance machine.

g. Place the IBM Key Management database files (`.kdb`) and stash (`.sth`) in a safe directory, and ensure that they are only readable by Administrator or `root` (or the user ID that was used to install the Response Time Monitoring agent).

Advanced Configuration of the Packet Analyzer

There are a number of advanced configuration options for the Packet Analyzer.

After you have configured the Packet Analyzer, there are a number of advanced configuration tasks you can perform to fine tune the performance and features.

Using load balancers

If you are using load balancers in your environment, some additional customization is required.

Procedure

If you are using a load balancer, follow these guidelines:

1. Turn off URL Rewrite on the load balancer.
2. Install a Response Time Monitoring agent on each web server that you want to monitor.

   Do not install Response Time Monitoring on the load balancer.
3. Add the JavaScript monitoring component to your application. For more information, see "Adding the JavaScript monitoring component to your application" on page 696.

What to do next

If you are running the Response Time Monitoring agent behind a load balancer, you can configure the load balancer to forward the IP address of the client to optimize monitoring performance. Use the following steps as an example:

1. In the HTTP header, set the IP address of the client in the **`X-Forwarded-For`** field.
2. Configure the Response Time Monitoring agent to use the IP address header of the client. Set the IP address header of the client in the **KFC_OVERRIDE_SOURCE_ADDR_HEADER** field in one of the following files, depending on your operating system:

   - AIX Linux `/opt/ibm/apm/agent/tmaitm6/wrm/kfcmenv`
   - Windows `C:\IBM\ITM\TMAITM6_x64\wrm\Analyzer\kfcmenv`

   For example:

   ```
   KFC_OVERRIDE_SOURCE_ADDR_HEADER=x-forwarded-for
   ```

   or if you are using WebSEAL:

   ```
   KFC_OVERRIDE_SOURCE_ADDR_HEADER=iv-remote-address
   ```

Configuring CPU overhead limit

If you are operating a high transaction load environment, you can limit the monitoring resources used by the Response Time Monitoring agent.

About this task

This function limits the CPU usage of the Response Time Monitoring agent by monitoring and reporting only a portion of the web traffic by using sampling. The `CPU overhead limit` is not configured by default. You must configure the `CPU overhead limit` on the server where the agent is installed.

Procedure

To configure `CPU overhead limit`, complete the following steps:

1. Open the following file in a text editor:

   Linux AIX `/opt/ibm/apm/agent/tmaitm6/wrm/kfcmenv`

   Windows `C:\IBM\ITM\TMAITM6_x64\wrm\Analyzer\kfcmenv`

2. Configure the values for the following parameters:

   KFC_MAX_PROTOCOL_PACKETRATE
   The initial maximum packet rate. For example, if you set the parameter as **KFC_MAX_PROTOCOL_PACKETRATE=2000**, the maximum packet rate is 2000 packets per second. This rate varies dynamically based on the value for **KFC_CPUTHROTTLE_TARGET** and the current CPU usage.

   KFC_CPUTHROTTLE_TARGET
   The percentage of the overall CPU resource that can be used by the `kfcmserver` process. For example, if you set the parameter as **KFC_CPUTHROTTLE_TARGET=10.0**, the `kfcmserver` process can use up to 10% of the overall CPU resource.

   Note: The value for parameter **KFC_CPUTHROTTLE_TARGET** is the percentage of the overall CPU resource that is available for the `kfcmserver` process. For example, if you have 4 CPU cores and **KFC_CPUTHROTTLE_TARGET** is set to 10, the Resource Monitor in Windows measures CPU resource as 400%. As a result, the `kfcmserver` process can use up to 40% of the total 400% CPU resources available.

Results

The CPU `overhead limit` is configured for the Response Time Monitoring agent.

Reconfiguring from IBM HTTP Server Response Time module to Packet Analyzer

You might want to change your monitoring environment from IBM HTTP Server Response Time module to Packet Analyzer.

Procedure

1. Uninstall the HTTP Server agent

   a) Edit `/etc/httpd/conf/httpd.conf` by commenting the Response Time plugin line. For example

   ```
   #include /opt/ibm/apm/agent/tmp/khu/khu.etc.httpd.conf.httpd.conf
   ```

   b) Uninstall the HTTP Server agent. For example

   ```
   /opt/ibm/apm/agent/bin/http_server-agent.sh uninstall
   ```

   c) Open a new command prompt window to clean the system variate before reconfiguring the Response Time agent in the next step.

2. Open the following file

   Linux AIX *install_dir*/config/*hostname*_t5.cfg
   Where *install_dir* is `/opt/ibm/apm/agent`

   Windows *install_dir*\TMAITM6_x64*hostname*_t5.cfg
   where *install_dir* is `C:\IBM\APM`

3. Set the parameters as follows:

   ```
   { KT5DISABLEANALYZER=NO } { KT5ENABLEWEBPLUGIN=NO }
   ```

4. Reconfigure the agent as follows:

   ```
   re-agent.bat config  install_dir\samples\rt_silent_config.txt
   ```

Customizing End User Transaction location values

You can customize the locations applied to specific IP addresses or address ranges in the End User Transaction dashboards for your particular environment.

Before you begin

Use the **Geolocation** tab in the **Agent Configuration** to customize location values.

Use this feature to set the location of IP addresses that are displayed in the dashboard as **Unknown**. These addresses might be internal addresses, for example `192.168.x.x` or `10.x.x.x`, or external IP addresses that are not resolved. You can also use this feature to override incorrect locations for IP addresses. For example, if you know that the IP address `9.1.1.1` is in Los Angeles, but it is shown as San Francisco, override the location, and set `9.1.1.1` to Los Angeles.

About this task

Customize location values in the End User Transaction dashboards by uploading a CSV file containing the values you require. You can find a sample CSV file on the **Geolocation** tab.

The CSV file must have the following values as a header. The values can be in any order, and the entries must match that order.

```
IP_ADDRESS, COUNTRY, REGION, CITY
```

For example,

```
IP_ADDRESS, COUNTRY, REGION, CITY
10.0.5.0/24, Australia, WA, Perth
10.1.0.6, Australia, VIC, Melbourne
```

You can specify a single IPv4 address, or a range. If you specify a range, ensure that you use a valid value in the range 1-32.

Procedure

To customize the location values displayed in the End User Transaction dashboards, complete the following steps in the Application Performance Dashboard.

1. Set up your CSV file or files, with IP addresses matched to locations.
2. Upload the CSV file.
   a) Go to **Agent Configuration** > **Geolocation**.
   b) Click **Upload CSV**, select the files you want to upload, and click **Open**.
      - Ensure that your CSV file lists general IP address ranges first, before more specific IP addresses.
      - Upload multiple files if required.
      - If the values in one file overlap those in another, the values in the newer file override the values in the first.
3. Expand **Upload Results** to check for errors. Check for the following problems:
   - Overrides
   - Invalid IP addresses
   - Invalid rows
   - Values longer than 250 characters

Results

Wait a few minutes and view your customized values in the End User Transaction dashboards.

What to do next

You can remove customized values if required. Complete one of the following steps:

- To remove some of your customized values, select the IP addresses that you want to remove, click **Clear Selected Entries**, and click **OK** to confirm the removal
- To remove all customized values, click **Clear All Entries**, and click **OK** to confirm the removal

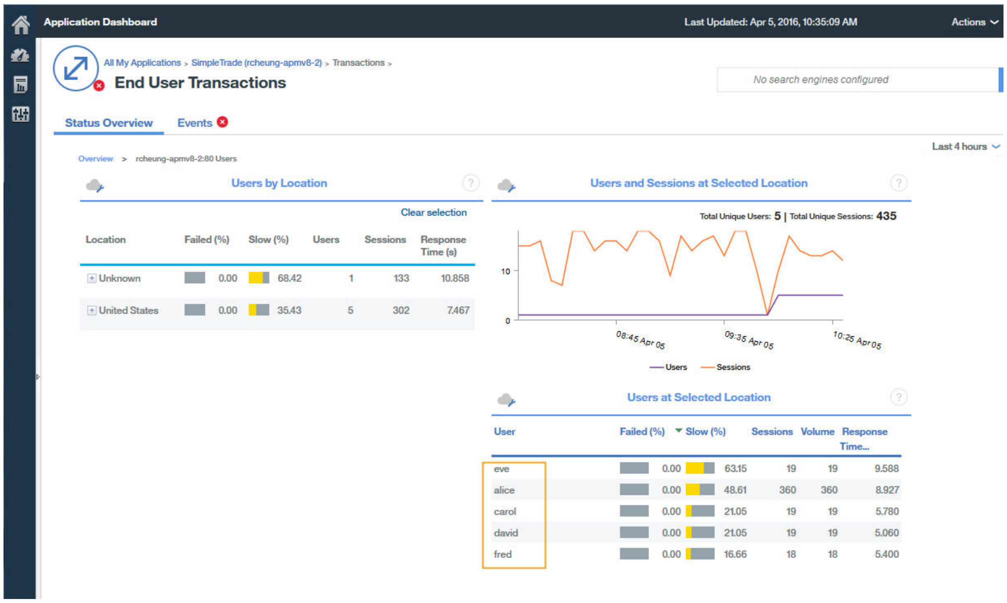
Tracking additional web applications

To track web applications in addition to those tracked by default, you must identify and configure user and session tracking methods.

Before you begin

If your application is not supported by the defaults, the dashboards do not contain user and session details, the username is displayed as `anonymous` or `unknown`, and no session information is available.

If user tracking methods are configured correctly, the user ID is extracted and is listed in the **End User Transactions** > **User Summary** > **Users at Selected Locations** dashboard.



Application Dashboard
Last Updated: Apr 5, 2016, 10:35:09 AM
Actions
All My Applications > SimpleTrade (rcheung-apmv8-2) > Transactions >
End User Transactions
No search engines configured
Status Overview
Events
Last 4 hours
Overview > rcheung-apmv8-2:80 Users
Users by Location
Clear selection
Users and Sessions at Selected Location
Total Unique Users: 5 | Total Unique Sessions: 435
10
0
08:45 Apr 05
09:35 Apr 05
10:25 Apr 05
Users
Sessions
Users at Selected Location

Location	Failed (%)	Slow (%)	Users	Sessions	Response Time (s)
Unknown	0.00	68.42	1	133	10.858
United States	0.00	35.43	5	302	7.467

User	Failed (%)	Slow (%)	Sessions	Volume	Response Time...
eve	0.00	63.15	19	19	9.588
alice	0.00	48.61	360	360	8.927
carol	0.00	21.05	19	19	5.780
david	0.00	21.05	19	19	5.060
fred	0.00	16.66	18	18	5.400

Note: The user tracking is based on session tracking. You need to set the correct session tracking methods variable first, in the Response Time agent configuration settings, for user and session tracking methods.

About this task

In IBM Application Performance Management V8.1.4 and later, you can use the **Agent Configuration** > **Response Time** page to add applications to track by either the Packet Analyzer or IBM HTTP Server Response Time module. Values defined on this page take precedence over the values in the `WRT_Defaults.xml` file.

To track additional applications, you must first identify the user ID and session ID methods and values for the application you want to monitor. For example:

1. Open the developer tool for your browser, so that you can see the requests for the application you want to monitor.
2. Select the last request in the browser network log, so that you can identify your test request easily.
3. Create a test request with parameters you will recognize. For example, log in to your website with `testuser`.
4. Select the test request and look at the `Headers`.

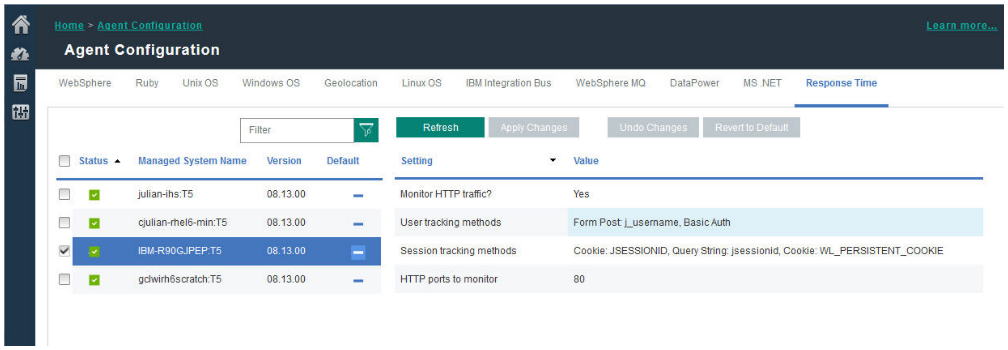
5. Identify the session ID from the request log. Session ID is typically specified in `cookie`, POST, `request/response header`, or `query string`. If the `cookie` is already defined in the default profile, you won't need to add it in step 2.
6. Identify the user ID from the request log. User ID can be specified in `cookie`, `request header`, POST, or `query string` content. For example, search for `testuser` which will give you the value for user ID.
7. Both **User tracking methods** and **Session tracking methods** must be updated with the correct value name of session and user in use in the customer's application code. How to identify the value name of Session and Username depends on the code of application. The following is the default value of User/Session setting in 8.1.4.

```
Session tracking methods=cookie\:JSESSIONID,querystring\:jsessionid,cookie\
       :WL_PERSISTENT_COOKIE
User tracking methods=formpost\:j_username,formpost\:uid,formpost\
       :ctl00%24MainContent%24uid,basicauth\:Authorization\: Basic
```

Procedure

After you have identified the user and session tracking methods and values used in your application, complete the following steps:

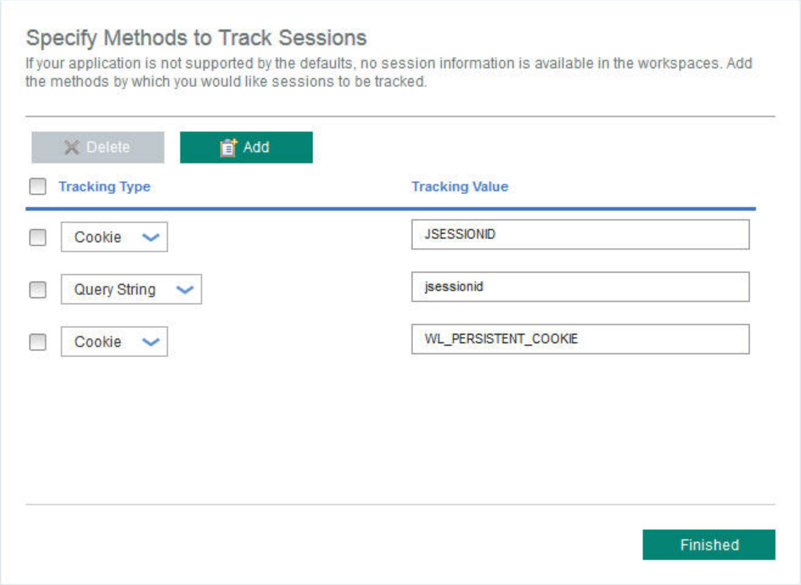
1. Go to the **Agent Configuration** > **Response Time** configuration page.



☐	Status	Managed System Name	Version	Default
☐	✓	julian-ihs:T5	08.13.00	–
☐	✓	cjulian-rhel6-min:T5	08.13.00	–
☑	✓	IBM-R90GJPEP:T5	08.13.00	–
☐	✓	gclwirh6scratch:T5	08.13.00	–

Setting	Value
Monitor HTTP traffic?	Yes
User tracking methods	Form Post: j_username, Basic Auth
Session tracking methods	Cookie: JSESSIONID, Query String: jsessionid, Cookie: WL_PERSISTENT_COOKIE
HTTP ports to monitor	80

2. Select the managed system that you want to update.
3. If required, update the session tracking methods:
   a) Click the value in the **Session tracking methods** field.


Specify Methods to Track Sessions

If your application is not supported by the defaults, no session information is available in the workspaces. Add the methods by which you would like sessions to be tracked.

☐	Tracking Type	Tracking Value
☐	Cookie	JSESSIONID
☐	Query String	jsessionid
☐	Cookie	WL_PERSISTENT_COOKIE

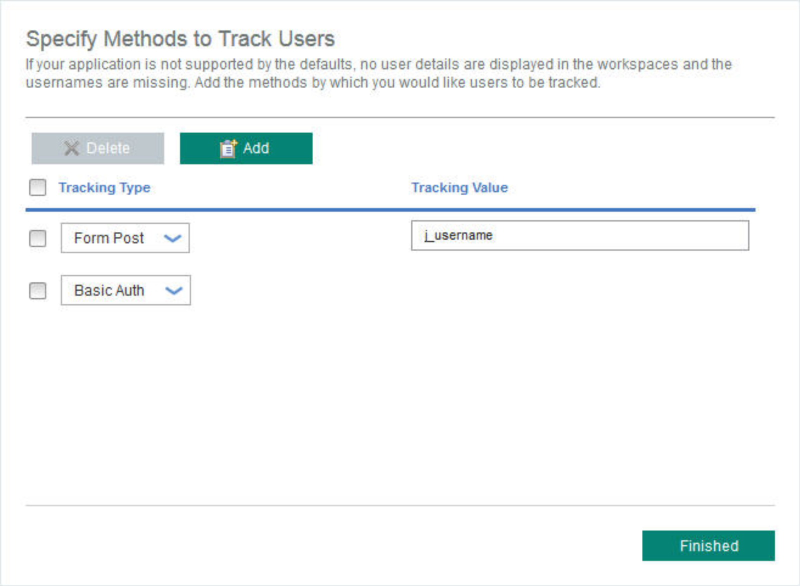
   b) In the **Specify Methods to Track Sessions** window, click **Add**.
   c) From the **Tracking Type** list, select the tracking type. For example, `Cookie`.

d) In the **Tracking Value** field, specify a value. For example, `WL_PERSISTENT_COOKIE`.

e) Click **Finished**.

4. If required, update the user tracking methods:

   a) Click the value in the **User tracking methods** field.



Specify Methods to Track Users

If your application is not supported by the defaults, no user details are displayed in the workspaces and the usernames are missing. Add the methods by which you would like users to be tracked.

□ Tracking Type	Tracking Value
□ Form Post	j_username
□ Basic Auth	

   b) In the **Specify Methods to Track Users** window, click **Add**.

   c) From the **Tracking Type** list, select the tracking type. For example, `Header`.

   d) In the **Tracking Value** field, specify a value. For example, `username`.

   e) Click **Finished**.

5. In the agent configuration page, click **Apply Changes**.

Results

Applications using the newly specified tracking methods are displayed in the Application Dashboard.

What to do next

Test that the user IDs and session information from your application are displayed in the Application Dashboard.

Specifying unique managed system name for the Response Time Monitoring agent

The Response Time Monitoring agent instance name displayed on the Cloud APM console is also known as the managed system name(MSN). You can use the agent configuration parameter to specify unique MSN for each agent instance.

About this task

The managed system name for the Response Time Monitoring agent is in the following format:

```
instancename:hostname:T5
```

T5 is the product code for the Response Time Monitoring agent.

Procedure

1. Stop all existing agent instances. If you do not have any existing agent instances, proceed to the next step. For more information about stopping agent instances, see “Using agent commands” on page 177.

2. Linux AIX Change **CTIRA_SUBSYSTEM_ID** in the `runagent` file. Normally all agent instances on a machine use the same hostname value.

   a) Make a backup copy of the file:

   ```
   Linux AIX install_dir/platform/t5/bin/runagent
   ```

   b) Edit the file. Add *newinstancename* on Linux or AIX systems.

   ```
   Linux AIX CTIRA_SUBSYSTEM_ID=newsubsystemid
   ```

3. Start existing instances of the agent.
4. Start the Cloud APM console. Modify your applications by removing the agent instances under the old MSNs and adding the new agent instances.

Configuring Ruby monitoring

You can monitor both on-premises and IBM Cloud Ruby applications. To monitor on-premises Ruby applications, configure the Ruby agent. To monitor IBM Cloud Ruby applications, configure the Ruby data collector.

About this task

The directions are for the most current release of the agent, except as indicated.

The following procedure is a roadmap for configuring the Ruby agent and the Ruby data collector, which includes both required and optional steps. Complete the configuration steps according to your needs.

Procedure

- To monitor on-premises Ruby applications, complete the following steps to configure the Ruby agent:
  a) Configure agent instances to monitor Ruby applications. See Configuring the Ruby agent to monitor Ruby applications.
  b) Install the data collector for monitoring data to display in Cloud APM console. See Installing the data collector.
  c) Optional: If you are a Cloud APM, Advanced user, you can complete the following tasks according to your needs:
    - To configure the data collector to collect diagnostics data, see Configuring the diagnostics data collector.
    - To enable method trace for requests and adjust the length for the file path parameter that is displayed in the Request Stack Trace widget, see Enabling method trace and adjusting the path display.
    - To increase the JVM heap size to avoid the `out of memory` error, see Increasing the JVM heap size.
    - To disable diagnostics, see Disabling or enabling diagnostics data for Ruby applications.
- To monitor IBM Cloud Ruby applications, complete the following tasks to configure the Ruby data collector:
  a) Configure the Ruby data collector for IBM Cloud applications. For instructions, see “Configuring the Ruby data collector for IBM Cloud applications” on page 715.
  b) Optional: To change the behavior of the Ruby data collector, see “Customizing the Ruby data collector for IBM Cloud applications” on page 716.

Configuring the Ruby agent

To have the Ruby agent monitor your applications, specify the Ruby run time. As a result, you use the run time to gather data from the Ruby applications and to configure the agent.

Before you begin

Determine the server that you use to start Ruby applications, and the qualified bin directory for the Ruby or Rake executable that the agent uses:

1. To determine the application server that you are using, run the following command:

   ```
   ps -ef | grep ruby
   ```

   You see the name of the server that is used to start your application. The possible server names are listed as follows:

   - `Passenger`
   - `Unicorn`
   - `Puma`
   - `Thin`

   If the command output does not indicate the server names that are shown in the preceding list, the server that you use to start the application might be WEBrick.

   Important: If you use multiple web servers to start your Ruby applications, you must create one agent instance for each application web server, for example, one instance for PUMA and one for Unicorn.

2. To determine the qualified bin directory for the Ruby or Rake executable that the Ruby agent uses, run the following command:

   ```
   which ruby
   ```

About this task

You can repeat this task to configure multiple agent instances according to your needs.

Procedure

1. To configure the agent, run the following command:
   `install_dir``/bin/ruby-agent.sh config` *`instance_name`*
   where *instance_name* is the name you want to give to the instance, and *install_dir* is the installation directory of the Ruby agent. The default installation directory is `/opt/ibm/apm/agent`.

   Important: Do not specify a long instance name. The total length of your host name and the agent instance name must not exceed 28 characters. If the length exceeds the limit, the Managed System Name is truncated, and the product code for the Ruby agent does not display correctly.

   The Managed System Name includes the instance name that you specify, for example, *`instance_name`*`:`*`host_name`*`:`*`pc`*, where *pc* is your two character product code for the agent. For example, if you specify Ruby2 as your instance name, your managed system name is `Ruby2:hostname:KM`, where *KM* is the two character product code for the Ruby agent.

2. When you are prompted with `Edit 'Monitoring Agent for Ruby' settings`, enter 1 to continue.
3. When you are prompted with `Fully Qualified Rubies Bin Directory`, specify the binary directory. for example, if you use Ruby Version Manager (RVM), enter `/usr/local/rvm/rubies/ruby-2.0.0-p247/bin`.
4. When prompted with `Auto Detect Ruby Applications Flag`, enter Y to continue. The agent receives the data that the agent data collector sends.
5. When you are prompted with `Application Server Process name`, press Enter to accept the default of `ruby`, or specify the value for the server that you use according to the following list:

- For WEBrick servers, accept the default or specify ruby; if Rails is installed by Ruby Stack, specify .ruby.bin.
- For Passenger servers, specify passenger.
- For Unicorn servers, specify unicorn.
- For Puma servers, specify puma.
- For Thin servers, if the applications are started by running the command thin start, accept the default to use ruby; if the applications are started by running the command thin start -d, specify thin; if Rails is installed by Ruby Stack and the applications are started by running the command thin start, specify .ruby.bin.

6. When you are prompted with Socket Data Source, press Enter to accept the default of 0 to use the ephemeral port.
7. When you are prompted with Edit 'Application' settings, enter 5 to exit the setting.
8. To start the agent, run the following command:
   install_dir/bin/ruby-agent.sh start *instance_name*

What to do next
Install the data collector for the Ruby agent to work properly and for the data to display in Cloud APM UI. For instructions, see Installing the data collector

Installing the data collector
You must install the data collector for the agent to work properly. After you install the data collector, monitoring data is displayed on the Application Performance Dashboard.

Before you begin
If you installed your Ruby on Rails application on a Linux system by using a non-root account, and you plan to collect diagnostics data, the non-root user must have access to the diagnostics data collector home directory. Verify that the non-root user has read and write access in the *install_dir*/install-images/kkm directory, where *install_dir* is the installation directory of the Ruby agent. The default installation directory is /opt/ibm/apm/agent. If required, provide read and write permissions by using the chmod 777 command.

Procedure

1. Stop your Ruby on Rails application.
2. Optional: If you are upgrading the Ruby data collector to a new version, you must first uninstall the old version data collector by running the following command:

   ```
   gem uninstall stacktracer
   ```

3. Install the diagnostics data collector. Enter gem install --local *install_dir*/lx8266/km/bin/stacktracer-*version*.gem, where *version* is the version number, and *install_dir* is the installation directory of the Ruby agent. The version number in the name of the stacktracer-*version*.gem file in the agent installation directory indicates the version number that you need to enter here. The default installation directory is /opt/ibm/apm/agent.

   Important: Install the data collector as the same user when you install and run the Ruby on Rails application.

4. Navigate to the home directory of your application, open its Gemfile, and add the following line to the end of the file: gem 'stacktracer', '*version*'
   where *version* is the version number of the data collector. The version number is indicated in the name of the stacktracer-*version*.gem file that is in the *install_dir* of the Ruby agent.
   For example, if you install the Ruby data collector Version 1.0 Fix Pack 8, you can find a stacktracer-01.00.08.00.gem file in the installation directory of you agent. And then you add the gem 'stacktracer', '01.00.08.00' line to your application to install the data collector.

Note: If there is only one version of `stacktracer` in the environment, add the `gem 'stacktracer'` line to the end of the file. Do not specify the version number in the line.

5. In the home directory of your application, enter `bundle install`.
6. Restart your Ruby on Rails application.

Results

The data collector is installed and configured and your Ruby on Rails application is started.

What to do next

- If you are not logged in, follow the instructions in “Starting the Cloud APM console” on page 957. Select **Performance** > **Application Performance Dashboard** to open the **All My Applications** dashboard, and drill down to the Ruby App resource monitoring dashboards and diagnostic dashboards to observe your Ruby on Rails applications from the status summary down to individual request instances.
- To see and modify the settings for the diagnostics data collector, continue to the next topic, “Configuring the diagnostics data collector” on page 711.
- To display method trace data for requests in Cloud APM UI, see Enabling method trace and adjusting the path display.
- When method trace is enabled or the data requests are large, you might receive `out of memory` errors. You can increase JVM heap size to avoid these errors. See Increasing the JVM heap size.
- You can disable and enable diagnostics data collection for one or more managed Ruby on Rails applications at any time through the Cloud APM console. See “Disabling or enabling diagnostics data for Ruby applications” on page 715. This function is not available for resource monitoring.

Configuring the diagnostics data collector

If you are a user of Cloud APM, Advanced, you can continue to configure the data collector for diagnostics data. Diagnostics data collection is disabled by default in the data collector configuration file.

Before you begin

You must have installed the diagnostics data collector and configured support for the collection of diagnostics data, as described in “Installing the data collector” on page 710.

About this task

The `instrumenter_settings.rb` configuration file appears after the agent registers a Ruby on Rails application's existence by properly configuring the `Gemfile`. This configuration file can be modified while the Ruby agent is running, and the changes are picked up automatically. Alternatively, you can apply the changes to all Ruby on Rails applications that are being monitored, which requires the applications to be stopped while you edit the configuration file.

Procedure

- To modify the data collector settings of a specific application that is running:
  1. Navigate to the `*install_dir*/install-images/kkm/dchome/*appClassName*/config` directory, where *appClassName* is the Ruby application class name, and *install_dir* is the installation directory of Ruby agent. The default installation directory is `/opt/ibm/apm/agent`.
  2. Open `instrumenter_settings.rb` in a text editor.
  3. Modify the data collector settings:

     :instrumentation_enabled

     To enable support for the collection of diagnostics data, set `:instrumentation_enabled => true`.

     To disable support for the collection of diagnostics data, set `:instrumentation_enabled => false`.

:sample_frequency

To modify the sampling frequency of requests, enter the number of requests between samplings.

The data collector collects diagnostic data only for sampled requests. If you set `:sample_frequency => 10`, for example, data is collected for 1 in every 10 requests.

:max_methods_to_instrument

To disable method trace collection or to enable method trace collection and limit the number of methods that are traced, set the value to zero, or enter the maximum number of methods to trace.

To disable the collection of method traces, set `:max_methods_to_instrument => 0`.

To enable the collection of method traces, set `:max_methods_to_instrument => 10000`. The value can be higher, but a much higher value might cause performance issues. When method data is collected, calls to methods are included in the Request Traces dashboard's Method Trace widget, which shows all the request instances and their nested requests.

:min_wallclock_to_include_in_trace

To modify the threshold that determines whether the request or method should be traced, set the minimum response time. If you set `:min_wallclock_to_include_in_trace => 0.001`, for example, only the requests and methods whose response times are longer than 1 millisecond are traced.

Remember: From the **Request Trace** diagnostic dashboard, you can drill down to a specific request instance from the Request Stack Trace group widget. The response times totals for the instance might be incorrect due to the filters set for `:min_wallclock_to_include_in_trace` and `:min_wallclock_to_include_stacks`, which can exclude some data.

:min_wallclock_to_include_stacks

To modify the threshold that determines whether the stacktrace information should be collected for a request or a method, set the minimum response time.

If you set `:min_wallclock_to_include_stacks => 0.1`, for example, the stacktrace information is collected for all the requests and methods whose response time is longer than 100 milliseconds.

:include_subclasses_of_these_modules

The Request Traces diagnostics dashboard helps you to identify the sequence of calls to nested requests and methods for a request instance. The data collector preemptively filters out methods from classes that are not included in the filter list. If operations that you want to trace are not included in the method stack traces, you can add them here.

To specify the methods to trace, add their class names.

Consider, for example, that you want to trace the Moped APIs in the following kind of Ruby code:

```
session = Moped::Session.new(['ip:27017'])
session.use(:HR)
session[:profiles].insert({......})
session[:profiles].find({...}).remove
```

Add the module names of these Moped APIs to the property:

```
:include_subclasses_of_these_modules => {"
ActionController" => true,
                     "ERB" => true,
                     "erb" => true,
                     "Arel" => true,
                     "Mongoid" => true,
                     "Moped" => true
                    },
```

Restriction: The method traces do not include class methods and private methods (methods defined in a class that have implicit or explicit "private" access specifiers).

:include_sql_text
To collect context data for methods, set this property to `true`.

:num_samples_per_file
To modify the maximum number of traced requests to store in each file, enter a value such as `:num_samples_per_file => 1000`. After the limit set here is reached, a new file is created.

Consider setting `:num_samples_per_file` to a lesser value if you adjust the configuration in a way that causes more data to be collected. For example, setting `:include_subclasses_of_these_modules` to trace more classes and methods can increase data collection. Setting any of the following properties to a lower value can also increase data collection: `:sample_frequency`, `:min_wallclock_to_include_in_trace`, and `:min_wallclock_to_include_stacks`.

:verbose_request_instrumentation
:verbose_class_instrumentation
:verbose_method_instrumentation
To increase the logging level of the diagnostics data collector, set these properties to `true`.

Tip: If operations that you specifically want to trace are not included in the method stacktraces, set `:verbose_class_instrumentation => true` and check the log to find out whether the class that you want to trace is instrumented. If it is not instrumented, add the class name of the module name of the class to the `:include_subclasses_of_these_modules` property.

4. If you edited any of the following properties, restart the corresponding Ruby on Rails application to have your changes take effect:

```
:include_subclasses_of_these_modules
:max_methods_to_instrument
```

The restart is necessary because these properties are used only when an application is launched to determine which class or method is to be instrumented by the Ruby data collector.

- To modify the data collector settings of all Ruby on Rails applications, complete these steps:

1. Stop any Ruby on Rails applications that are currently running.
2. Remove the `instrumer_settings.rb` from the `*install_dir*/install-images/kkm/dchome/*application_name*/config` directory.
3. Modify the data collector settings in `*Gem_dir*/gems/stacktracer-*version*/config/instrumenter_settings_template.rb` where *version* is the version number, such as 01.00.05.00, and *Gem_dir* is the installation directory of `stacktracer-version.gem`, such as `/usr/local/rvm/gems/ruby-2.1.4/`. For more information, see step “3” on page 711 in the procedure for modifying the data collector settings of a specific application.
4. Restart any Ruby on Rails applications that are currently running.

Results

The configuration of the diagnostics data collector has changed for the running application that you specified or for all applications.

Enabling method trace and adjusting the path display

IBM Cloud Application Performance Management, Advanced with diagnostic data enables users to have a **Request Traces** dashboard. If method data is collected, calls to methods are shown. The **Method Trace** widget displays request instances and their nested requests. You can enable method trace to include calls to methods in the nested requests. You can also adjust the configuration of the **Request Stack Trace** widget to show more than the default 50 characters of each file path.

About this task

Method trace is disabled by default. Complete the first procedure to enable method trace for display in the **Request Traces** dashboard. You can enable method trace by changing a setting in the configuration file.

Complete the second procedure to adjust the number of characters shown for the file path in the **Request Stack Trace** widget.

Procedure

- To enable method trace, edit the `instrumenter_settings.rb` settings:
  a) Locate the `instrumenter_settings.rb` file in the Ruby agent installation, for example, `*install_dir*/install-images/kkm/dchome/*appClassName*/config` where *appClassName* is the Ruby application class name, and *install_dir* is the installation directory of Ruby agent. The default installation directory is `/opt/ibm/apm/agent`.
  b) Open `instrumenter_settings.rb` in a text editor.
  c) Set the following property to 10000.

  ```
  max_method_to_instrument
  ```

  The value can be higher, but a much higher value might cause performance issues. (See also "Increasing the JVM heap size" on page 714.)
  d) Restart the Ruby on Rails applications to begin method data collection.

  For more information about all the `instrumenter_settings.rb` properties, see "Configuring the diagnostics data collector" on page 711.
- To adjust the file path display size in the **Request Stack Trace** widget, edit the `dfe.properties` file:
  a) Locate the `dfe.properties` file in the Ruby agent installation, for example, `*install_dir*/lx8266/km/bin/dfe.properties`
  where *install_dir* is the installation directory of Ruby agent. The default installation directory is `/opt/ibm/apm/agent`.
  b) Open `dfe.properties` in a text editor.
  c) Change the maximum size of the file path to display in each stack trace element by adjusting the value of the following property:

  ```
  dfe.stacktrace.filepath.maxsize
  ```

  d) Restart the Ruby agent.

Increasing the JVM heap size

When method trace is enabled for the Ruby Diagnostics **Request Traces** dashboard or data requests are very large, you can increase the JVM heap size to avoid out of memory errors.

About this task

The Ruby agent is an agent based on Java, and the default JVM heap size is 384 MB. Take these steps to increase the heap size and thus reduce the likelihood of the out of memory condition. The out of memory condition can occur from frequent large data requests and when you have method trace turned on.

Procedure

1. Find the JVM heap size setting in the Ruby agent installation directory, for example, `*install_dir*/lx8266/km/bin/runDeepDiveClient.sh`
   Where *install_dir* is the installation directory of Ruby agent. The default installation directory is `/opt/ibm/apm/agent`.

   The default value is `-Xmx384m`.
2. Increase the value, for example, to 1024 MB, shown as `-Xmx1024m`:
   ```
   export JAVA_OPT="-Djlog.common.dir=$CANDLEHOME/logs -DCONFIG_DIR=
   $DC_RUNTIME_DIR -Dkqe.cache.interval=60 -Xmx1024m -Dkqe.timespan=900 -
   Djlog.propertyFileDir.CYN=$CANDLEHOME/$ITM_BINARCH/$PRODUCT_CODE/bin"
   ```
3. Restart the Ruby agent.

Disabling or enabling diagnostics data for Ruby applications

If you have IBM Cloud Application Performance Management, Advanced, you can use the **Agent Configuration** page in the Cloud APM console to disable or enable diagnostics data collection at any time for one or more managed systems.

Before you begin

- You must have Cloud APM, Advanced in your environment.
- You must install and configure the Monitoring Agent for Ruby on a virtual machine, as described in "Installing agents" on page 125 on AIX systems or "Installing agents" on page 135 on Linux systems and in "Configuring Ruby monitoring" on page 708.
- You must install the diagnostics data collector and configure support for the collection of diagnostics, as described in "Installing the data collector" on page 710.

About this task

After you configure support for diagnostics data in the data collector configuration, the collection of diagnostics data is disabled by default for each managed system. To display data in the diagnostics dashboards, you must enable the collection of diagnostics data for each managed system you are monitoring.

Take these steps to enable and disable the collection of diagnostics data for each managed system:

Procedure

1. From the navigation bar, select **System Configuration->Agent Configuration**.

   The **Agent Configuration** page is displayed.
2. Click the **Ruby** tab.
3. Select the check boxes of the managed systems on which you want to disable or enable diagnostics data collection.
4. From the **Actions** list, select one of the following options to disable or enable diagnostics data collection for the selected managed systems:
   - Select **`Disable Data Collection`**. The status in the `Data Collector Enabled` column is updated to No for each of the selected managed systems.
   - Select **`Enable Data Collection`**. The status in the `Data Collector Enabled` column is updated to `Yes` for each of the selected managed systems.

Results

You configured the collection of diagnostics data for each of the selected managed systems.

Configuring the Ruby data collector for IBM Cloud applications

To collect information about Ruby applications on IBM Cloud, you must configure the Ruby data collector.

Before you begin

1. Download the data collector package from IBM Marketplace. For detailed instructions, see "Downloading your agents and data collectors" on page 105.

Procedure

1. Extract files from the data collector package. The `ruby_datacollector_8.1.4.0.tgz` package is included in the extracted directory.
2. Extract the files in `ruby_datacollector_8.1.4.0.tgz` by running the following command:

   ```
   tar -zxf ruby_datacollector_8.1.4.0.tgz
   ```

You get an `ibm_ruby_dc` folder.

3. Copy the entire `etc` folder in `ibm_ruby_dc` to the root folder of you Ruby application by running the following command:

   ```
   cp -r directory to the etc folder  home directory of your Ruby application
   ```

   The following command extract the data collector to the `/opt/ibm/ccm/ibm_ruby_dc/etc` directory and the home directory of your Ruby application is `/root/ruby_app/`:

   ```
   cp -r /opt/ibm/ccm/ibm_ruby_dc/etc /root/ruby_app/
   ```

4. Add the following section to the `Gemfile` in the home folder of your Ruby application:

   ```
   gem 'logger',  '>= 1.2.8'
   source 'https://maagemserver.ng.bluemix.net/' do
     gem 'ibm_resource_monitor'
     gem 'stacktracer'
   end
   ```

5. Run the `bundle lock` command to regenerate the `Gemfile.lock` file.
6. From the directory that contains the `manifest.yml` file of your Ruby application, run the following command:

   ```
   cf push
   ```

   Tip: For a sample `manifest.yml` file, see "Sample manifest.yml file" on page 188.

Results

The data collector is configured and is connected to the Cloud APM server.

What to do next

You can verify the monitoring data for your IBM Cloud application is displayed in the Cloud APM console. For instructions on how to start the Cloud APM console, see Starting the Cloud APM console. For information about using the Applications editor, see Managing applications.

Customizing the Ruby data collector for IBM Cloud applications

You can add environment variables in the IBM Cloud user interface (UI) to customize the monitoring for your IBM Cloud application. Use the following information to add the variables according to your needs.

User-defined environment variables for the Ruby data collector

You can use the information in the following table to customize Ruby monitoring on IBM Cloud.

Table 203. Supported user-defined environment variables for Ruby monitoring on IBM Cloud

Variable name	Importance	Value	Description
APM_BM_GATEWAY_URL	Optional	• https://*<server ip or hostname>*:443 • http://*<server ip or hostname>*:80	The target on-premises server gateway URL.

Table 203. Supported user-defined environment variables for Ruby monitoring on IBM Cloud (continued)

Variable name	Importance	Value	Description
APM_KEYFILE_PSWD	Optional	Encrypted password of the key file	The encrypted key file password that is paired with the key file. If you are a Linux user, you can use the `echo -n` *`<keyfile password>`* `|` `base64` command to encrypt your password. **Note:** Set this variable only when you configured the Gateway to use HTTPS.
APM_KEYFILE_URL	Optional	http://*<hosted http server>*:*<port>*/ keyfile.p12	The URL to download the key file. **Note:** Set this variable only when you configure the Gateway to use HTTPS.
kkm_instrumentation_enabled	Optional	• true • false	Enables or disables the collection of diagnostics data. true: If you set the value to `true`, diagnostics data are collected. false: If you set the value to `false`, diagnostics data are not collected. The default value is `true`.
kkm_max_methods_to_instrument	Optional	Maximum number of methods that are traced	The maximum number of methods that are traced. You can disable method trace by setting the value to `0`. By default, the value is `10000` and method trace is enabled. **Note:** It is recommended that you do not set the value higher than 10000. A value much higher than 10000 might decrease application execution efficiency.
kkm_sample_frequency	Optional	Sampling frequency of requests	The number of requests from which a sample request is taken, for example, if you set the value to `10`, monitoring data is collected for one in every 10 requests. The default value is `10`.

Table 203. Supported user-defined environment variables for Ruby monitoring on IBM Cloud (continued)

Variable name	Importance	Value	Description
kkm_min_wallclock_to_include_in_trace	Optional	Response time threshold for collecting method trace, in seconds	If the response time of a request instance exceeds the value of this variable, the data collector collects its method trace. If you set it to 0.001, for example, requests and methods whose response time is longer than 1 millisecond are traced. The default value is 0, which means method trace is enabled for all requests and methods.
kkm_min_wallclock_to_include_stacks	Optional	Response time threshold for collecting stack trace, in seconds	If the response time of a request instance exceeds the value of this variable, the data collector collects its stack trace. If you set it to 0.001, for example, requests and methods whose response time is longer than 1 millisecond are traced. The default value is 0, which means stack trace is enabled for all requests and methods.

Unconfiguring the Ruby data collector for IBM Cloud applications

If you do not need to monitor your Ruby environment or if you want to upgrade the Ruby data collector, you must first unconfigure previous settings for the Ruby data collector.

Procedure

1. Go to the application root folder.
2. Remove the following lines from the Gemfile in the home folder of your Ruby application:

```
gem 'logger',  '>= 1.2.8'
source 'https://maagemserver.ng.bluemix.net/' do
  gem 'ibm_resource_monitor'
  gem 'stacktracer'
end
```

3. Run the bundle lock command.
4. From the application home directory, run the following command to re-push the application to IBM Cloud for the changes to take effect.

```
cf push
```

Results

You have successfully unconfigured the Ruby data collector.

What to do next

After you unconfigure the data collector, the Cloud APM console continues to display the data collector in any applications that you added the data collector to. The Cloud APM console will show that no data is available for the application and will not indicate that the data collector is offline. For information about how to remove the data collector from applications and from resource groups, see “Removing data collectors from Cloud APM console” on page 188.

Configuring SAP monitoring

To monitor a SAP system, the Monitoring Agent for SAP Applications must connect to an application server in the system to be monitored so that the agent can access the Advanced Business Application Programming (ABAP) code that is provided with the product.

Before you begin

- Review the hardware and software prerequisites, see Software Product Compatibility Reports for SAP agent
- The SAP agent does not support non-unicode SAP systems.

About this task

The SAP agent is a multiple instance agent; you must create the first instance and start the agent manually.

- To configure the agent on Windows systems, you can use the **IBM Performance Management** window or the silent response file.
  - “Configuring the agent on Windows systems” on page 719
  - “Configuring the agent by using the silent response file” on page 721
- To configure the agent on Linux or AIX systems, you can run the script and respond to prompts, or use the silent response file.
  - “Configuring the agent on Linux or AIX systems” on page 720
  - “Configuring the agent by using the silent response file” on page 721

After you install the SAP agent, you can import the Advanced Business Application Programming (ABAP) transport on the SAP system to support data collection in the SAP system. For more information, see “Importing the ABAP transport on the SAP system” on page 727.

After you configure the SAP agent, you must verify the agent configuration. For more information, see “Verifying agent configuration” on page 734.

After you configure the SAP agent, you can add Database Communication Port number that is required for OSLC (Open Source Lifecycle Collaboration) compliance. For more information, see “Adding Database Communication Port number” on page 738.

To delete the ABAP transport from the SAP system, you must import delete transport to the SAP system. For more information, see “Deleting the ABAP transport from the SAP system” on page 733.

The new CCMS design is enabled by default. Entry is present in the database table/IBMMON/ITM_CNGF for isnewccmsdesign parameter whose value is set to YES.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the “Change history” on page 54.

Configuring the agent on Windows systems

You can configure the SAP agent on Windows systems by using the **IBM Performance Management** window so that the agent can collect data of the SAP Applications Server that is being monitored.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.

2. In the **IBM Performance Management** window, right-click **Template** under the **Task/SubSystem** column, and click **Configure Using Defaults**.

   The **Monitoring Agent for SAP Applications** window opens.
3. In the **Enter a unique instance name** field, type an agent instance name and click **OK**.

   Important: The agent instance name must match the 3-digit system identifier (SID) of the managed SAP Applications Server. For example, if the SID of the managed SAP Applications Server is PS1, enter PS1 as the instance name.
4. Configure the SAP agent in the Application Server mode or the Logon Group mode.
   - Complete the following steps to configure the SAP agent in the Application Server mode:
     a. In the **Connection Mode** field, select **Application Server Mode** and click **Next**.
     b. In the **Specify Application Server Information** area, specify values for the configuration parameters and click **Next**.
     c. In the **Specify Logon Information to the SAP System** area, specify values for the configuration parameters and click **OK**.

     For information about the configuration parameters, see "Configuration parameters of the agent" on page 722
   - Complete the following steps to configure the SAP agent in the Logon Group mode:
     a. In the **Connection Mode** field, select **Logon Group Mode** and click **Next**.
     b. In the **Specify Logon Group Information** area, specify values for the configuration parameters and click **Next**.
     c. In the **Specify Logon Information to the SAP System** area, specify values for the configuration parameters and click **OK**.

     For information about the configuration parameters, see "Configuration parameters of the agent" on page 722

   Important: For the Application Server mode, it is mandatory to configure the Dialog Instance having dispatcher on the SAP system where the message server or ASCS is configured. For the Logon Group mode, it is not mandatory to configure the Dialog Instance having dispatcher on the SAP system where the message server or ASCS is configured.
5. In the **IBM Performance Management** window, right-click the agent instance that you created and click **Start**.

   Important: If you want to create another instance of the SAP agent, repeat Steps 1 to 6. Use a unique system identifier for each SAP agent instance that you want to create.

What to do next

- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.
- You must edit the predefined `R3_Alert_Crit` situation and `R3_Alert_Warn` situation to set the condition of Alert Status attribute as `Alert Status!= DONE` so that these situations do not get triggered for closed CCMS alerts.

Configuring the agent on Linux or AIX systems

You can configure the SAP agent on Linux or AIX systems so that the agent can collect data of the SAP Applications Server that is being monitored.

Procedure

1. On the command line, change the path to the agent installation directory.
   Example: `/opt/ibm/apm/agent/bin`
2. Run the following command where instance_name is the name that you want to give to the instance:

```
./sap-agent.sh config instance_name
```

Important: The agent instance name must match the 3-digit system identifier (SID) of the managed SAP Applications Server. For example, if the SID of the managed SAP Applications Server is PS1, enter PS1 as the instance name.

3. When the command line displays the following message, type 1 and press Enter: `Edit ‘Monitoring Agent for SAP Applications’ setting? [1=Yes, 2=No]`
4. Configure the SAP agent by using the Application Server mode or the Logon Group mode.
   - Complete the following steps to configure the SAP agent in the Application Server mode:
     a. When the command line displays the following message, type 1 and press Enter: `Connection Mode [ 1=Application Server Mode, 2=Logon Group Mode ]`
     b. Specify values for the configuration parameters.

     For information about the configuration parameters, see “Configuration parameters of the agent” on page 722
   - Complete the following steps to configure the SAP agent in the Logon Group mode:
     a. When the command line displays the following message, type 2 and press Enter: `Connection Mode [ 1=Application Server Mode, 2=Logon Group Mode ]`
     b. Specify values for the configuration parameters.

     For information about the configuration parameters, see “Configuration parameters of the agent” on page 722

   Important: For the Application Server mode, it is mandatory to configure the Dialog Instance having dispatcher on the SAP system where the message server or ASCS is configured. For the Logon Group mode, it is not mandatory to configure the Dialog Instance having dispatcher on the SAP system where the message server or ASCS is configured.
5. Run the following command to start the SAP agent:

```
./sap-agent.sh start instance_name
```

Important: If you want to create another instance of the SAP agent, repeat Steps 1 to 5. Use a unique System Identifier for each SAP agent instance that you create.

What to do next

- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see “Starting the Cloud APM console” on page 957.
- You must edit the predefined `R3_Alert_Crit` situation and `R3_Alert_Warn` situation to set the condition of Alert Status attribute as `Alert Status!= DONE` so that these situations do not get triggered for closed CCMS alerts.

Configuring the agent by using the silent response file

You can configure the SAP agent on Windows, Linux, or AIX systems by using the silent response file.

Procedure

1. In a text editor, open the `sap_silent_config.txt` file that is available at the *install_dir* `\samples` path, and specify values for all the configuration parameters.

   Windows `C:\IBM\APM\samples`

   Linux AIX `/opt/ibm/apm/agent/samples`

   For information about the configuration parameters, see “Configuration parameters of the agent” on page 722
2. On the command line, change the path to the bin directory:

Windows *install_dir*`\BIN`

Linux AIX *install_dir*`\bin`

3. Run the following command:

   Windows `sap-agent.bat config` *instance_name* *install_dir*`\samples\sap_silent_config.txt`

   Linux AIX `sap-agent.sh config` *instance_name* *install_dir*`\samples\sap_silent_config.txt`

   Important: The agent instance name must match the 3-digit system identifier (SID) of the managed SAP Applications Server. For example, if the SID of the managed SAP Applications Server is PS1, enter PS1 as the instance name.

4. Start the agent.

   Windows In the **IBM Performance Management** window, right-click the agent instance that you created, and click **Start**.

   Linux AIX Run the following command: `./sap-agent.sh start` *instance_name*

   Important: If you want to create another instance of the SAP agent, repeat Steps 1 to 4. Use a unique System Identifier for each SAP agent instance that you create.

What to do next

- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.
- You must edit the predefined `R3_Alert_Crit` situation and `R3_Alert_Warn` situation to set the condition of Alert Status attribute as `Alert Status!= DONE` so that these situations do not get triggered for closed CCMS alerts.

Configuration parameters of the agent

When you configure the SAP agent, you can change the default value of the parameters, such as the SAP hostname and the SAP system number.

The following table contains detailed descriptions of configuration parameters of the SAP agent.

Table 204. Names and descriptions of configuration parameters of the SAP agent

Parameter name	Description	Mandatory field	Examples
SAP Hostname (Primary)	The host name of the SAP application server to which the agent connects. If your SAP servers communicate over a private LAN, the computers that host the servers have two or more network cards. For the host name, enter a name by which the application server can be reached from external systems, such as the SAPGUI logon. Do not use the private LAN host name. The default value is the host name where the agent is installed.	Yes	`saphost.domain.com`
SAP System Number (Primary)	The two-digit SAP system or instance number that is used for connecting to a SAP host server. The default value is 00.	Yes	
SAP Hostname (Alternate 1)	The second choice for the host name if the primary host is unavailable.	No	

Table 204. Names and descriptions of configuration parameters of the SAP agent (continued)

Parameter name	Description	Mandatory field	Examples
SAP System Number (Alternate 1)	The system number for the host name of the first alternate.	No	
SAP Hostname (Alternate 2)	The third choice for the host name if both the SAP Hostname (Primary) and SAP Hostname (Alternate 1) hosts are unavailable.	No	
SAP System Number (Alternate 2)	The system number for the host name of the second alternate.	No	
SAP Client Number	The SAP client number for the RFC logon to SAP. The default value is 000. If the IBMMON_AGENT user that is generated by ABAP is used, enter the client number that was specified in the transport import. This number is the same as the nnn client number under the profile.	Yes	
SAP User Id	The SAP user ID for the RFC logon to SAP. The default value is IBMMON_AGENT, which is the predefined user ID that is created during the import.	Yes	
SAP User Password	Use the default password or specify a different password.	Yes	
Confirm SAP User Password	The password that is specified in the **SAP User Password** field.	Yes	

Table 204. Names and descriptions of configuration parameters of the SAP agent (continued)			
Parameter name	**Description**	**Mandatory field**	**Examples**
SAP Language Code	The language code that indicates the language that the agent uses when it connects to the SAP system. The specified language determines the language in which you see SAP information, such as alert messages, syslog messages, and job log messages. All SAP systems are delivered in English and German. If you require a different language, confirm with your SAP administrator that the language is installed on the SAP system. If you specify a language that is not supported, the agent cannot connect to the SAP system. The following languages and codes are supported: • CS - Czech • EN - English • FR - French • DE - German • HU - Hungarian • IT - Italian • ES - Spanish • JA - Japanese • KO - Korean • PL - Polish • PT - Portuguese • RU - Russian • ZH - Chinese • ZF - Traditional Chinese	Yes	
RFC Trace	The Remote Function Call (RFC) trace setting for the *SAPTRACE* variable. When you select this check box, you activate the RFC tracing and the default value is no RFC tracing. For the command line, 2 = No trace and 1 = Do trace. Because the RFC tracing generates extensive diagnostic information, use it carefully. For more information about the RFC tracing, contact IBM support.	No	
SAP Logon Group	The name of the SAP Server Logon group.	Yes	
SAP Message Server Name	The host name of the SAP message server.	Yes	

Table 204. Names and descriptions of configuration parameters of the SAP agent (continued)

Parameter name	**Description**	**Mandatory field**	**Examples**
SAP Message Service	The name of the service where the SAP message server is located. You must include service names in the following operating system services files: • `/etc/services` • `\windows\system32\drivers\etc\services`	Yes	You might use the message service name sapmsTV1, or the full message service port number 3601.
SAP Route String	Specify the SAP router string if you want access to the SAP server with a SAP router.	No	The router string `/H/host/H/` must be in the following format: `/H/beagle/H/brittany/H/` or `/H/amsaix11.tivlab.raleigh.ibm.com/W/tivoli/H/amsaix25`
SNC	Specify whether you want to enable or disable Secure Network Communications (SNC). Default value is disabled.	Yes	**`sap_conn.sap_snc_mode`**`=true` or `false`
SNC Security Level	The security level of SNC.	Yes	**`sap_snc_mode1.sap_snc_qop`**=*QOP value*. Default value is 8.
Client or Agent SNC Name	The SNC name of the client or agent.	Yes	**`sap_snc_mode1.sap_snc_client`**= *Client SNC Name*
Partner or SAP Server SNC Name	The SNC name of the partner or SAP Server.	Yes	**`sap_snc_mode1.sap_snc_server`**= *Server SNC Name*
SAP Cryptolibrary Path	The path of SAP Cryptolibrary.	Yes	**`sap_snc_mode1.sap_snc_library`**= *Crypto library path*

Configuring local environment variables

You can configure local environment variables to change the behavior of the SAP agent.

Procedure

1. On Windows systems, click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, from the **Actions** menu, click **Advanced > Edit ENV File**.

3. On Linux or UNIX systems, go to the command line and edit the `.sa.environment` file from the `<candle_home>/config` directory. Where, `candle_home` is the agent installation directory.

   Note: The `.sa.environment` file is a hidden file.

4. In the environment variable file, enter values for the environment variables.

   For information about the environment variables that you can configure, see "Local environment variables" on page 726.

Local environment variables

You can change the behavior of the SAP agent by configuring the local environment variables.

Variables for limiting the alerts to be cached by SAP Agent

To limit the alerts to be cached by SAP agent, use the following environment variables:

- **`MAX_CCMS_ALERT_THRESHOLD`**: To set the number of ccms alerts that the SAP agent will cache, use the *MAX_CCMS_ALERT_THRESHOLD* variable. The default value is 1000.
- **`MAX_MAI_ALERT_THRESHOLD`**: To set the number of mai alerts of each type that the SAP agent will cache, use the *MAX_MAI_ALERT_THRESHOLD* variable. The default value is 500.

Variable to set the threshold for long running jobs

To set the threshold for long running jobs, use the following environment variables:

- **`LONG_RUNNING_JOB_THRESHOLD`**: To set the threshold for duration (in hours) of long running jobs, use *LONG_RUNNING_JOB_THRESHOLD* variable. It is used for calculating the count of long running jobs greater than the threshold. Minimum value that can be set is 4 and maximum is 24. Default value of this parameter is 5.

Variables to delete idx file at SAP Agent restart and SAP system restart

To delete `IDX` file at SAP Agent restart and SAP system restart, use the following environment variables:

- **`DELETE_IDXFILE_AT_AGENT_RESTART`**: This parameter enables the feature of `IDX` file deletion when SAP Agent is restarted. `IDX` file stores the timestamp. It is recommended to set the parameter value to `N`. When set to `N`, it collects all alerts from the timestamp stored in `IDX` file. If you want to collect all alerts after the SAP agent restart, set the value to `Y`.
- **`DELETE_IDXFILE_AT_SAP_RESTART`**: This parameter enables the feature of `IDX` file deletion when SAP System is restarted. It is recommended to set the parameter value to `N`. When set to `N`, it collects all alerts from the timestamp stored in `IDX` file. If you want to collect all alerts after the SAP system restart, set the value to `Y`.

Variable to replace invalid characters with blank characters

To replace the invalid characters with blank characters, use the following environment variables:

- **`SAP_BLANK_INVALID_CHARACTERS`**: This parameter enables the feature of replacing any invalid characters with blank characters. If it is set to `Y`, then SAP agent will replace invalid characters with blank characters. If its set to `N`, then SAP agent will not replace invalid characters with blank characters. Default value of this parameter is `N`. It is recommended to set as `Y` for UTF languages.

SAP hostname is trimmed according to Managed System Name length limit

The `Managed System Name` of any resources that is published on APM console is limited to 32 characters. The SAP agent supports the trimming of domain name to form the `Managed System Name` within the limit.

Scenario 1

The `Managed System Name` for sub-node of **Sys** type has the following format:

SID-DBHOST:**Sys**
Where:

- The *SID* is the SAP system ID.
- The *DBHOST* is the SAP system hostname.

Example:
Given the *SID* is **P27** and the *DBHOST* is **VPT02F90.mycorporation.co.in**, the fully qualified domain name (FQDN) of the `Managed System Name` that is formed would be **P27-VPT02F90.mycorporation.co.in:Sys**.

When the `Managed System Name` has more than 32 characters, SAP agent trims the domain name to form the Managed System Name **P27-VPT02F90:Sys**. The trimmed `Managed System Name` of the sub-node will be published on the APM console.

Note: If the `Managed System Name` length that includes the domain name is less than or equal to 32-characters, the FQDN of the `Managed System Name` will not be trimmed. The FQDN of the `Managed System Name` is published on the APM console accordingly.
The trimming of the domain name, when necessary to meet the `Managed System Name` length limit is applicable to all sub-node types published by SAP agent.

Scenario 2

The `Managed System Name` for the agent instance sub-node **mySAP** has the following format:

$SAPSYSTEMNAME-$dbhost:*$CTIRA_HOSTNAME*:**mySAP**
Where:

- The *$SAPSYSTEMNAME* is the agent instance name provided during configuration.
- The *$dbhost* is the SAP system hostname.
- The *$CTIRA_HOSTNAME* is the agent machine hostname.

Example:
Given the *$SAPSYSTEMNAME* is **SA2**, *$dbhost* is **VPT02F90. mycorporation.co**, and the *$CTIRA_HOSTNAME* is **mysap1-v27.mycorp.co**, the fully qualified domain name (FQDN) of the `Managed System Name` that is formed would be **SA2-VPT02F90.mycorporation.co.in: mysap1-v27:mySAP**.

Note: The default domain name of the agent machine hostname is trimmed and it will be **mysap1-v27**.

The `Managed System Name` has more than 32 characters. Firstly, the SAP agent trims the domain name of the SAP system hostname to form **SA2-VPT02F90:mysap1-v27:mySAP**.
If the resultant `Managed System Name` remains exceeding 32 characters, the SAP agent will trim the trailing character(s) of the agent machine hostname to form the `Managed System Name` within 32 characters limit. Then, the `Managed System Name` of the sub-node is published on the APM console.

Importing the ABAP transport on the SAP system

You can install one SAP agent for each SAP system where you import the Advanced Business Application Programming (ABAP) transport request to support data collection in the SAP system.

Before you begin

Before you import ABAP transport on the SAP system, ensure that the following prerequisites are met:

- To import the product transport request, R3trans Version 01.07.04, or later is required because the Dynpro and Export and Import tables are incompatible. The basic operation of the agent is not affected by the Dynpro or Export and Import incompatibility issues; only the SAP configuration windows are affected.
- You must ensure that you import the SAP agent V7.1.1 transport on the client where the MAI configuration is available to monitor the Solution Manager System. To view features of the PI system, import the SAP agent V7.1.1 transport on the PI system on a client where PI configuration is available.

- To view data in the group widgets that are under SLM subnode, you must complete the MAI configurations for PI and Solution Manager. You must also configure business process monitoring so that you can view data in the BPM Alerts group widget. To view data for the Latest Critical and High Priority Alerts group widget, make the following configurations:
  - In Solution Manager 7.1, run SOLMAN_SETUP transaction and select **System Monitoring**, activate or enable the third-party component, and add **Implementation: BADI Definition for Alert Reactions** and third-party connector.
  - Set the scope filter to **All Alerts and Metrics**.
  - Ensure that the Implementation state is **Active**.

  For more information, see the following Online Service System (OSS) Notes, which include a list of required SAP service pack levels:

  - OSS Note 454321
  - OSS Note 330267
  - OSS Note 743155
- To monitor the SAP systems, the SAP agent needs the SAP statistics data. On SAP 7.0 systems, you must set the SAP system time zone to match the time zone for the operating system so that SAP statistics are collected with the correct time stamps. Similarly, update the SAP system time zone for the SAP agent so that the agent can collect data. For more information about this issue, see SAP Note 926290.

About this task

For information about importing the SAP transport, see "Importing the SAP transport" on page 729.

MAI Alert related prerequisites for importing the ABAP transport

You must verify the MAI Alert related prerequisites before you import the ABAP transport.

Configuration settings in the `transport.prop` file

When you use the new MAI Alert fetching mechanism that includes fetching MAI Alerts without configuring email notification settings and without BAdi implementation, then you must modify the following configuration setting in the `transport.prop` file.

Add the SPLEVEL=X line, where X is the support pack (SP) level of the Solution Manager system.

For example, if the System ID is S10 and the support pack level is 13, then add SPLEVEL=13.

Important: For the SAP system with SP level 10, or later, the value of the Technical Name (MEA) attribute is not populated on the Latest MAI Alerts with Rating 'Red' group widget in the SAP Solution Manager Dashboard when the MAI Alerts are fetched without configuring email notification in the SAP Solution Manager and without BAdi implementation. The value of the Technical Name (MEA) attribute is populated on the Latest MAI Alerts with Rating 'Red' group widget in the SAP Solution Manager Dashboard when the MAI Alerts are fetched by configuring email notification in the SAP Solution Manager and BAdi implementation.

Determination of old and new mechanism for fetching MAI Alerts based on the Solution Manager Support Pack (SP) Level

Old MAI Alert fetching mechanism

This mechanism is based on configuring email notification settings and the /IBMMON/ITM_IMPL_ALRTINBX BAdi implementation with the IF_ALERT_DYN_COFIGURATION interface to collect MAI Alerts and send them to the SAP agent.

New MAI Alert fetching mechanism

This mechanism is based on fetching MAI Alerts without configuring email notification settings and without the /IBMMON/ITM_IMPL_ALRTINBX BAdi implementation with the IF_ALERT_DYN_COFIGURATION interface.

You can use the following table to understand the usage of the `transport.prop` file and its dependency on the configuration of email notification settings.

Table 205. Usage of transport.prop file and its dependencies

SAP system SP Level	**transport.prop settings**		**Configuration of email notification settings**	**MAI Alert mechanism to be used**
	MAI_ CONFIGURED	**Solution Manager SP level**		
Any	No or file does not exist	Not Applicable	Configured or not configured	The SLM subnode does not appear instead the SOL subnode appears.
SP 6 through 9	Yes	Mentioned	Configured	Old mechanism
SP 6 through 9	Yes	Not mentioned	Configured	Old mechanism
SP 6 through 9	Yes	Not mentioned	Not configured	Old mechanism does not work because the configuration of email notification settings is mandatory.
SP 6 through 9	Yes	Mentioned	Not configured	Old mechanism does not work because the configuration of email notification settings is mandatory.
SP 10, or later	Yes	Mentioned	Configured	New mechanism
SP 10, or later	Yes	Mentioned	Not configured	New mechanism
SP 10, or later	Yes	Not mentioned	Configured	Old mechanism
SP 10, or later	Yes	Not mentioned	Not configured	Old mechanism does not work because the configuration of email notification settings is mandatory.

Importing the SAP transport

The SAP agent provides a set of Advanced Business Application Programming (ABAP) routines to support data collection in the SAP system. This ABAP code is delivered as an SAP transport that must be installed on each SAP system that is to be monitored. Your SAP administrator installs the transport.

About this task

The **ZITM_610AUTH** authorization profile and **ZITM_610AUT** authorization role are valid until the 6.1 release only. From release 6.2 or later, the **/IBMMON/AUTH** authorization profile is used. To protect against unauthorized use, the ABAP code that is installed in the SAP system is not visible from within the SAP system. In addition, this code cannot be modified or generated. You must obtain the support for this code from the IBM software support website.

In addition to installing ABAP code, the transport also installs translated language text elements to provide multicultural support for SAP transport text elements.

Important: Before you import the transport into the SAP system, you must not start the SAP agent instance that is configured to monitor the SAP system.

When you import the SAP transport, users get implicitly defined in the SAP system.

Use this procedure to import the SAP transport into the SAP system.

Procedure

1. Copy the IBM Tivoli Monitoring transport file from the following paths on the computer where the agent is installed.
   - For Windows: `*install_dir*\TMAITM6_x64\ABAP`
   - For Non-Windows: *install_dir*/*intrp*/sa/ABAP, where *intrp* must be **lx8266** or **aix526**.
2. Copy the following transport files from the paths that are mentioned in step 1 into the SAP environment:
   - K711_00xxxU.ITM and R711_00xxxU.ITM

     These files are Unicode versions of the transport. They contain the SAP agent ABAP code and Unicode support for text strings for Latin code pages and double-byte code pages.
   - K711_00xxx_DELETE.ITM and R711_00xxx_DELETE.ITM

     These files remove the ABAP code. The DELETE transport does not need to be imported, unless you stop the use of product entirely and want to remove the transports from the SAP systems. See "Deleting the ABAP transport from the SAP system" on page 733
3. Copy your transport files to the SAP Transport System data directory as follows, and do not change the transport file name:

   Unicode transport

   a. Copy the K711_00xxxU.ITM file to the cofiles directory

   b. Copy the R711_00xxxU.ITM file to the data directory.
4. To install the single IBM Tivoli Monitoring transport file on the SAP system, select one of the following file import options:
   - For the SAP system that is a Solution Manager 7.1 Service Pack 6 level, or later and has MAI configured, you must create the transport.prop file in the usr/sap/SID/DVEBMGS*instancenumber*/work work directory of the SAP system. If the SAP system is a distributed system with ABAP SAP Central Services (ASCS), create the transport.prop file in the Central Instance (CI) usr/sap/SID directory. Then, add MAI_CONFIGURED = YES entry in that file. This entry creates a MAI_CONFIGURED = YES entry in the /IBMMON/ITM_CNFG table. You can now import the single IBM Tivoli Monitoring transport file on the SAP system.

     Note: Before you import the single IBM Tivoli Monitoring transport file, you must create the transport.prop file in the usr/sap/SID/DVEBMGS*instancenumber*/work work directory of the SAP system and add MAI_CONFIGURED = YES entry in that file. You must not edit the entry in the /IBMMON/ITM_CNFG table.
   - For all other SAP systems with basis version equal to 7.0, or later and Solution Manager V7.1 without MAI configuration, you must directly import the single IBM Tivoli Monitoring transport file.
5. Run the following command to import SAP transport:

   ```
   tp addtobuffer ITMK711_00xxxU SID
   pf=\usr\sap\trans\bin\PROFILE_NAME
   ```

   Where:

   SID

   Target SAP system ID.

PROFILE_NAME

Name of the tp profile file. Make sure that the current tp parameter file is specified when you import the agent transport files from the command line. The tp parameter file is typically named `TP_DOMAIN_`*`SID`*`.PFL`. This file name is case sensitive on UNIX systems.

nnn

Number for the target client where the agent runs and for which the user ID, IBMMON_AGENT, authorization profile, and `/IBMMON/AUTH`, are defined.

Alternately, you can use the SAP STMS transaction to import the ITMK711_00xxxU.ITM transport requests. Ensure that the following options are selected in the **Import Options** tab of the **Import Transport Request** window.

- **Leave Transport Request in Queue for Later Import**
- **Import Transport Request Again**
- **Overwrite Originals**
- **Overwrite Objects in Unconfirmed Repairs**

For the SAP Basis version, if the **Ignore Invalid Component Version** option is enabled, ensure that it is selected.

Results

Depending on your SAP release level, when you run the **`tp import`** command, you might receive return code 4, which does not indicate a problem. Receiving return code 4 is an expected result from the **`import`** command.

Users and authorizations required by the SAP agent

To safeguard against unauthorized access to the SAP system, you can assign authorizations to a user who logs in to the SAP system. These authorizations define the access levels for a user in the SAP system.

After you import the ABAP transport, the SAP agent creates the default user ID as IBMMON_AGENT in the SAP system with the default password as ITMMYSAP. This user is a system user and the /IBMMON/AUTH authorization profile is associated with the user. The /IBMMON/AUTH profile and the IBMMON_AGENT user are created after ABAP transport is imported. With the /IBMMON/AUTH profile, the IBMMON_AGENT user can access transactions that are required to read performance data from the SAP system. Some examples of transactions that are used are as follows:

- CCMS alerts and administration
- Authorization for PI/XI message monitoring
- Solution Manager authorizations

You can create any other system type user for the agent. The user must be assigned the /IBMMON/AUTH profile.

To view and access data of SAP components, ensure that the user that is created for the agent has all the authorizations that are specified in the following table:

Table 206. The list of authorizations

Components	Authorization objects	Authorization description
General system authorizations that include the following components: • SAP Instance • SAP System	S_ADMI_FCD	To access the SAP system
	S_BDS_DS -BC-SRV-KPR-BDS	To access the document set
	S_BTCH_JOB	To run operations on the background jobs
	S_CCM_RECV	To transfer the central system repository data
	S_C_FUNCT	To make C kernel function calls in the ABAP programs
	S_DATASET	To access files
	S_RFC	To check RFC access. The S_RFC authorization object contains the following two sub-authorizations: • RFC1: To provide the authorizations for the RFC1 function group. • SDIFRUNTIME: To provide the authorizations for the SDIFRUNTIME function group.
	S_RFCACL	To check authorization for RFC users
	S_RZL_ADM	To access Computing Center Management System (CCMS) for R/3 System administration
	S_TCODE	To check authorizations for starting the transactions that are defined for an application
	S_TOOLS_EX	To display external statistics records in monitoring tools
Authorizations for PI that include the SAP Process Integration	S_XMB_MONI	To access XI message monitoring

Table 206. The list of authorizations (continued)

Components	Authorization objects	Authorization description
Authorizations for MAI that include the SAP Solution Manager	AI_DIAGE2E	To restrict E2E Diagnostics functions
	AI_LMDB_OB	To access Landscape Management Database (LMDB) objects
	SM_MOAL_TC	To control the access to the alerting and monitoring functionality in SAP Solution Manager
	SM_WC_VIEW	To restrict access to specific UI elements in work centers of the Solution Manager
	S_RFC_ADM	To control rights for administering RFC destinations
	S_RS_AUTH	To specify analysis authorizations within a role
	SM_APPTYPE	To access Solution Manager app type
	SM_APP_ID	To access applications provided in work centers

Deleting the ABAP transport from the SAP system

If you choose to remove the SAP agent from your system, you must import delete transport to the SAP system. Delete transport deletes the SAP agent dictionary objects and function modules.

Before you begin

Before you delete the transport from the SAP system, you must stop the SAP agent instance that is configured to monitor the SAP system.

If the SAP system is version 7.20 or later, before you import the delete transport, in your transport profile, you must add the following transport profile parameter: **`tadirdeletions=true`**. This transport profile parameter is available in tp version 375.57.68 and also in the R3trans version 6.14 release 700 or higher. For more information about removing transport requests from the SAP system, see Deleting transport requests.

Procedure

1. Go to the following path:
   - For Windows: *`install_dir`*`\TMAITM6_x64\ABAP`
   - For Non-Windows: *`install_dir`*`/`*`intrp`*`/sa/`ABAP, where *intrp* must be **`lx8266`**or **`aix526`**.
2. Copy the transport files into the SAP environment.
3. Copy the K711_00xxx_DELETE and R711_00xxx_DELETE files to the SAP Transport System data directory as follows:
   a) Copy the K711_00xxx_DELETE file to the `cofiles` directory.
   b) Copy the R711_00xxx_DELETE file to the `data` directory.
4. Run the following commands to import the delete transport:
   a) **`tp addtobuffer ITMK711_00xxx_DELETE SID pf=\usr\sap\trans\bin\`***PROFILE_NAME*

b) **tp import ITMK711_00xxx_DELETE SID client=nnn U16 pf=\usr\sap\trans\bin\** *PROFILE_NAME* where:

SID
Target SAP system ID.

PROFILE_NAME
Name of the tp profile file.

nnn
Number for the target client where the agent is to run.

Verifying agent configuration

After you install the SAP agent, you must verify the agent configuration by downloading, copying, and verifying the NetWeaver RFC SDK V7.20 library. You must also verify the configuration of Solution Manager V7.1 with MAI_Monitoring, verify MAI Alerts, and verify the configuration setting specific to third-party component.

Verify the agent configuration by completing the following procedures:

- "Downloading the NetWeaver RFC SDK V7.20 library" on page 734
- "Copying the NetWeaver RFC SDK V7.20 library in SAP agent setup" on page 735
- "Verifying the NetWeaver RFC SDK V7.20 library" on page 735
- "Verifying the configuration of Solution Manager V7.1 with MAI-Monitoring" on page 736
- "Verifying MAI Alerts" on page 737
- "Verifying configuration settings specific to third-party component" on page 737

Downloading the NetWeaver RFC SDK V7.20 library

Download the NetWeaver RFC SDK V7.20 library after you finish installing the SAP agent. All the files that are related to the NetWeaver RFC SDK V7.20 library are available for download from the SAP website.

Procedure

1. Log in to SAP Marketplace by using the following URL:
   http://service.sap.com
2. Click **SAP Support Portal**.
3. Enter your Service Marketplace user name and password.
4. Click **Software Downloads** and expand the **Support Packages and Patches** link.
5. Click **Browse our Download Catalog**, and then click **Additional Components**.
6. Click **SAP NetWeaver RFC SDK**, and then click **SAP NetWeaver RFC SDK 7.20**.
7. Select the operating system where you have the SAP agent.
8. Download the `*.`SAR file on your computer.
9. To extract the SAP `Netweaver RFC SDK *.`SAR file by using the SAPCAR utility that is provided by SAP, run the following command:
   `sapcar -xvf` *`SAP NetWeaver RFC SDK File Name`*`.SAR`

   Note: You can download the SAPCAR utility from the SAP website.
10. Navigate to the `lib` folder inside the extracted folder.

What to do next

Copy the NetWeaver RFC SDK V7.20 library in to the SAP agent setup.

Copying the NetWeaver RFC SDK V7.20 library in SAP agent setup

The NetWeaver RFC SDK V7.20 library contains files that you must manually copy in the SAP agent setup location.

Procedure

1. Navigate to the directory where you downloaded the NetWeaver RFC SDK V7.20 library.
2. Copy the files to the SAP agent setup location.
   - For Windows 64-bit operating systems you must copy the following files:
     - `icuin34.dll`
     - `libicudecnumber.dll`
     - `libsapucum.dll`
     - `icudt34.dll`
     - `icuuc34.dll`
     - `sapnwrfc.dll`

     You must copy the files to `install_dir\TMAITM6_x64` location.
   - For operating systems other than Windows, you must copy the files to the `install_dir/` *`intrp`*`/sa/lib` location, where *intrp* is the operating system code (aix526, li6263, sol606). You must copy the following files:
     - `libsapnwrfc.so`
     - `ibicudecnumber.so`
     - `ibicuuc34.a`
     - `libicui18n34.a`
     - `libicudata34.a`
     - `libsapucum.so`

What to do next

Verify the version of the NetWeaver RFC SDK V7.20 library that is downloaded.

Verifying the NetWeaver RFC SDK V7.20 library

You must verify the version of the file after you copy the extracted file.

Procedure

- Windows To verify the version of the file, complete the following steps:
  a) Right-click `sapnwrfc.dll` and click **Properties**.
  b) Click the **Version** tab.
  c) In the **Product Version** section, ensure that you have the following version: 720, patch 514, changelist 1448293, or later.
- Linux AIX To verify the version of the file, complete the following steps:
  a) Go to the `lib` folder in the extracted `*.`SAR file.
  b) Run the following command: **`strings libsapnwrfc.so | grep SAPFileVersion`**
  c) You must see the following message: `[root@IBMSAP2V6 lib]# strings libsapnwrfc.so | grep SAPFileVersion GetSAPFileVersion #[%]SAPFileVersion: 7200, 514, 22, 6206 .GetSAPFileVersion`

  Note: The message shows that this library has the version 720 patch 514, or later.

Verifying the configuration of Solution Manager V7.1 with MAI-Monitoring

To receive data for MAI Alerts, you must verify whether the Solution Manager V7.1 is configured correctly.

About this task

You can use Solution Manager V7.1 with MAI-Monitoring and Alerting Infrastructure to monitor the Managed Systems. Solution Manager V7.1 monitors itself and the satellite systems. Each satellite system has a plug-in and diagnostics agents. Diagnostics agents fetch the data for Host or Operating System level. Each host can have multiple diagnostics agents for different Solution Managers monitoring the host. Following are the keywords that are used in Solution Manager MAI Monitoring:

- Metrics: Data from the satellite systems.
- Alerts: Notifications that are based on some crossovers of threshold values that can be configured.
- Incident: Alerts that are converted into tickets and assigned to any user.

To verify the configuration of Solution Manager V7.1 with MAI monitoring, you must verify the basic settings, global level settings, and template level settings.

Procedure

1. To verify the basic settings, enter the Transaction Code: `SOLMAN_SETUP` and click **Enter**.

   Ensure that all the LEDs are green in the following tabs:

   - Overview
   - Basic Configuration
   - Managed System Configuration

   Note: There are different categories of Managed Systems such as Technical Systems, Technical Scenarios, Host, Database, Instance, PI Domain, Technical Component, and Connection. You must configure these Managed Systems according to business requirements. The MAI Alerts are based on the Managed Systems that you configured.
2. Enter the Transaction code: SE38 and click **Enter**.
3. Provide the program name as `RTCCTOOL` and run the report.

   Ensure that all the LEDs are green in the output.
4. To verify the global level settings, enter the Transaction code: `SOLMAN_WORKCENTER` and click **Enter**.

   Ensure that all the LEDs are green in the following tabs:

   - Overview
   - Configure Infrastructure
   - Pre-requisites
   - Configure
5. Verify whether the **Global Settings** for **Notification** status is **Active**.
6. To verify the template level settings, enter the Transaction Code: `SOLMAN_SETUP` and click **Enter**.

   In **Technical Settings**, in the **Auto-Notifications** list, ensure **Active** is selected.

   Note: For initial troubleshooting, ensure that email notifications are active.
7. For MAI system monitoring, verify the configuration of End-User Experience Monitoring (EEM) by using the following steps:

   a) Enter the Transaction code: SE37 and press **Enter**.

   b) Enter **`AI_EEM_LIST_ALL_SCENARIOS`** in the **Function Module name** field and press F8.

      There must be an entry for End-User Experience Monitoring (EEM).

Verifying MAI Alerts

To ensure that Solution Manager MAI is configured correctly for monitoring the MAI Alert Inbox in Technical Monitoring, you must verify that you receive MAI Alerts as output.

Procedure

1. Enter the Transaction code `SOLMAN_WORKCENTER` and click **Enter**. Check whether you can view MAI Alerts in the Solution Manager MAI Alert Inbox under Technical Monitoring.
2. Check for BAdi implementation by using the following steps:
   a) Enter the Transaction code: SE19 and click **Enter**
   b) Enter `/IBMMON/ITM_IMPL_ALRTINBX` in the **Enhancement Implementation** field.
   c) Click **Display** and check if BAdi implementation is active in **Runtime Behavior** section.
3. Check whether the database `/IBMMON/ITM_ALIX` contains MAI Alerts by using the following steps:
   a) Enter the Transaction code: SE16 and press **Enter**.
   b) In the **Table Name** field, enter `/IBMMON/ITM_ALIX` and run it. Ensure that you are receiving MAI Alerts in the table.
4. Enter the Transaction code: SE37 and click **Enter**.
5. In the **Function Module Name** field, enter `/IBMMON/ITM_MAIALRT_INX` and press F8.

   You must see MAI Alerts as output.

What to do next

If you are not able to view MAI Alerts in the `/IBMMON/ITM_ALIX` database, you must verify the settings in the Third-Party Component.

Verifying configuration settings specific to third-party component

If you are not able to view MAI Alerts, then you must verify the settings in the third-party component.

Procedure

1. Verify that Third-Party Component is active.
2. Verify that in **OS Adapter**, under **BAdi Implementation**, **Alert Reaction** is available. If **Alert Reaction** is not available, remove the default settings, and select the **BAdi implementation - Alert Reaction**.
3. Check the template settings by using the following steps:
   a) Verify the settings that are used to transfer specific alerts to the Third-Party System such as SAP ABAP 7.0.0.
   b) Select **Expert Mode**, select **Alerts**, and then click **Third Party Component**.

      Ensure that you are able to view the Alert Reaction BAdi name.

      Note: Ensure that the latest SAP notes are implemented. For Solution Manager V7.1 Service Pack 8, check if the following notes are implemented:

      - https://service.sap.com/sap/support/notes/1959978
      - https://service.sap.com/sap/support/notes/1820727
4. If you are not able to view MAI Alerts in the `/IBMMON/ITM_MAIALRT_INX` database, you must run the following Solution Manager MAI configurations steps for Third-Party Component:
   a) Enter the Transaction code: `SOLMAN_SETUP` and click **Enter**.
   b) In **Technical Monitoring**, select **System Monitoring**.
   c) Click **Configure Infrastructure** tab and then click **Default Settings** tab.
   d) Click **Third Party Components** tab and then click **Edit**.
   e) Select **Active** from the list.
   f) Ensure that scope filter is set as **All alerts, Events and Metrics (with Internal Events)** for the selected connector.

Note: OS Command Adapter is also one of the methods to push data to the third-party connector. To configure the OS Command Adapter, read the configuration detail settings in the How-to guide for OS Command Adapter.

Adding Database Communication Port number

A Database Communication Port number is essential to uniquely identify the database entity in the integrated scenarios. To achieve inter-component collaboration, the components of SCM AI included OSLC (Open Source Lifecycle Collaboration). In OSLC compliance, it is essential to identify the collaborating components uniquely. Therefore, the Database Communication Port number is important.

About this task

When you import the relevant IBM Tivoli Monitoring transport into the SAP System, the `/IBMMON/ITM_PORT` database table is created automatically. The table contains the following database fields:

- System ID
- System Hostname
- DB (Database) Communication Port #

Procedure

To add SAP Database Communication Port number for the SAP agent that is required for OSLC compliance, do the following steps:

1. Go to the SE16 Transaction Code, and press Enter.
2. In the **Database table name** field, enter `/IBMMON/ITM_PORT`, and press F7.
3. When the selection screen of the **/IBMMON/ITM_PORT** database table appears, press F8.

   The **/IBMMON/ITM_PORT** database table contains the following three database fields:

   - System ID
   - System Hostname
   - DB Communication Port #

   Note: The SAP systems that appear in the `/IBMMON/ITM_PORT` database table are for both the Java and ABAP architectures.
4. In the **DB Communication Port #** field, enter the relevant SAP Database Communication Port number for the respective SAP System ID and SAP System Hostname, and save the changes.

   Note: If you do not enter any value in the **DB Communication Port #** field in the `/IBMMON/ITM_PORT` database table, then, by default the DB Communication Port number is 0.

Advanced installation and configuration of the SAP agent

These are advance installation and configurations which are SAP agent specific.

The following installation and configuration topics are described:

- “SAP function module” on page 739
- “SAP user IDs” on page 739
- Utilities for the SAP agent
- “SAP RFC connections” on page 739
- “Test Connection feature” on page 749
- “Optional advanced configuration in SAP” on page 742
- “CEN CCMS reporting” on page 748
- “Uninstalling the Advanced Business Application Programming (ABAP) transport from the SAP system” on page 749

Note: The advance installation and configuration of the SAP agent contains references to IBM Tivoli Monitoring so as to make the documentation compatible with ABAP transport custom transaction code UI.

SAP function module

When the data volume is high on the SAP server, you might experience problems with certain widgets causing a slow response time from the server. If the widgets are not critical, you can disable the associated SAP function module.

By default, the SAP agent function modules are enabled. However, the following function modules are disabled by default:

- HTTP services under the SYS subnode (`/IBMMON/ITM_HTTP_SRVS`)
- XML messages under the PI/XI subnode (`/IBMMON/ITM_SXMB_MONI_NEW`)
- Sync/Async communication under the PI/XI subnode (`/IBMMON/ITM_SYN_ASYN_COMM`)
- qRFC inbound queue details under the Sys subnode (`/IBMMON/ITM_QIN_QDETAILS`)

After disabling the SAP function module, if you select a widget, data isn't displayed on the IBM Application Performance Management dashboard. Therefore, you avoid any performance-related problems.

SAP user IDs

This section provides information about SAP user IDs and permissions required by the SAP agent.

User IDs support the following purposes:

- "SAP RFC connections" on page 739
- "Basic agent monitoring" on page 739

SAP RFC connections

The SAP agent uses Remote Function Calls (RFC) connections for internal Centralized Computing Center Management (CCMS) polling and CCMS alert data collection. This behavior is specific to the SAP RFC architecture.

The SAP agent opens one dedicated RFC connection to the SAP system that is monitored by the agent. The SAP system then opens one internal connection per application server for data collection through function modules and programs. If CCMS alerts are collected by the agent, the SAP system opens one additional (system internal) RFC connection to each application server for this collection thread. When data collection starts, one RFC connection for the agent is opened. Then, up to twice the number of SAP application servers for additional internal system RFC connections are opened.

You must ensure that the instance that is monitoring can accommodate the additional RFC sessions, especially in large systems with 10 or more instances. When the anticipated RFC load for monitoring might adversely affect system performance and tolerances, adjust the SAP profile parameter. Contact your SAP Administrator and see the following SAP Notes:

- Terminal Sessions (default setting: 200) 22099
- Communication/Gateway/Conversation Settings 887909 316877 384971

Basic agent monitoring

The SAP agent creates an IBMMON_AGENT in the SAP system when the agent transport is imported.

This user ID is IBMMON_AGENT with the default password ITMMYSAP. It is preconfigured to be Communication Type user-only and to use the /IBMMON/AUTH authorization profile. This profile, which is created at transport import time, contains the minimal set of permissions to run the agent Advanced Business Application Programming (ABAP) code. Also, this profile accepts a set of limited actions on your SAP system.

If this user ID name is unacceptable, for example, if it violates your installation naming conventions, you can create a different user ID. The user ID can be any allowable SAP user ID, but it requires the complete set of permissions in the /IBMMON/AUTH profile. The user ID requires Communication Type user-only access.

The default user ID provides sufficient authority only for the following purposes:

- Monitoring and data collection
- Closing Computing Center Management System (CCMS) alerts
- Enabling, disabling, and resetting gateway statistics
- Resetting Oracle database statistics

If you choose to limit the action capabilities of the agent, you can remove some of the action permissions such as closing CCMS alerts.

To access data on the IBM Application Performance Management UI Portal for specific components, ensure that you have appropriate authorizations. Following table lists the authorizations that are required to access the data from different sub nodes:

Table 207. The list of authorizations

Sub nodes	**Authorization objects**	**Authorization description**
General system authorizations that include the following sub nodes: • Ins • Sys	S_ADMI_FCD	To access the System
	S_BDS_DS -BC-SRV-KPR-BDS	To access the Document Set
	S_BTCH_JOB	To run operations on the background jobs
	S_CCM_RECV	For transferring the Central System Repository data
	S_C_FUNCT	To make C calls in the ABAP programs
	S_DATASET	To access files
	S_RFC	To check RFC access. The S_RFC authorization object contains the following two sub-authorizations: • RFC1: To provide the authorizations for the RFC1 function group. • SDIFRUNTIME: To provide the authorizations for the SDIFRUNTIME function group.
	S_RFCACL	For RFC User
	S_RZL_ADM	To access Computing Center Management System (CCMS): System Administration
	S_TCODE	To check Transaction Code at Transaction Start
	S_TOOLS_EX	To access Tools Performance Monitor

Table 207. The list of authorizations (continued)		
Sub nodes	**Authorization objects**	**Authorization description**
Authorizations for Solution manager that include the following sub nodes: • Lds • Sol	D_MD_DATA -DMD	To view Data Contents of Master Data
	D_SOLMANBU	To access a Session Type of the Solution Manager
	D_SOLM_ACT	To access a Solution in the Solution Manager
	D_SOL_VSBL	To view a Solution in the Solution Manager
	S_CTS_SADM	To view System-Specific Administration (Transport)
	S_TABU_RFC	To view Client Comparison and Copy: Data Export with RFC
Authorizations for PI that includes the PI sub node	S_XMB_MONI	To access XI Message Monitoring
Authorizations for MAI that includes the Slm sub node	AI_DIAGE2E	To access Solution Diagnostics end-to-end analysis
	AI_LMDB_OB	To access Landscape Management Database (LMDB) Objects
	SM_MOAL_TC	To access Monitoring and Alerting
	SM_WC_VIEW	To access Work Center User Interface Elements
	S_RFC_ADM	To access Administration options for RFC Destination
	S_RS_AUTH	To access BI Analysis in Role
	SM_APPTYPE	To access Solution Manager App Type
	SM_APP_ID	To access applications provided in Work center

Using Central User Administration (CUA)
The Central User Administration (CUA) is used to monitor a SAP system.

Procedure

To use the predefined user ID and authorization role to monitor a SAP system set up with Central User Administration, complete one of the following steps:

- Install the transport into the Central User Administration parent logical system client.
- Manually create the user ID or role in the client where you want to install the transport. The user ID or role is in the client where the transport is installed (imported).
- Manually create the user ID or role in the Central User Administration parent logical system client. Then, distribute the user ID or role to the client where the agent runs.
- Manually create the user ID or role in the Central User Administration parent logical system client and run the agent in this client.

Optional advanced configuration in SAP

You can configure the SAP agent by using standard SAP or agent-provided SAP functions.

Use agent-provided transactions in SAP to customize a number of agent behaviors. After you run the /n/IBMMON/ITM_CONFIG transaction to access the main configuration menu in SAP, select one of the following configuration options:

- "Copy, back up, restore feature and transactions" on page 742
- "Copy, back up, and restore data by using transactions" on page 743
- "Command-line utility tool" on page 743
- "Running the command-line utility on a Windows environment" on page 744
- "Running the command-line utility on a Non-Windows environment" on page 744
- "Alerts maintenance" on page 745
- "Select monitor sets and monitors transaction" on page 746
- "Configure Dialog Step Response Threshold in the SAP system" on page 746

Note: You must preface all /IBMMON/ITM* transactions with /n.

Configuration changes made in these transactions are used immediately by the SAP agent except for those changes made to maintain managed groups. When the managed group configuration changes, the changes are discovered by the SAP agent at the next heartbeat.

Use SAP standard functions to complete the following configuration: "Configure Dialog Step Response Threshold in the SAP system" on page 746

Copy, back up, restore feature and transactions

The copy, back up, and restore features are available to you after you log on to the SAP server and run the following transaction: /n/IBMMON/ITM_CONFIG.

Copy, backup, and restore operations allow you to copy, backup, and restore the IBM Tivoli Monitoring configuration data.

Use this feature to select from the following functions and to save the IBM Tivoli Monitoring configuration data:

- **Copy**

  Use this feature to copy the IBM Tivoli Monitoring configuration settings from one SAP server to another SAP server. For example, you might want to copy the IBM Tivoli Monitoring configuration settings from agent **a1** to SAP server instance SAP2. This agent runs on system **m1** and is configured for SAP server instance SAP 1. All the IBM Tivoli Monitoring configuration settings, except the SAP server instance monitoring settings are copied to the target SAP system. You implement the copy feature by using either the command line utility or the SAP GUI.

- **Backup**

  You can store agent-specific configurations that you completed on the SAP server by taking a backup of the system. Use this feature to save IBM Tivoli Monitoring specific configuration settings on the SAP system. You use the /IBMMON/ITM_CONFIG transaction to enter the settings. The backup file is stored in the work directory on the SAP server to the following path: /usr/sap//DVEBMGS/work.

- **Restore**

  Use this feature to restore IBM Tivoli Monitoring configuration data on the SAP server from the work directory. You can restore the IBM Tivoli Monitoring configuration data on the same SAP server where you completed the backup procedure of this configuration data or another SAP server. You can restore IBM Tivoli Monitoring configuration data to specific SAP and IBM Tivoli Monitoring tables. Configuration files are stored with a date and time stamp so you can select the point to which you want to restore your files.

Agent-specific configurations include configuration settings in the /IBMMON/ITM_CONFIG transaction in SAP. You can complete the following configuration procedures:

- Sample the frequency for alerts.
- Enable specific alerts.
- Store log file names.
- Manage group definitions.
- Select monitor sets and monitors.
- Select SAP instances for monitoring purposes.

Copy, back up, and restore data by using transactions

On the SAP user interface, you copy, back up, and restore data by using the `/n/IBMMON/ITM_CONFIG` transaction.

Before you begin

Use the copy, backup, and restore procedures to copy the IBM Tivoli Monitoring configuration settings from one SAP server to another SAP server. All the IBM Tivoli Monitoring configuration settings, except the SAP server instance monitoring settings are copied to the target SAP system.

Procedure

Complete the following procedures to copy, back up, and restore your data on SAP:

- **Copy**

  a. Enter the target SAP system ID and the existing file name as `source system id__<filenam>date_time`.

     The `/IBMMON/ITM_COPY` transaction creates an IBM Tivoli Monitoring configuration file in the work directory with the filename as SAP `target SAP system id__<filename>_date_time`.

  b. Click **Execute** to copy the IBM Tivoli Monitoring configuration data to the file.

  c. Click **Back** or **Cancel** for returning to the previous IBM Tivoli Monitoring configuration screen.

     Input parameters expected are **`Target System id`** and **`filename`** that are to be copied.

- **Backup**

  a. Log on to the SAP server and start the `/IBMMON/ITM_CONFIG` transaction.

  b. Select **Backup**.

  c. Enter the backup filename.

     The file name is stored as `sys_id_<filename>_date_time`.

  d. Click **Execute** to run the backup and to store the file on the Application Server.

     Note: The backup file is stored in the work directory of the application server.

  e. Click **Back** or **Cancel** for returning to the previous IBM Tivoli Monitoring configuration screen.

- **Restore**

  a. Log on to the SAP server and start the `/IBMMON/ITM_CONFIG` transaction.

  b. Select **Restore**.

  c. Enter the filename to restore as `sys_id_<filename>_date_time`.

  d. Click **Execute** to restore IBM Tivoli Monitoring configuration data.

  e. Click **Back** or **Cancel** for returning to the previous IBM Tivoli Monitoring configuration screen.

Command-line utility tool

You can use the command-line utility tool to copy, backup, and restore IBM Tivoli Monitoring configuration data on the SAP server.

You can run the command-line utility tool on Windows and Non-Windows environment. See "Running the command-line utility on a Windows environment" on page 744 and "Running the command-line utility on a Non-Windows environment" on page 744.

- **Copy**

  Run the **backup** command to copy the IBM Tivoli Monitoring configuration file from the agent directory SAP server instance sap1 to sap2. Enter the file name and sap1 as the source system from the sap1 agent directory. Then, the ABAP function is called that copies the IBM Tivoli Monitoring settings from this file to the IBM Tivoli Monitoring configuration file for Sap2. Now select **Copy** from the sap1 agent directory utility tool and enter a file name and sap2 as the target SAP system.

- **Backup**

  After running the command-line utility tool, select the **Backup** option. Then, you need to enter the file name and the SAP system ID. The tool calls the `/IBMMON/ITM_BACKUP` SAP function module. The function module reads the specific IBM Tivoli Monitoring configuration settings that are stored in tables and stores them with a row and column separator. Then, the command-line utility tool reads the string and writes the data into a file. The file name that is generated has the following format: ID>_<filename>-<date&time>. This file is stored in the directory where the utility program is stored.

- **Restore**

  After you run the command-line utility tool, enter the file name to restore and the target SAP system where you want to restore the file. The command-line utility tool reads the file from the agent directory and calls the `/IBMMON/ITM_RESTORE` SAP function module. Then, the tool passes the IBM Tivoli Monitoring configurations as a string. The SAP function module updates the specific IBM Tivoli Monitoring tables and restores the specific IBM Tivoli Monitoring configurations.

Running the command-line utility on a Windows environment

You can run the command-line utility on a Windows environment to complete copy, backup, and restore procedures.

Procedure

1. Depending on your operating system, complete one of the following procedures:
   - For a 64-bit operating system, set the CANDLEHOME path using command **`set CANDLE_HOME = C:\IBM\APM`** and run the **`ksacopybackuprestore.bat`** command from the following path: `%candle_home%\ TMAITM6x64`.
2. To create a backup file, complete the following steps:

   a) Select **Backup** and enter the file name and source SAP system name.

   b) The backup file is created with the following format: SYS ID>_<filename>_<date&time>.
3. To restore the file, complete the following steps:

   a) Select **Restore** and enter the target SAP system name.

   b) Enter the filename.
4. To copy the file, complete the following steps:

   a) From the source agent, select **Backup** and create a backup file.

   b) Copy the backup file from the source agent directory to the target agent directory.

   c) From the source directory, run the command-line utility tool and select **Copy**.

   d) Enter the file name and the target SAP system.

Running the command-line utility on a Non-Windows environment

You can run the command-line utility on a Non-Windows environment to complete copy, backup, and restore procedures.

Procedure

1. Run the **`ksacopybackuprestore.sh`** command from the following path: `/candle_home/<arch>/sa/shell`.
2. To create a backup file, complete the following steps:

   a) Select **Backup** and enter the file name and source SAP system name.

b) The backup file is created with the following format: SYS ID>_<filename>_<date&time>.

   The backup file is saved to this location: `%candlehome% / arch /sa/bin`.

3. To restore the file, complete the following steps:
   a) Select **Restore** and enter the target SAP system name.
   b) Enter the file name.
4. To copy the file, complete the following steps:
   a) From the source agent, select **Backup** and create a backup file.
   b) Copy the backup file from the source agent directory to the target agent directory.
   c) From the source directory, run the command-line utility tool and select **Copy**.
   d) Enter the file name and the target SAP system.

Alerts maintenance

You can modify alerts that are generated by Tivoli Monitoring by changing their status and thresholds.

This transaction is used to enable or disable alerts generated by Tivoli Monitoring and to set warning and critical thresholds. All alerts generated by Tivoli Monitoring are shown with their current status and threshold values.

When you modify alert status and thresholds, the modified values are used at the next sample time.

Default sample period maintenance

The default sample period provides information about real-time reporting for certain attribute groups.

Some attribute groups have an implicit date and time for each record in the group. For example, the R/3_Abap_Dumps attribute group reports the create time for the dump and the R/3_System_Log attribute group reports the create time for the log entry. These records have a date and time field. You can obtain a report for a short history of the table instead of just the most recent information. This time interval is the time span for data collection and is used as the real-time interval when collecting data. The /IBMMON/ITM_PERIOD transaction defines a default sample period (time span for real-time reporting) for each of these attribute groups. The sample period identifies the length of the data sample period that starts from the current time and works back in time.

Log file name maintenance

Specific log files that are matched only to instances are included in IBM Tivoli Monitoring reports with log file information.

This transaction is used to identify which log files to consider for inclusion in IBM Tivoli Monitoring reports that contain log file information. All log files with a name that matches the specified name patterns on the specified instances are included in the report at the next data collection interval.

Managed group maintenance

The Managed Group names transaction monitors and processes specific transactions in the SAP system.

Use this transaction to maintain IBM Tivoli Monitoring Managed Group definitions. All Managed Group names are passed to the IBM Application Performance Management UI Portal and shown in the Managed System Selection Lists. At the time of data collection, only data that matches the Attribute selection conditions are sent to the SAP agent. This data is shown in reports or used for evaluation in situations and policies.

You use Managed Groups to monitor subsets of information in the SAP system. You focus only on the parts of the SAP system in which you are interested and you ignore other parts that do not concern you. For example, if you are only interested in the response time of transactions that are part of the Financial Application, you create a Managed Group named Financials. Then, you include only Financial transaction codes in it. Whenever the Financials Managed Group is processed by the Tivoli Enterprise Portal only information that contains the specified transaction codes is considered when showing a report, evaluating a situation, or evaluating a policy.

Select monitor sets and monitors transaction

Use the select monitor sets and monitors transaction to edit the Centralized Computing Central Management (CCMS) alerts configuration. For example, you can turn off CCMS alert collection completely.

This transaction is used to select the CCMS monitors from which IBM Tivoli Monitoring retrieves alerts. By default, the Entire System monitor is selected the first time this window is shown. You can change the monitor set, the monitor, or both the monitor set and monitor, and then save the configuration. You can select a maximum of three monitors for which to collect CCMS alerts.

To turn off CCMS alert collection completely, clear the check boxes for all of the monitors and save this configuration.

The agent that is already running reads this configuration and collects the CCMS alerts for the monitors that you selected. However, any CCMS alerts that were already collected by the agent before changing the CCMS alerts configuration remain with the agent and IBM Tivoli Monitoring.

In addition to selecting monitors and monitors sets, this transaction specifies the number of occurrences of an alert type to retrieve. Also, it helps you to decide whether to automatically close the older occurrences of the alerts that are not retrieved.

Configure Dialog Step Response Threshold in the SAP system

You can configure a Dialog Step Response Threshold for any transaction by running the SE16 transaction.

Procedure

1. In the **Table Name** field, type `/IBMMON/ITM_TRSH`, and then select **Table Contents (F7)** to access the table.
2. To view the current threshold settings, select **Execute (F8)**. The transaction names are shown under **WORKLOAD** column; the threshold values are shown under the **THRESHOLD** column.
3. To add a new threshold setting, select **Create (F5)**. Type the transaction name in the **WORKLOAD** field. The following wildcards are accepted for the **WORKLOAD** value:
   - * matches multiple characters
   - + matches any single character
4. Type the threshold value, in milliseconds, in the **THRESHOLD** field. Select **Save** to save this setting. New and changed threshold values do not take effect immediately, but take effect under either of the following conditions:
   - The agent is restarted.
   - The agent reopens its RFC connection to the SAP system. This procedure occurs every 12 heartbeats, which, by default, is about every 2 hours and 10 minutes.

Results

The value entered for the **Threshold** column is returned in the Dialog Step Response Threshold attribute of the R/3_Transacation_Performance attribute group.

Batch Job Operations

You can fetch all Batch Jobs within a specified time interval.

Procedure

Follow the steps after "Importing the ABAP transport on the SAP system" on page 727.

Remember: Critical Constant is set for all the batch jobs.

1. To fetch all Active and Canceled Batch Jobs within a specified time interval.

   Add the following entry in `/IBMMON/ITM_CNFG` table.

Table 208. /IBMMON/ITM_CNFG

PARM_NAME	VALUE_CHAR
BATCH_JOBS_PERF	YES

2. To fetch all Canceled jobs within a specified time interval and all Active jobs irrespective of time interval.

   Add the following entry in `/IBMMON/ITM_CNFG` table.

   Table 209. /IBMMON/ITM_CNFG

PARM_NAME	VALUE_CHAR
BATCH_JOBS_PERF	YES_LONG_RUN

3. To fetch all Batch Jobs within a specified time interval and all Active Batch Jobs irrespective of time interval.

   Add the following entry in `/IBMMON/ITM_CNFG` table.

   Table 210. /IBMMON/ITM_CNFG

PARM_NAME	VALUE_CHAR
BATCH_JOBS_PERF	YES_ALL

   Note:

   - If the configuration parameter is not added, it fetches all Batch Jobs within a specified time interval without Critical Constant set.
   - Number of rows that are fetched is always equal to value of Critical Constant set in Transaction Code `/n/IBMMON/ITM_CONFIG`.

Improving /IBMMON/ITM_MAIALRT_INX Function Module's performance

You can enhance the /IBMMON/ITM_MAIALRT_INX Function Module's performance for SAP agent.

Procedure

Follow the steps to improve the /IBMMON/ITM_MAIALRT_INX function module's performance.

1. Log on toSAP agent GUI.
2. Run SE16 transaction code and enter the table name as `/IBMMON/ITM_CNFG` and press F7.
3. Press F5 or click **Create Entries** and add the following entry in the `IBMMON/ITM_CNFG` table.

   Table 211. /IBMMON/ITM_CNFG

PARM_NAME	VALUE_CHAR
MAI_ALERTS_PERF	YES

   Note:

   - If the Critical Constant is not set in the Transaction Code - `/N/IBMMON/ITM_CONFIG`, then default value is 2500.
   - This process is only applicable for fetching the MAI Alerts from the SAP system where the`PERIOD_START` and `PERIOD_END` is initial.

     Remember: Now the Function Module /IBMMON/ITM_MAIALRT_INX fetches the number of MAI Alerts equivalent to the Critical Constant set in the Transaction Code - `/N/IBMMON/ITM_CONFIG`.
   - If this entry in the `/IBMMON/ITM_CNFG` is not created by default, then the 2500 latest MAI alerts are fetched.

- The number of rows that are fetched is always equal to value of Critical Constant set in Transaction Code `/n/IBMMON/ITM_CONFIG`.

CEN CCMS reporting
Centralized (CEN) Computing Center Management System (CCMS) is a SAP monitoring capability.

Use this capability to report CCMS alerts for multiple SAP systems to a central monitoring hub. You monitor the SAP environment from one CCMS console. Centralized CCMS reporting is best used in the following environments:

- Primarily a CCMS operation where CCMS alerts are the only monitoring data needed.
- Centralized CCMS is part of the SAP environment.
- Large SAP environments with many SAP systems such as ISV and ISP.
- IBM Tivoli Monitoring V5.x integration with SAP agent CCMS adapters.
- Collect alerts from non-ABAP SAP components and application servers.

The SAP agent supports Centralized CCMS for reporting alerts only. Then, you place one SAP agent on a Centralized SAP system and view CCMS alerts for the entire SAP environment. This support is provided in the following ways:

- When reporting CCMS alerts, the agent checks if the alerts are associated with the SAP system that is directly monitored by the agent. If the agent determines that an alert belongs to a different SAP system, it assumes Centralized CCMS and automatically creates additional R3_Group managed systems.
- The <local_SID>-All_CCMS_alerts:Grp managed system is used to report the complete set of alerts from all remote SAP systems. The value of <local_SID> is the system identifier for the SAP system that is directly monitored. For example, if the local SAP system is QA1, this group name would be QA1-All_CCMS_alerts:Grp.
- The <local_SID>-<remote_SID>_CCMS_alerts:Grp managed system is used to report all alerts for one remote SAP system. The value of <local_SID> is the system identifier for the SAP system that is directly monitored. The value of <remote_SID> is the system identifier for the remote SAP system. For example, if the local SAP system is QA1 and the remote SAP system is QA2, this group name would be QA1-QA2_CCMS_alerts:Grp.
- Each of these managed systems in the Navigator tree has the complete set of widgets under it, but only the Alerts widgets has meaningful data.

The SAP agent maintains its definitions of Centralized CCMS groups in the Advanced Business Application Programming (ABAP) code in the directly managed SAP system. You might need to modify these definitions if a SAP system for which you are receiving centralized alerts is also being monitored directly by another instance of the SAP agent. You do not want alerts reported under both systems. You can limit the centralized alert reporting as follows:

- Use the `/IBMMON/ITM_CONFIG` transaction to Maintain Managed Groups. Change the All CCMS alerts group. Remove the remote system from this list by editing the group definition to EXCLUDE the remote system identifier.
- Use the`/IBMMON/ITM_CONFIG` transaction to Maintain Managed Groups. Delete the <remote_SID> CCMS alerts group. For example, if the remote SAP system is QA2, this group name would be QA2 CCMS alerts.

Alternatively, you can use Centralized CCMS to report alerts from all SAP systems, but prevent alert reporting from each locally installed agent. Use the following steps to set up this configuration:

- Configure an instance of the SAP agent to monitor the Centralized CCMS system. Allow the agent to detect and report all alerts from all remote SAP systems.
- Configure an instance of the SAP agent to monitor each remote SAP system. Disable alert collection and reporting for these agent instances by using the /IBMMON/ITM_CONFIG transaction to Select Monitor Sets and Monitors. Within this function, clear the check boxes for all monitors and save this configuration.

The SAP agent support for Centralized CCMS is used in a pure CCMS monitoring environment to view all alerts on a common console. Also, it can be used with its complete set of functions to provide situations, policies, and Take Action commands for the remote SAP systems.

Uninstalling the Advanced Business Application Programming (ABAP) transport from the SAP system
If you choose to remove the SAP agent from your system, you must import Delete transport to the SAP system. Delete transport deletes the SAP agent dictionary objects and function modules.

Before you begin

If the SAP system is version 7.20 or later, before you import the delete transport, in your transport profile, you must add the following transport profile parameter: **`tadirdeletions=true`**. This transport profile parameter is available in tp version 375.57.68 and also in the R3trans version 6.14 release 700 or higher. For more information about removing transport requests from the SAP system, see Deleting transport requests.

Procedure

1. Go to the/ABAP directory on the product CD.
2. Copy the transport files into the SAP environment.
3. Copy the `K711_00xxx_DELETE` and `R711_00xxx_DELETE` files to the SAP Transport System data directory as follows:
   a) Copy the `K711_00xxx_DELETE` file to the `cofiles` directory.
   b) Copy the `R711_00xxx_DELETE` file to the `data` directory.
4. Run the following commands:
   a) **`tp addtobuffer ITMK711_00xxx_DELETE SID pf=\usr\sap\trans\bin\`***PROFILE_NAME*
   b) **`tp import ITMK711_00xxx_DELETE SID client=nnn U16 pf=\usr\sap\trans\bin\`** *PROFILE_NAME*

   Where:

   SID
   Target SAP system ID

   PROFILE_NAME
   Name of the tp profile file

   nnn
   Number for the target client where the agent is to run

SAP instance customization

By default, all the instances of the SAP system are monitored and shown on the IBM Application Performance Management UI Portal.

As an administrator, you choose which SAP instance you want to monitor. Also, as an administrator, you can turn off an SAP instance that you don't want to monitor.

The `/IBMMON/ITM_INSTANCE` custom transaction links to the `/IBMMON/ITM_CONFIG` transaction.

You select the **SAP Instances** option to view the available instances of the SAP server. Then, you select the instance that you want to monitor. These instances are displayed on the IBM Application Performance Management UI Portal. Any inactive or cleared instances aren't shown on the IBM Application Performance Management UI Portal.

Test Connection feature

The Test Connection feature allows you to verify that you can connect your agent to the SAP system that is monitored.

You enter parameters on the GUI to complete the test connection procedure. If you connect to the SAP system successfully, a success message is displayed. Alternatively, if the connection fails, a failure message is displayed.

The **Test Connection** button is available only in the IBM Application Performance Management window.

Modifying the threshold value of an alert

You can modify the **max ccms alert** threshold value that is associated with an alert. By default, the value is 1000, which means that you can view 1000 alerts in the IBM Application Performance Management. Older alerts are removed from the cache

Procedure

1. Complete one of the following steps:
   - On Windows operating system, open the `<cancle home>\tmaitm6\KSAENV` file.
   - On a Non-Windows operating system open the `<candle home>/config/sa.ini` file.
2. Add the *MAX_CCMS_ALERT_THRESHOLD=< Value>* to the end of the file.

   Note: The value must be greater than 100.

Master Control Panel

Master Control Panel provides central console with a user-friendly GUI for all types of configuration settings like Logging of Function Modules, Performance Improvement, Enable/Disable Function Modules, and other configurations like CCMS Old/New Design setting, MAI Configuration, Roll Key Format and Statistical File. Master control panel has some unique features that make it more efficient:

- No special authorizations are required for accessing the Master Control Panel
- Configuration settings set by user are clearly distinguishable with the help of Traffic light signal indicators
- In case of PMR investigation, there is an inbuilt functionality in Master Control Panel in which user can collect diagnostic information containing all Configuration Settings, System Information, Transport Information, Transaction Tracking details, Authorization Objects etc.

To launch Master Control Panel:

1. Go to Transaction code - `/n/ibmmon/itm_mcp` and press **Enter**
2. Master Control Panel will be launched

Set Configurable Parameters

This topic provides detailed information about Set Configurable Parameters operation available under Master Control Panel

This section allows you to set all types of configuration settings provided under **Select Configurations** such as:

- Enable / Disable Logging for Function Modules
- Performance Improvement for Function Modules
- Other Configuration settings that include setting of CCMS design, Roll Key Format, MAI configuration, Statistical File

Click on **Set Configurable Parameters** or icon in the **Master Control Panel Main Screen** to launch for Set Configurable Parameters.

Logging of Function Modules

Click on **Logging of Function Modules** or icon in the **Set Configurable Parameters** to launch **Logging of Function Modules**.

Features of logging function modules are:

- Master Control Panel enables Logging of any Function Modules without navigating to database table - `/IBMMON/ITM_CNFG`

- The Application logs generated by applying the settings for the required Function Modules can be accessed using Transaction Code SLG1
- Function Modules/Attribute groups are populated only if logging mechanism is implemented
- Logging of function modules displays information in a tabular format that contains
  - Function Module Name
  - Attribute Group Name
  - Logging Status with Traffic light signal – Red 🔴⚪⚪ or Green ⚪⚪🟩
  - Logging status – Logging Enabled/Logging Disabled
- The Function Modules are displayed according to their nodes with the respective tabs. Nodes are fetched dynamically according to SAP system configurations. For example, SLM node Function Modules are fetched only on systems where MAI is configured, PI node Function Modules are fetched only on systems where Process Integration is configured

Logging of Function Modules enable few operations such as:

- Enable Logging
  - Click on **Enable Logging** to enable logging for required Function Modules and click on **Enable Logging** button
  - The traffic light signal corresponding to the selected Function module turns green ⚪⚪🟩 if Logging is enabled successfully
  - A success message `Logging Enabled` will appear on screen indicating that the Logging has been enabled
  - The function modules not having logging as enabled shows a red traffic signal light 🔴⚪⚪

  Note: Logging can not be enabled for Function Modules with data collection disabled.
- Disable Logging
  - To Disable Logging, select the required Function Modules and click on **Disable Logging** button
  - The traffic light signal corresponding to the selected Function module turns to red traffic light signal 🔴⚪⚪ if Logging is disabled successfully
  - A success message `Logging Disabled` appears on screen indicating that the Logging has been disabled.
  - The function modules for which logging is enabled shows a green traffic light signal ⚪⚪🟩.

  Note: Logging can not be disabled for Function Modules with data collection is disabled.
- Select All
  - A **Select All** button is available at the bottom of the table
  - This operation selects all the entries in the table

  Note: This option does not select the Function module for which data collection is disabled.
- Deselect All
  - A **Deselect All** button is available at the bottom of the table
  - This operation lets you deselect all the entries in the table

  Note: This option does not select the Function module for which data collection is disabled.

Performance Improvement

Click on **Performance Improvement** or ▦ icon in the **Set Configurable Parameters** to launch **Performance Improvement** window.

Features of Performance Improvement are:

- Performance Improvement is used when there is large number of data or data fetching is taking long time
- It populates only those Function Modules/Attribute groups for which Performance Improvement is implemented
- Data is populated according to nodes and system configurations. For example, SLM node Function Module is populated on systems that do not have MAI configuration and vice versa
- Performance Improvement displays information in a tabular format that contains:
  - Function Module Name
  - Attribute Group Name
  - Performance Improvement Status with Traffic light signal – Red 🔴○○ or Green ○○🟩
  - Buttons for Enable/Disable
  - Value of parameter set in the database table - `/IBMMON/ITM_CNFG`
- A single button **Enable/Disable** is used to enable or disable the functionality.
- Performance improvement is provided to two Function Modules/Attribute Groups given in the section **Data Fetching can be controlled for below attributes**

Data fetching can be controlled for attributes such as:

- Function Module `/IBMMON/ITM_JOBS`
  - To enable the parameters for this Function module, click on **Enable/Disable** button.
  - A default time period gets set in the SAP system for fetching Batch Jobs
  - Four configuration options are available under this feature:
    1. Fetch all Canceled and Active Jobs present within time interval set in the system
       - This configuration setting fetches all Canceled jobs and Active jobs present in the system within the predefined interval set
       - To enable this configuration, click on the radio button for this corresponding configuration and click on **Set Configuration** button
       - The traffic light signal corresponding to the configuration option turns to green○○🟩 if configuration parameter is set successfully
       - The total number of rows returned for the Function module are equivalent to the critical constant value set
       - All other Batch Job configuration settings are removed
    2. Fetch all Active Jobs in the system and all Canceled jobs present within the time interval set in the system
       - This configuration setting fetches all Canceled jobs present in the system within the predefined interval set and all Active jobs irrespective of the time interval set
       - To enable this configuration, click on the radio button for this corresponding configuration and click on **Set Configuration** button
       - The traffic light signal corresponding to the configuration option turns to green○○🟩 if configuration parameter is set successfully

    - The total number of rows returned for the Function module are equivalent to the critical constant value set
    - All other Batch Job configuration settings are removed

3. Fetch all Active Jobs in the system and all Canceled, Scheduled, Finished, Released, Ready jobs present within the time interval set in the system
    - This configuration setting fetches all Canceled, Scheduled, Finished, Released, Ready jobs present in the system within the predefined interval set and all Active jobs irrespective of the time interval set
    - To enable this configuration, click on the radio button for this corresponding configuration and click on **Set Configuration** button
    - The traffic light signal corresponding to the configuration option turns to green ○○🟩 if configuration parameter is set successfully.
    - The total number of rows returned for the Function module are equivalent to the critical constant value set
    - All other Batch Job configuration settings are removed
4. Remove Batch Job configurations
    - This configuration setting fetches all Active, Canceled, Scheduled, Finished, Released, Ready jobs present in the system within the predefined interval set
    - To enable this configuration, click on the radio button for this corresponding configuration and click on **Set Configuration** button
    - The traffic light signal corresponding to all the configuration options turns red 🔴○○ if configuration parameter is removed successfully

After setting the configurations, click on **Confirm** ✓. The `Parameter Value Set` column shows the selected configuration value. Only one configuration can be set at a time.

Note: Enable/Disable Performance parameter is not permitted if the data collection for Function Module is disabled

- Function Module - /IBMMON/ITM_MAIALRT_INX
    - To enable the parameters for this Function module, click on **Enable/Disable** button.
    - The traffic light signal corresponding to the Function module turns green ○○🟩 if performance parameter is enabled successfully
    - The `Parameter Value Set` column shows the selected configuration value
    - The same **Enable/Disable** button must be clicked to disable the performance parameter
    - The traffic light signal corresponding to the Function module turns red 🔴○○ if performance parameter is disabled successfully. The `Parameter Value Set` column will be blank

Note: Enable/Disable Performance parameter is not permitted if the data collection for function module is disabled.

Critical Constant:

- Critical constant value is a limit on total number of entries returned by Function Module. Click on **Set Critical Constant** tab to set a valid critical constant value.
- Default value gets populated in the text box. You can change it if required.

Other Configurations

Click on **Other Configurations** or icon in the **Set Configurable Parameters** window to launch **Other Configurations**.

Other configurations enables operations such as:

- CCMS design
  - This operation enables setting of Old CCMS Design or New CCMS Design according to the requirement
  - To enable New CCMS Design, select the **New CCMS Design** radio button and click on **Set CCMS Design** button.
  - The traffic light signal corresponding to the configuration option turns green ○○🟩 if New CCMS Design is enabled successfully
  - The text corresponding to the configuration is shown as `New CCMS Design is Enabled`
  - To enable Old CCMS Design, select the **Old CCMS Design** radio button and click on **Set CCMS Design** button.
  - The traffic light signal corresponding to the configuration option turns yellow ○⚠○ if Old CCMS Design is enabled successfully
  - The text corresponding to the configuration is shown as `Old CCMS Design is Enabled`
- MAI Configuration
  - This is a predefined setting. If MAI configuration is enabled, the traffic light signal is shown as green ○○🟩 with a message `MAI Configuration is Enabled`
  - If MAI configuration is not enabled, the traffic light signal is shown as red 🔴○○ with a message `MAI Configuration is not Enabled`.
- Roll Key Format
  - This operation enables you to set or remove Roll Key Format according to the requirement
  - To set the Roll Key Format, select the **Yes** radio button and click on **Set Roll Key Format** button
  - The traffic light signal corresponding to the configuration option turns green ○○🟩 if Roll Key Format is set successfully
  - The text corresponding to the configuration is shown as `Roll Key Format is Set`
  - To remove the Roll Key Format, select the **No** radio button and click on **Set Roll Key Format** button
  - The traffic light signal corresponding to the configuration option turns red 🔴○○ if Roll Key Format is removed successfully
  - The text corresponding to the configuration will be shown as `Roll Key Format is not Set`
- Statistical File
  - This setting is applicable for systems that have statistical records
  - At times, few dumps can occur due to statistical records. To avoid these dumps, Statistical file dumps are set to No in the database table
  - This operation allows you to set or remove Statistical File settings according to the requirement
  - To set the Statistical file dumps to No, select the **Enable** radio button and click on **Set Statistical File Setting** button
  - The traffic light signal corresponding to the configuration option turns green ○○🟩 if the Statistical file dumps are set to No successfully

- The text corresponding to the configuration is shown as `Statistical File Settings Enabled (Statfile Dumps are set to "NO")`.
- To remove the Statistical file settings, select the **Disable** radio button and click on **Set Statistical File Setting** button
- The traffic light signal corresponding to the configuration option turns to red if the Statistical file settings are removed successfully
- The text corresponding to the configuration will be shown as `Statistical File Settings Disabled`

ITM Maintenance Transactions

This topic provides detailed information about ITM Maintenance Transaction and how to launch it.

ITM Maintenance Transaction is an old functionality already present with the Transaction code `/n/ibmmon/itm_config`. It is merged in Master Control Panel.

To launch the ITM Maintenance Transactions through Master Control Panel refer the given step:

1. Click on **ITM Maintenance Transactions** or icon in the **Master Control Panel** main screen to launch **ITM Maintenance Transactions**

Enable/Disable Function Modules

This topic provides detailed information on Enable/Disable Function Modules of Master Control Panel.

Click on **Enable/Disable Function Modules** or icon in the **Master Control Panel** to launch **Enable/Disable Function Modules**.

Features of Enable/Disable Function Modules are:

- Master Control Panel enables to Enable or Disable Data collection of any Function Modules without navigating to database table `-/ IBMMON/ITM_CNFG`
- This section populates only those Function Modules or Attribute groups that are applicable to enable or disable data collection
- This operation provides information in a tabular format that contains:
  - Function Module Name
  - Attribute Group Name
  - Data collection Status with Traffic light signal – Red or Green
  - Data collection status- Data Collection Enabled or Data Collection Disabled
- The Function Modules are displayed according to their nodes. Nodes are fetched dynamically according to SAP system configurations. For example, SLM node Function Modules are fetched only on systems where MAI is configured, PI node Function Modules are fetched only on systems where Process Integration is configured.

Enable/Disable Function modules enables operations such as:

- Enable Data Collection
  - To enable Data Collection, select the required Function Modules and click on **Enable Data Collection** button
  - The traffic light signal corresponding to the selected Function module turns green if Data Collection is enabled successfully
  - A message `Data Collection Enabled` appears on screen indicating that the Data Collection has been enabled
- Disable Data Collection

- To disable Data Collection, select the required Function Modules and click on **Disable Data Collection** button
- The traffic light signal corresponding to the selected Function module turns red if Data Collection is disabled successfully
- A message `Data Collection Disabled` appears on screen indicating that the Data Collection has been disabled
- The function modules for which logging is enabled is shown with a green traffic light signal

- Select All
  - A **Select All** button is provided in the bottom of table
  - It lets you select all the entries in the table
- Deselect All
  - A **Deselect All** button is provided in the bottom of table
  - It lets you deselect all the entries in the table

Collect Diagnostic Information

This topic provides detailed information about Collect Diagnostic Information operation available under Master Control Panel

Click on **Collect Diagnostic Information** or icon in the **Master Control Panel** to launch **Collect Diagnostic Information**.

Diagnostic Information collected through this operation:

- List of Enable/Disable Function Modules
- Function Module sub-nodes
- Logging for Function Module
- Other Configuration Settings
- Product Information
- SAP Transaction Tracking
- Statistical file
- System Information
- Transport Information
- Authorization Objects

Note: The report of diagnostic information is downloaded by clicking on the control tab . The report collects most of the data that is present in the database table - `/IBMMON/ITM_CNFG`. If there is no data in the table, the report is not populated.

Configuring SAP HANA Database monitoring

You must configure the SAP HANA Database agent so that the agent can collect data of the SAP HANA database server that is being monitored.

Before you begin

Review the hardware and software prerequisites, see Software Product Compatibility Reports for SAP HANA Database agent

Following are the prerequisites before you configure the SAP HANA Database agent

1. Ensure to create users in all the databases (system and tenant) of the SAP HANA system with the following privileges:

- Role: Monitoring
- System privileges: Monitor Admin

  The user name and password for the system and tenant databases must be the same.

2. When the switching between master to standby connectivity takes place on the SAP HANA Database agent system the agent uses the hostname of Standby Server which needs to be resolved on the agent system. To resolve the hostname to an IP address you need to add a mapping entry in host file of the machine on which the agent is installed.

Note: If you configure the agent using Master Host, then enter the fully qualified host name or IP address of Master Host. If user is configuring agent using Stand by Host, then enter the fully qualified host name or IP address of Stand Host. When you configure the agent through Stand by node the Master node must be down along with the host machine.

About this task

The SAP HANA Database agent is a multiple instance agent. You must create the first instance and start the agent manually.

Procedure

- Windows To configure the agent on Windows systems, complete the following steps:

  a) Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.

  b) In the **IBM Performance Management** window, right-click **Template** under the **Task/SubSystem** column, and click **Configure Using Defaults**.

  The **Monitoring Agent for SAP HANA Database** window opens.

  c) In the **Enter a unique instance name** field, type an agent instance name and click **OK**.

  Important: The agent instance name must match the 3-digit HANA database system identifier (SID). For example, if the SID of the managed SAP HANA database is H01, enter `H01` as the instance name.

  d) In the **Monitoring Agent for SAP HANA Database** window, specify values for the following fields:

  Instance Name
  The default value for this field is identical to the value that you specified in the **Enter a unique instance name** field.

  Server Name
  The fully qualified host name or IP address of the SAP HANA server where the system database is installed.

  Database Name
  The name of the SAP HANA database.

  Port Number
  The SQL port number of the index server service on the system database of the SAP HANA database server.

  HANA DB Administrator
  The user name for accessing the SAP HANA database server.

  HANA DB Administrator Password
  The password for accessing the SAP HANA database server.

  Confirm HANA DB Administrator Password
  The password that is specified in the **HANA DB Administrator Password** field.

  e) Click **OK**.

  f) In the **IBM Performance Management** window, right-click the agent instance that you created and click **Start**.

- Linux AIX To configure the agent on Linux or AIX systems, complete the following steps:

  a) On the command line, change the path to the agent installation directory.

Example: `/opt/ibm/apm/agent/bin`

b) Run the following command where instance_name is the name that you want to give to the instance:

```
./sap_hana_database-agent.sh config instance_name
```

Important: The instance name must match the 3-digit HANA database system identifier (SID). If the SID of the managed SAP HANA database is H01, enter `H01` as the instance name.

c) When the command line displays the following message, type 1 and press Enter: `Edit ‘Monitoring Agent for SAP HANA Database’ setting? [1=Yes, 2=No]`

d) Specify values for the following agent parameters:

Server Name
The fully qualified host name or IP address of the SAP HANA server where the system database is installed.

Database Name
The name of the SAP HANA database.

Port Number
The SQL port number of the index server service on the system database of the SAP HANA database server.

HANA DB Administrator
The user name for accessing the SAP HANA database server.

HANA DB Administrator Password
The password for accessing the SAP HANA database server.

Confirm HANA DB Administrator Password
The password that is specified **HANA DB Administrator Password** field.

e) Run the following command to start the SAP HANA Database agent:

```
./sap_hana_database-agent.sh start instance_name
```

- To configure the agent by using the silent response file, complete the following steps:

a) In a text editor, open the `sap_hana_silent_config.txt` file that is available at the *install_dir*`\samples` path, and specify values for all the parameters.

Windows `C:\IBM\APM\samples`

Linux AIX `/opt/ibm/apm/agent/samples`

b) On the command line, change the path to *install_dir*`\bin`

c) Run the following command:

Windows `sap_hana_database-agent.bat config` *instance_name install_dir* `\samples\sap_hana_silent_config.txt`

Linux AIX `sap_hana_database-agent.sh config` *instance_name install_dir*`\samples\sap_hana_silent_config.txt`

d) Start the agent.

Windows In the **IBM Performance Management** window, right-click the agent instance that you created, and click **Start**.

Linux AIX Run the following command: `./sap_hana_database-agent.sh start` *instance_name*

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the “Change history” on page 54.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring SAP NetWeaver Java Stack monitoring

You must configure the SAP NetWeaver Java Stack agent so that the agent can collect resource monitoring data of the SAP NetWeaver Application Server that is being monitored. To monitor transaction tracking and diagnostics data, you must complete some configuration tasks.

Before you begin

Review the hardware and software prerequisites, see Software Product Compatibility Reports for SAP NetWeaver Java Stack agent

Ensure to complete the following prerequisites tasks before you configure the agent:

- Copy the following JAR files to the `bin` directory:
  - `sapj2eeclient.jar` (the SAP J2EE Engine client API that includes the JMX Adapter)
  - `logging.jar` (the logging library)
  - `com_sap_pj_jmx.jar` (the SAP-JMX library)
  - `exception.jar` (the SAP exception framework)

  The `bin` directory is at the following path:

  Windows `candle_home\TMAITM6_x64`

  Linux `candle_home/interp/sv/bin`

  Important: The JAR files are the same for all the supported operating systems. These files are available in the Diagnostics Agent patch or Software Update Manager (SUM).
- In Environment Variables, add *<candleHome>\svdchome\<build number>\toolkit\lib\win64\ttapi* to path variable.
- To enable Transaction tracking and Diagnostic Data feature, install and configure agent along with Data Collector on SAP NetWeaver Java Server. If the SAP NetWeaver Java Server is Multi Host – Multi Instance then agent needs to be installed and configured along with Data Collector on each host for which Transaction Tracking and Diagnostic Data needs to be enabled.
- Assign the NWA_READONLY role to the *Guest* user for collecting the transaction tracking and diagnostics data.

About this task

The SAP NetWeaver Java Stack agent is a multiple instance agent. You must create the first instance and start the agent manually.

- To configure the agent on Windows systems, you can use the GUI or the silent response file.
- To configure the agent on Linux or AIX systems, you can use the command line or the silent response file.

To configure the collection of transaction tracking and diagnostics data, complete the following tasks:

1. Configure the data collector. For details, see "Configuring the data collector" on page 762.
2. Enable the collection of transaction tracking and diagnostics data. For details, see "Enabling the collection of transaction tracking and diagnostics data" on page 763.

The directions that are mentioned in this topic are for the most current release of the agent, except as indicated. For information about how to check the version of an agent in your environment, see Agent version.

Configuring the agent on Windows systems

You can configure the agent on Windows operating systems by using the **IBM Performance Management** window.

Before you begin

Ensure that the files, which are listed in the "Before you begin" section of the "Configuring SAP NetWeaver Java Stack monitoring" on page 759 topic, are available in the `bin` directory.

About this task

The SAP NetWeaver Java Stack agent provides default values for some parameters. You can specify different values for these parameters.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Template** under the **Task/SubSystem** column, and click **Configure agent**.

   The **Monitoring Agent for SAP NetWeaver Java Stack** window opens.
3. In the **Enter a unique instance name** field, type an agent instance name and click **OK**.

   Important: The agent instance name must match the 3-digit SAP NetWeaver Java Stack system identifier (SID). For example, if the SID of the managed SAP NetWeaver Java Stack is P14, enter P14 as the instance name.
4. In the **Monitoring Agent for SAP NetWeaver Java Stack** window, specify values for the configuration parameters and click **OK**.

   For information about the configuration parameters, see "Configuration parameters of the agent" on page 765.
5. In the **IBM Performance Management** window, right-click the agent instance that you created and click **Start**.

What to do next

- Log in to the Cloud APM console to view the resource monitoring data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see Starting the Performance Management console.
- To collect transaction tracking and diagnostics data, configure the data collector and enable data collection for transaction tracking and diagnostics.

Configuring the agent on Linux or AIX systems

To configure the agent on Linux or AIX systems, you must run the script and respond to prompts.

Before you begin

Ensure that the files, which are listed in the "Before you begin" section of the "Configuring SAP NetWeaver Java Stack monitoring" on page 759 topic, are available in the `bin` directory.

Procedure

1. On the command line, change the path to the agent installation directory.

   Linux `/opt/ibm/apm/agent/bin`

   Linux AIX `/opt/ibm/apm/agent/bin`
2. Run the following command:

   ```
   ./sap_netweaver_java_stack-agent.sh config instance_name
   ```

where *instance_name* is the name that you want to give to the instance.

Important: The agent instance name must match the 3-digit SAP NetWeaver Java Stack system identifier (SID). For example, if the SID of the managed SAP NetWeaver Java Stack is P14, enter P14 as the instance name.

3. When the command line displays the following message, type 1 and press Enter:

```
Edit ‘Monitoring Agent for SAP NetWeaver Java Stack’ setting? [1=Yes, 2=No]
```

4. When you are prompted, specify values for the configuration parameters.

   For information about the configuration parameters, see “Configuration parameters of the agent” on page 765

5. Run the following command to start the agent:

```
./sap_netweaver_java_stack-agent.sh start instance_name
```

What to do next

- Log in to the Cloud APM console to view the resource monitoring data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see Starting the Performance Management console.
- To collect transaction tracking and diagnostics data, configure the data collector and enable data collection for transaction tracking and diagnostics.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

Before you begin

Ensure that the files, which are listed in the "Before you begin" section of the “Configuring SAP NetWeaver Java Stack monitoring” on page 759 topic, are available in the `bin` directory.

About this task

The silent response file contains the agent configuration parameters with default values defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

1. In a text editor, open the `sap_netweaver_java_stack_silent_config.txt` file that is available at the following path, and specify values for the configuration parameters.

   Windows `C:\IBM\APM\samples`

   Linux AIX `/opt/ibm/apm/agent/samples`

   For information about the configuration parameters, see “Configuration parameters of the agent” on page 765

2. On the command line, change the path to `install_dir\bin`
3. Run the following command:

   Windows
   ```
   sap_netweaver_java_stack-agent.bat config instance_name
   install_dir\samples\sap_netweaver_java_stack_silent_config.txt
   ```

   Linux AIX
   ```
   ./sap_netweaver_java_stack-agent.sh config instance_name
   install_dir\samples\sap_netweaver_java_stack_silent_config.txt
   ```

4. Start the agent.

   Windows In the IBM Cloud Application Performance Management window, right-click the agent instance that you created, and click **Start**. Alternatively, you can also run the following command: `sap_netweaver_java_stack-agent.bat start` *`instance_name`*

   Linux AIX Run the following command: `./sap_netweaver_java_stack-agent.sh start` *`instance_name`*

What to do next

- Log in to the Cloud APM console to view the resource monitoring data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see Starting the Performance Management console.
- To collect transaction tracking and diagnostics data, configure the data collector and enable data collection for transaction tracking and diagnostics.

Configuring the data collector

You can use configure the data collector for each application server instance that you want to monitor.

Before you begin

Ensure that the files, which are listed in the "Before you begin" section of the "Configuring SAP NetWeaver Java Stack monitoring" on page 759 topic, are available in the `bin` directory.

Procedure

To configure the data collector by responding to prompts, complete the following steps:

1. On command line, change the path to Windows `install_dir\svdchome\`*`build no`*`\bin\configNW` or Linux AIX `install_dir/svdchome/`*`build no`*`/bin/configNW` directory and run the following script:

   Windows config.bat

   Linux AIX config.sh
2. Select the SAP NetWeaver Application Server version by typing the number that corresponds to product for which you want to configure the data collector and press Enter.
3. When you are prompted for user name, enter the user name that is configured on the SAP NetWeaver Application Server with Java Stack, and press Enter.
4. When you are prompted for password, enter the password, and press Enter.
5. When you are prompted to reenter password, enter the password again, and press Enter.
6. When you are prompted for a P4 port number, enter the P4 port number of the SAP NetWeaver Application Server instance available on the local computer, and press Enter.

   Important: Use this formula for calculating the P4 port number: 50000 + (instance number*100) + 4
7. When you are prompted to select the SAP NetWeaver Application Server instance number, enter the number that corresponds to the instance that you want to configure, and press Enter.

   Remember: For each instance, you need to configure the data collector separately.
8. If you are prompted to enter path to Java home, then use JAVA_HOME from SAP instance. For example, `E:\usr\sap\J01\J04\exe\sapjvm_6`.
9. When you are prompted, enter 1 if you want to enable the collection of transaction tracking data. Otherwise, enter 2, and press Enter.
10. When you are prompted, enter 1 if you want to enable the collection of diagnostic data. Otherwise, enter 2 and press Enter.

Results

The path is generated for loading class files.

What to do next

1. Add the path that is generated to the appropriate environment variable.

   Windows *PATH*
   Linux *LD_LIBRARY_PATH* and *LIBPATH*
   AIX *LD LIBRARY PATH* and *LIB PATH*

   Remember:

   Windows Add the generated path to *PATH* environment variable.

   Linux Add the generated path to *LD_LIBRARY_PATH* and *LIBPATH* in the `/home/`*`sid`*`adm/.cshrc` file in the following format.

   ```
   setenv LD_LIBRARY_PATH /path
   ```

   ```
   setenv LIBPATH /path
   ```

   AIX Add the generated path to *LD LIBRARY PATH* and *LIB PATH* in the `/etc/environment` file in following format.

   ```
   LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/path
   ```

   ```
   LIBPATH=$LIBPATH:/path
   ```
2. Restart the application server instances.
3. Enable data collection for transaction tracking and diagnostics. For details, see "Enabling the collection of transaction tracking and diagnostics data" on page 763.

Enabling the collection of transaction tracking and diagnostics data

On the **Agent Configuration** page, you can enable or disable the data collection for transaction tracking or diagnostics.

Before you begin

Ensure that the data collector is configured. For details, see "Configuring the data collector" on page 762.

About this task

When you enable the collection of transaction tracking data, the agent collects data of the following components:

- Servlet JSP
- RemoteEJB
- JMS

Procedure

Complete the following steps to configure the data collection for each SAP NetWeaver Application Server instance.

1. Log in to the Cloud APM console.
2. From the navigation bar, click **System Configuration** > **Agent Configuration**.

   The **Agent Configuration** page is displayed.
3. Click the **NetWeaver** tab.
4. Select the check boxes of the SAP NetWeaver Application Server instances for which you want to configure the data collection and complete any of the following actions from the **Actions** list.

- To enable transaction tracking, click **Set Transaction Tracking** > **Enabled**. The status in the **Transaction Tracking** column is updated to Enabled for each selected SAP NetWeaver Application Server instance.
- To enable the diagnostic data collection, select **Set Diagnostic Mode** > **Enabled Diagnostic Mode Only**. The status in the Diagnostic Mode column is updated to Enabled for each selected SAP NetWeaver Application Server instance.
- To enable the diagnostic data collection and method trace, select **Set Diagnostic Mode** > **Enabled Diagnostic Mode and Method Trace**. The status in the Diagnostic Mode and Method Trace columns is updated to Enabled for each selected SAP NetWeaver Application Server instance.
- To disable transaction tracking, click **Set Transaction Tracking** > **Disabled**. The status in the **Transaction Tracking** column is updated to Disabled for each selected SAP NetWeaver Application Server instance.
- To disable diagnostic data collection, click **Set Diagnostic Mode** > **Disabled Diagnostic Mode and Method Trace**. The status in the **Diagnostic Mode and Method Trace** columns is updated to Disabled for each selected SAP NetWeaver Application Server instance.

Results

The data collection is configured for each SAP NetWeaver Application Server instance.

What to do next

Log in to the Cloud APM console to view the transaction tracking and diagnostics data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Removing the data collector configuration

You can roll back the changes that were done when the data collector was configured for a SAP Netweaver Application Server with Java Stack instance.

Procedure

To remove the data collector configuration by responding to prompts, complete the following steps:

1. On command line, change the path to Windows install_dir\svdchome\build no\bin\configNW or Linux AIX install_dir/svdchome/*build no*/bin/configNW directory, and run the following script:

   Windows unconfig.bat

   Linux AIX unconfig.sh

   All the instances for which data collector is configured are listed.
2. Enter the number that corresponds to the instance for which you want to remove the data collector configuration and press Enter.

   Tip: To remove the data collector configuration of multiple instances, enter the number that corresponds to the instances separated by commas. To remove the data collector configuration of all instances, you can run the following scripts:

   Windows config.bat -a

   Linux AIX config.sh -a

What to do next

Restart the SAP NetWeaver AS with Java Stack instances.

Restoring the SAP NetWeaver Application Server instance

You can use the restore utility to restore JVM parameters if the SAP NetWeaver Application Server instance does not start after SAP NetWeaver Data Collector configuration or to restore SAP NetWeaver Application Server instance.

Procedure

To restore the SAP NetWeaver Application Server instance by responding to prompts, complete the following steps:

1. On command line, change the path to Windows install_dir\svdchome\build no\bin\configNW or Linux AIX install_dir/svdchome/*build no*/bin/configNW directory and run the following script:

   Windows restoreNW.bat

   Linux AIX restoreNW.sh
2. Select the SAP NetWeaver Application Server version by typing the number that corresponds to product for which you want to restore the JVM parameters and press Enter.
3. When you are prompted for user name, enter the user name for the SAP NetWeaver Application Server instance, and press Enter.
4. When you are prompted for user password, enter the user password for the SAP NetWeaver Application Server instance, and press Enter.
5. When you are prompted for a P4 port number, enter the P4 port number of the SAP NetWeaver Application Server instance available on the local computer, and press Enter.

   If instance information is not found by using the entered P4 port, then the `Could not connect to SAP NetWeaver Server` message is displayed and you are prompted to provide path to the SAP NetWeaver Application Server instance home.

   Example, usr\sap\*System_Name**Instance_Number*
6. When you are prompted to select the SAP NetWeaver Application Server instance number, enter the number that corresponds to the instance that you want to restore, and press Enter.

Results

The following message is displayed:

```
Restore successful. Please restart the instance.
```

Configuration parameters of the agent

When you configure the SAP NetWeaver Java Stack agent, you can change the default value of the parameters, such as SAP_NETWEAVER_P4_HOSTNAME and SAP_NETWEAVER_P4_PORT.

The following table contains detailed descriptions of configuration parameters of the SAP NetWeaver Java Stack agent. You must specify a value for all the fields because these fields are mandatory.

Table 212. Names and descriptions of configuration parameters

Parameter name	Description
Instance Name	The name of the instance. The default value for this field is identical to the value that you specified in the **Enter a unique instance name** field.
SAP_NETWEAVER_P4_HOSTNAME	The host name or IP address of the SAP NetWeaver Application Server.
SAP_NETWEAVER_P4_PORT	The P4 port number of the SAP NetWeaver Application Server.
SAP_NETWEAVER_P4_USERNAME	The user name of the administrator for accessing the SAP NetWeaver Application Server.

Table 212. Names and descriptions of configuration parameters (continued)

Parameter name	Description
SAP_NETWEAVER_P4_ PASSWORD	The password of the administrator for accessing the SAP NetWeaver Application Server.
Confirm SAP_NETWEAVER_P4_ PASSWORD	The password that is specified for the **SAP_NETWEAVER_P4_PASSWORD** parameter.

Configuring Siebel monitoring

The Siebel agent provides a central point of monitoring for your Siebel resources, including Siebel statistics, user sessions, components, tasks, application server, Siebel Gateway Name Server, process CPU and memory usage, and log event monitoring.

Before you begin

- Read the entire "Configuring Siebel monitoring" on page 766 topic to determine what is needed to complete the configuration.
- The directions here are for the most current release of the agent, except as indicated.
- Make sure that the system requirements for the Siebel agent are met in your environment. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Siebel agent.
- Before you configure the Siebel agent, you must verify the Siebel user account that is used by the Siebel agent.

Per Component Statistics Monitoring is disabled by default. You can enable Per Component Statistics Monitoring.

About this task

The Siebel agent is a multiple-instance agent. You must create the first instance, and start the agent manually.

Procedure

1. To configure the agent on Windows systems, you can use the IBM Performance Management window or the silent response file.
   - "Configuring the agent on Windows systems" on page 768.
   - "Configuring the agent by using the silent response file" on page 773.
2. To configure the agent on Linux and UNIX systems, you can run the script and respond to prompts, or use the silent response file.
   - "Configuring the agent by responding to prompts" on page 772.
   - "Configuring the agent by using the silent response file" on page 773.

What to do next

In the Cloud APM console, go to your Application Performance Dashboard pages to view the data that was collected. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you are unable to view the data in the agent dashboards, first check the server connection logs and then the data provider logs. The default paths to these logs are as follows:

- Linux AIX `/opt/ibm/apm/agent/logs`
- Windows `C:\IBM\APM\TMAITM6_x64\logs`

For help with troubleshooting, see the Cloud Application Performance Management Forum.

Verify Siebel user account

You must verify the user account that is used to run the Siebel agent before you configure the agent.

About this task

The user account that is used to run the Siebel agent must have permissions to run the Siebel **srvrmgr** command line utility. To verify that the user account has the required permissions, perform these steps:

Procedure

1. Log on to the computer with the user account that is used to run the Siebel agent.
2. Change directory to the location where the Siebel server is installed.
3. Source the Siebel environment file:

   source siebenv.sh
4. Run the following command:

```
srvrmgr /s Siebel_server /g Siebel_gateway /e Siebel_enterprise
/u useraccount /p password
/c "list servers"
```

where

Siebel_server
Name of the Siebel Application Server.

Siebel_gateway
Name of the currently active Gateway Name Server.

Siebel_enterprise
Name of the Siebel Enterprise.

useraccount
User account that you use to log on to the computer.

password
Password that is associated with the user account.

If the user account has the required permissions, you see output that is similar to the following example where the fields returned are limited to three:

```
Connected to 1 server(s) out of a total of 1 server(s) in the enterprise

srvrmgr:s82win8> list servers show SBLSRVR_NAME, HOST_NAME, SBLSRVR_STATUS

SBLSRVR_NAME  HOST_NAME  SBLSRVR_STATUS
------------  ---------  --------------------------------
s82win8       s82win8    16.0.0.0 [23057] LANG_INDEPENDENT

1 row returned.
```

If the **srvrmgr** command does not run correctly, consult with the Siebel administrator of the server. Ensure that you set the required Siebel environment variables for the user account and the user account has the appropriate permissions to run the **srvrmgr** command.

Enabling Per Component Statistics Monitoring

Per Component Statistics Monitoring is disabled by default. You can enable Per Component Statistics Monitoring by using the KUY_ENABLE_COMP_STATS environment variable.

Before you begin

Because of a known issue with servers of Siebel V8.1 and later, gathering Siebel Component Statistics might have a negative effect on the memory usage of Siebel Gateway Server. This issue is addressed in the Oracle published technote that is named "Gateway Service on Siebel 8.1 or 8.2 Might Consume High Memory Consumption: Recovery (Doc ID 1269177.1)". A fix for the issue is provided in that article. The fix is implemented on the Siebel server.

If Per Component Statistic Monitoring is required in the environment, apply the Oracle fix to the Gateway Servers of Siebel V8.1 and laterbefore you enable Per Component Statistics Monitoring.

About this task

After you apply the Oracle fix, complete the following steps to enable Per Component Statistics Monitoring in the Siebel agent:

Procedure

1. Go to the agent installation directory of the Siebel agent:
   - Windows *install_dir*\TMAITM6_x64
   - Linux AIX *install_dir*/config
2. Edit the Siebel agent configuration file to set KUY_ENABLE_COMP_STATS to true.
   - Windows KUYENV_*instance_name*
   - Linux AIX uy.environment

   where *instance_name* is the instance name of the Siebel agent.
3. Restart the agent.

   Important: To make this setting the default for all new agent instances, set KUY_ENABLE_COMP_STATS to true in the configuration template files:

   - Windows KUYENV
   - Linux AIX This setting is already made the default for new agents instances by editing uy.environment in Step 2.

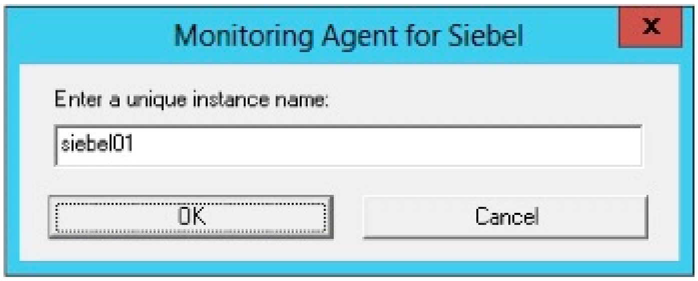
Configuring the agent on Windows systems

You can configure the Siebel agent on Windows operating systems by using the IBM Cloud Application Performance Management window. After you update the configuration values, you must start the agent to save the updated values.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Cloud Application Performance Management**.
2. In the **IBM Performance Management** window, right-click the **Monitoring Agent for Siebel** template, and then click **Configure agent**.

   Remember: After you configure an agent instance for the first time, the **Configure agent** option is disabled. To configure the agent instance again, right-click on it and then click **Reconfigure**.
3. Enter a unique instance name then click **OK**. Use only letters, Arabic numerals, the underline character, and the minus character in the instance name. For example: siebel01.



Monitoring Agent for Siebel

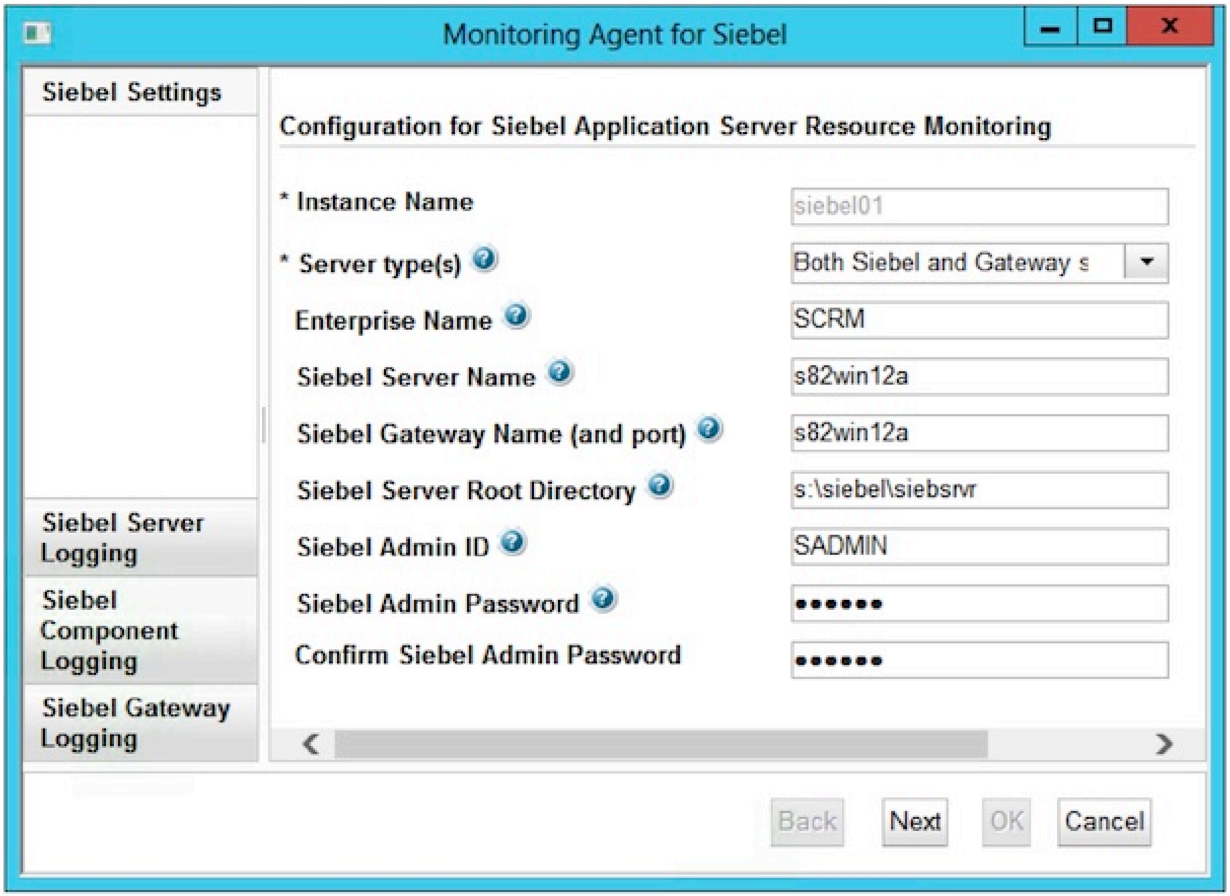
Enter a unique instance name:

siebel01

Figure 22. The window to enter a unique instance name.

4. Select a Server type and enter values for the required fields for that server type, then click **Next**.

   See Table 213 on page 774 for an explanation of each of the configuration parameters.



Monitoring Agent for Siebel

Configuration for Siebel Application Server Resource Monitoring

* Instance Name: siebel01

* Server type(s): Both Siebel and Gateway s

Enterprise Name: SCRM

Siebel Server Name: s82win12a

Siebel Gateway Name (and port): s82win12a

Siebel Server Root Directory: s:\siebel\siebsrvr

Siebel Admin ID: SADMIN

Siebel Admin Password: ••••••

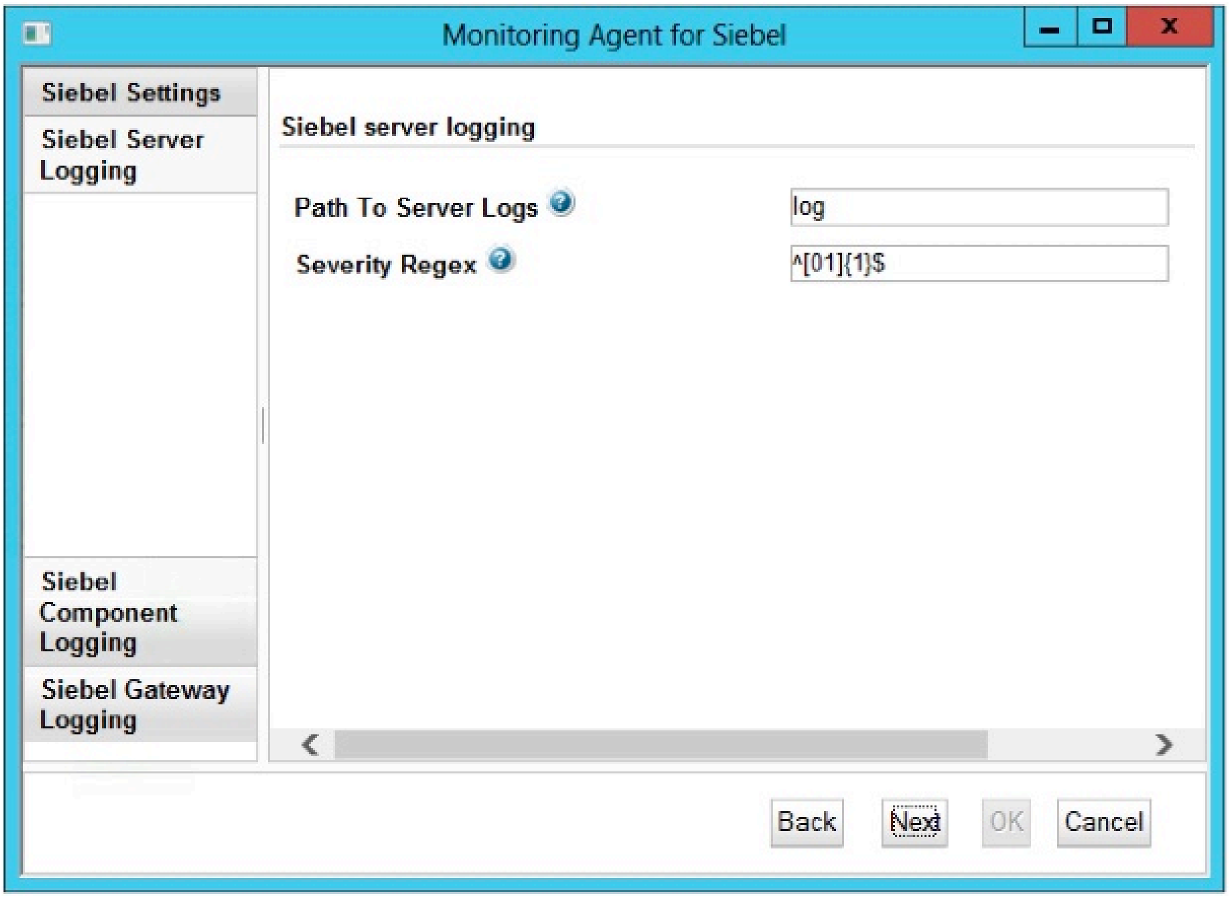
Confirm Siebel Admin Password: ••••••

Figure 23. The window for configuration parameters for Siebel server types that are installed on a Siebel host

Important: If the Siebel agent is installed on a computer with the Siebel Gateway Name Server but without the Siebel Server, data that is displayed in the Application Dashboard is applicable only to the Siebel Gateway Name Server for this instance. All other Siebel agent views are empty.

5. Optional: Edit the values for Siebel server logging, then click **Next**.

   See Table 214 on page 775 for an explanation of each of the configuration parameters.



Siebel server logging

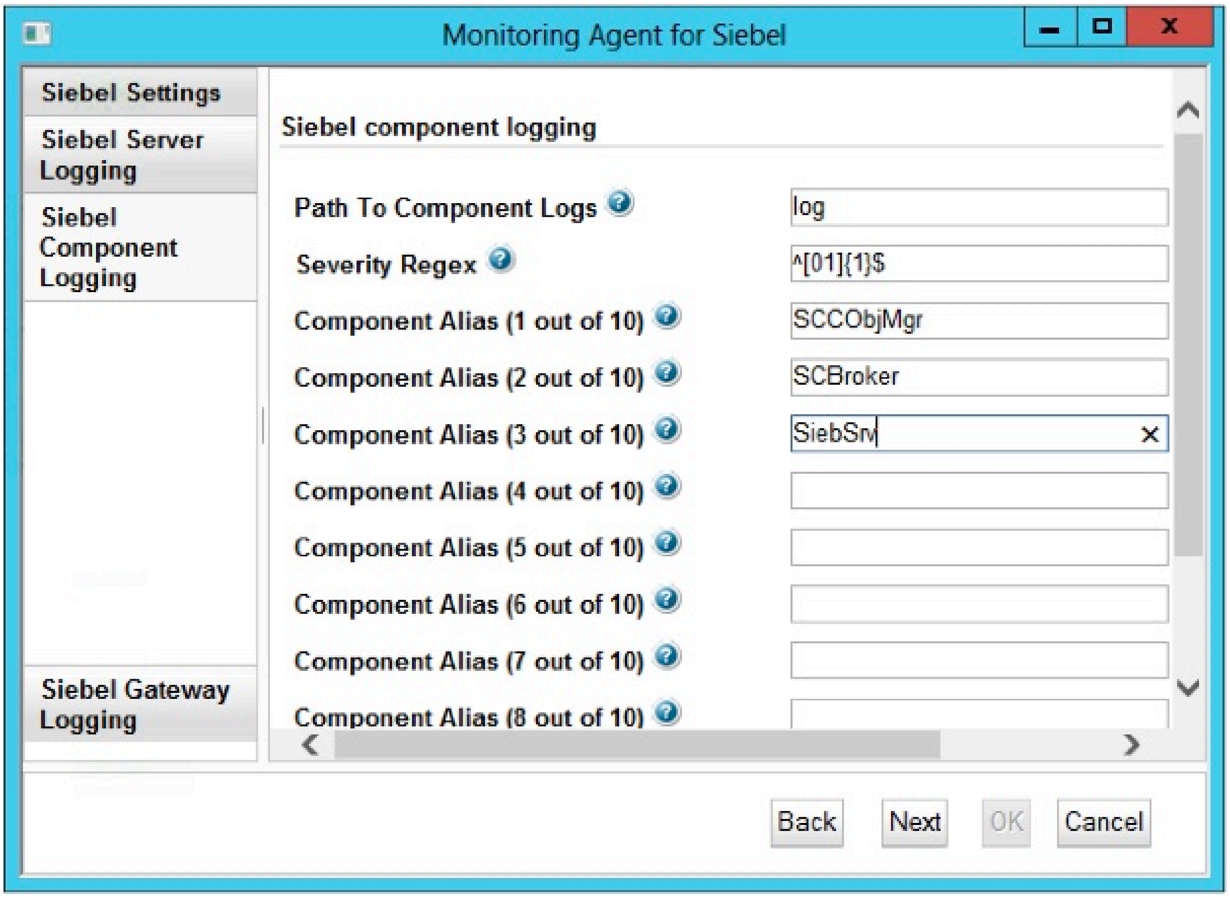
Path To Server Logs: log

Severity Regex: ^[01]{1}$

Figure 24. The window to specify Siebel server logging settings.

6. Optional: Edit the values for Siebel component logging, then click **Next**. By default, logs of components in Table 217 on page 776 are monitored by the Siebel agent. To add up to 10 additional component logs to be monitored, specify the corresponding component alias, for example, `SCBroker`.

   See Table 215 on page 775 for an explanation of each of the configuration parameters.



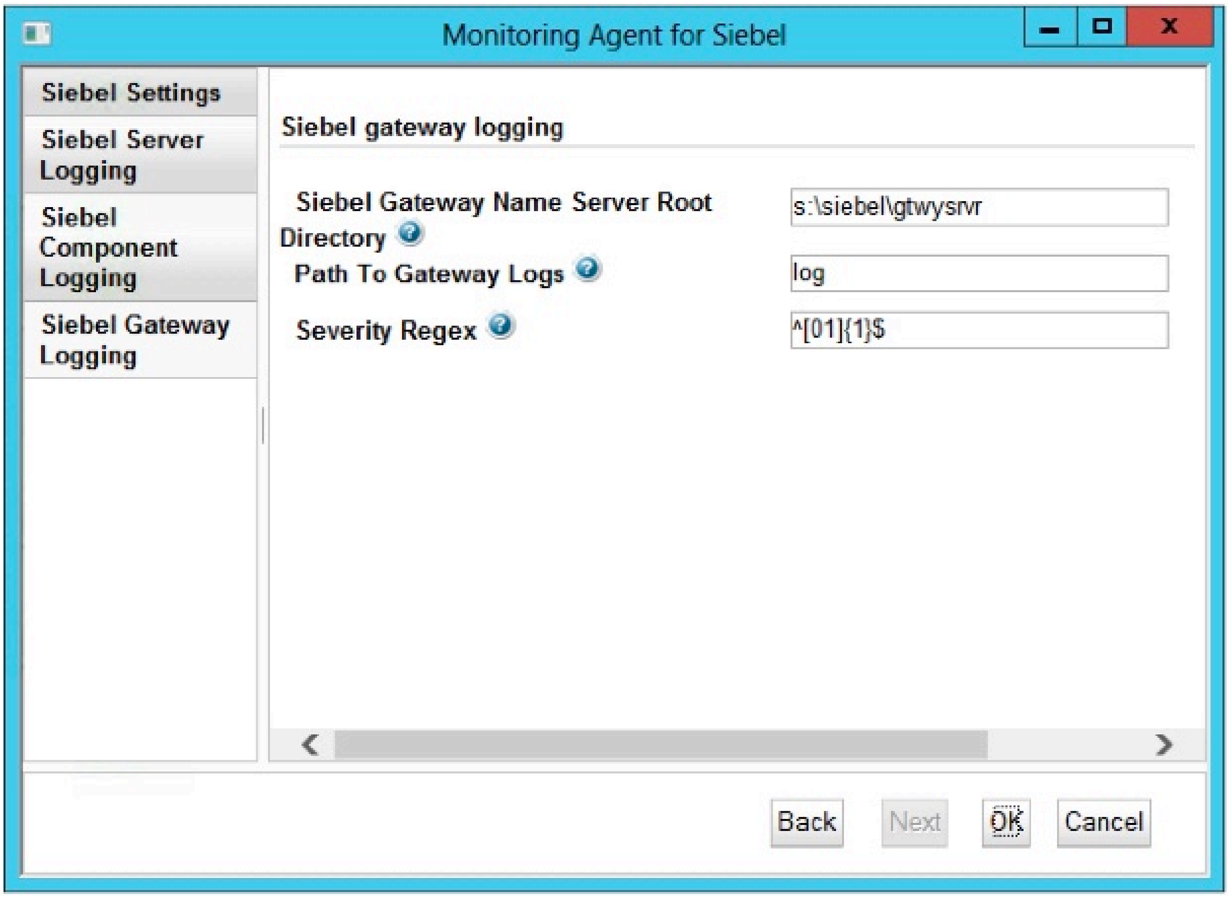
Siebel component logging

Path To Component Logs	log
Severity Regex	^[01]{1}$
Component Alias (1 out of 10)	SCCObjMgr
Component Alias (2 out of 10)	SCBroker
Component Alias (3 out of 10)	SiebSrvr
Component Alias (4 out of 10)	
Component Alias (5 out of 10)	
Component Alias (6 out of 10)	
Component Alias (7 out of 10)	
Component Alias (8 out of 10)	

Figure 25. The window to specify extra component logs that you want to monitor.

7. Optional: Edit the values for Siebel gateway logging.

   See Table 216 on page 776 for an explanation of each of the configuration parameters.



Siebel gateway logging

Siebel Gateway Name Server Root Directory: s:\siebel\gtwysrvr

Path To Gateway Logs: log

Severity Regex: ^[01]{1}$

Figure 26. The window to specify Siebel gateway logging settings.

8. Click **OK** to complete the configuration.
9. In the IBM Cloud Application Performance Management window, right-click the instance that you configured, and then click **Start**.

Configuring the agent by responding to prompts

After installation of the Siebel agent, you must configure it before you start the agent. If the Siebel agent is installed on a local Linux or UNIX computer, you can follow these instructions to configure it interactively through command line prompts.

About this task

Remember: If you are reconfiguring a configured agent instance, the value that is set in the last configuration is displayed for each setting. If you want to clear an existing value, press the space key when the setting is displayed.

Procedure

- Follow these steps to configure the Siebel agent by running a script and responding to prompts.

  a) On the command line, run the following command:

  install_dir/bin/siebel-agent.sh config **instance_name**

  where *install_dir* is the path where the agent is installed and *instance_name* is the name that you want to give to the agent instance.

  Example

  ```
  /opt/ibm/apm/agent/bin/siebel-agent.sh config example-inst01
  ```

b) Respond to the prompts to set configuration values for the agent.

See "Configuration parameters for the Siebel agent" on page 774 for an explanation of each of the configuration parameters.

c) Run the following command to start the agent:

```
install_dir/bin/siebel-agent.sh start instance_name
```

where *install_dir* is the path where the agent is installed and *instance_name* is the name of the agent instance.

Example

```
/opt/ibm/apm/agent/bin/siebel-agent.sh start example-inst01
```

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

The silent response file contains the agent configuration parameters with default values that are defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

- To configure the Siebel agent in the silent mode, complete the following steps:

a) In a text editor, open the `siebel_silent_config.txt` file that is available at the following path:

- Linux AIX `install_dir/samples/siebel_silent_config.txt`
- Windows `install_dir\samples\siebel_silent_config.txt`

where *install_dir* is the path where the agent is installed.

Example

- Linux AIX `/opt/ibm/apm/agent/samples/siebel_silent_config.txt`
- Windows `C:\IBM\APM\samples\siebel_silent_config.txt`

b) In the `siebel_silent_config.txt` file, specify values for all mandatory parameters. You can also modify the default values of other parameters.

See "Configuration parameters for the Siebel agent" on page 774 for an explanation of each of the configuration parameters.

c) Save and close the `siebel_silent_config.txt` file, and run the following command:

- Linux AIX `install_dir/bin/siebel-agent.sh config instance_name install_dir/samples/siebel_silent_config.txt`
- Windows `install_dir\bin\siebel-agent.bat config instance_name install_dir\samples\siebel_silent_config.txt`

where *install_dir* is the path where the agent is installed and *instance_name* is the name of the agent instance.

Important: Ensure that you include the absolute path to the silent response file. Otherwise, the agent data is not shown in the dashboards.

Example

- Linux AIX `/opt/ibm/apm/agent/bin/siebel-agent.sh config example-inst01 /opt/ibm/apm/agent/samples/siebel_silent_config.txt`
- Windows `C:\IBM\APM\bin\ siebel-agent.bat config example-inst01 C:\IBM\APM\samples\siebel_silent_config.txt`

d) Run the following command to start the agent:

- Linux AIX `install_dir/bin/siebel-agent.sh start instance_name`
- Windows `install_dir\bin\siebel-agent.bat start instance_name`

where *install_dir* is the path where the agent is installed and *instance_name* is the name of the agent instance.

Example

- Linux AIX `/opt/ibm/apm/agent/bin/siebel-agent.sh start example-inst01`
- Windows `C:\IBM\APM\bin\siebel-agent.bat start example-inst01`

Configuration parameters for the Siebel agent

The configuration parameters for the Siebel agent are displayed in tables which group them according to categories.

1. Siebel Settings - General Siebel environment settings.
2. Siebel Server Logging - Settings specific to monitoring Siebel server logs.
3. Siebel Component Logging - Settings specific to monitoring a custom list of Siebel component logs.
4. Siebel Gateway Logging - Settings specific to monitoring Siebel gateway logs.

Table 213. Siebel Settings

Parameter name	Description	Required for server type choice	Silent configuration file parameter name
Server type(s)	Indicates the server types that are installed on the local computer.	• **Gateway server only** • **Siebel server only** • **Both Siebel and Gateway server**	**KUY_SERVER_TYPE**
Enterprise Name	The name of the Siebel enterprise.	• **Siebel server only** • **Both Siebel and Gateway server**	**KUY_ENTERPRISE**
Siebel Server Name	The name of the Siebel Server to monitor. **Note:** This name is not the host name of the server. It is the server name that is used when you run the Siebel **srvrmgr** command.	• **Siebel server only** • **Both Siebel and Gateway server**	**KUY_SERVER**

Table 213. Siebel Settings (continued)

Parameter name	Description	Required for server type choice	Silent configuration file parameter name
Siebel Gateway Name	The Siebel Gateway Name Server to monitor and optionally the port, for example, `gtwysrvr` or `gtwysrvr:1234`.	- **Siebel server only** - **Both Siebel and Gateway server**	**KUY_GATEWAY**
Siebel Server Root Directory	The base installation directory for the Siebel Application Server.	- **Siebel server only** - **Both Siebel and Gateway server**	**KUY_INSTALL_ROOT**
Siebel Admin ID	The Siebel specific user ID that the agent uses to authenticate to the Siebel enterprise when you run the **srvrmgr** command. For example: `SADMIN`	- **Siebel server only** - **Both Siebel and Gateway server**	**KUY_ADMIN_ID**
Siebel Admin password	The password for the Siebel Server administrator.	- **Siebel server only** - **Both Siebel and Gateway server**	**KUY_ADMIN_PASSWORD**

Table 214. Siebel Server Logging Settings

Parameter name	Description	Silent configuration file parameter name
Path To Server Logs	The relative path from "Siebel Server Root Directory" to server logs. To disable capturing of Siebel server logging, enter any invalid path. For example: `xyz`.	**KUY_SERVER_LOGGING_PATH**
Severity Regex	The regular expression that is used to capture Siebel server logs matching a severity level. Using the default of `^[01]{1}$` facilitates capturing level 0 and level 1 errors.	**KUY_SERVER_LOGGING_SEVERITY_REGEX**

Table 215. Siebel Component Logging Settings

Parameter name	Description	Silent configuration file parameter name
Path To Component Logs	The relative path from "Siebel Server Root Directory" to server logs. To disable capturing of Siebel server logging, enter any invalid path. For example: `xyz`.	**KUY_COMPONENT_LOGGING_PATH**

Table 215. Siebel Component Logging Settings (continued)

Parameter name	Description	Silent configuration file parameter name
Severity Regex	The regular expression that is used to capture Siebel server logs matching a severity level. Using the default of `^[01]{1}$` facilitates capturing level 0 and level 1 errors.	**KUY_COMPONENT_LOGGING_SEVERITY_ REGEX**
Component Alias (*N* out of 10)	The Component alias for which to monitor an additional Component's log. Example: `SCBroker`. Where *N* is 1 - 10 optional components.	**KUY_CUSTCOMPLOG_00** through **KUY_CUSTCOMPLOG_09**

Table 216. Siebel Gateway Logging Settings

Parameter name	Description	Silent configuration file parameter name
Siebel Gateway Name Server Root Directory	The base installation directory for the Siebel Gateway Name Server.	**KUY_GATEWAY_ROOT**
Path To Gateway Logs	The relative path from Siebel Gateway Name Server Root Directory to gateway logs. To disable capturing of Gateway Name Server logging, enter any invalid path. For example: `xyz`.	**KUY_GW_LOGGING_PATH**
Severity Regex	The regular expression that is used to capture Siebel server logs matching a severity level. Using the default of `^[01]{1}$` facilitates capturing level 0 and level 1 errors.	**KUY_GW_LOGGING_SEVERITY_REGEX**

Siebel component logs that are always monitored

Component logs are always monitored for 10 Siebel components.

Table 217. Siebel component aliases and names for which component logs are always monitored.

Component alias	Component name
SCCObjMgr	Call Center Object Manager
SMObjMgr	Marketing Object Manager
SSEObjMgr	Sales Object Manager
CommInboundRcvr	Communications Inbound Receiver
CommOutboundMg	Communications Outbound Manager
CommSessionMgr	Communications Session Manager
WorkMon	Workflow Monitor Agent
WfProcBatchMgr	Workflow Process Batch Manager

Table 217. Siebel component aliases and names for which component logs are always monitored. (continued)

Component alias	Component name
WfProcMgr	Workflow Process Manager
SiebSrvr	Siebel Server

Configuring Sterling Connect Direct monitoring

You must configure the Sterling Connect Direct agent so that the agent can collect data from the Connect Direct servers to monitor the statistics of file transfer and health of Connect Direct servers.

Before you begin

Review the hardware and software prerequisites, see Software Product Compatibility Reports for Sterling Connect Direct agent

About this task

- To configure the agent on Windows systems, you can use the IBM Cloud Application Performance Management window or the silent response file.
- To configure the agent on Linux systems, you can run the script and respond to prompts, or use the silent response file.

Configuring the agent on Windows systems

You can use the IBM Cloud Application Performance Management window to configure the agent on Windows systems.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Template** in the **Task/SubSystem** column, and click **Configure agent**.
3. In the **Enter a unique instance name** field, type an agent instance name and click **OK**.

   Note: Limit the length of agent instance name. Preferably in the range 7 - 10 characters.
4. In the **Monitoring Agent for Sterling Connect Direct** window, in the **Connect Direct Server Details** tab specify values for the configuration parameters and click **OK**.

   For information about the configuration parameters, see "Configuration parameters of the agent" on page 779.
5. Click **Next**.
6. On Java Parameters tab, keep default values and click **Next**.
7. On Java API Client Configuration tab, click **OK**.
8. In the **IBM Performance Management** window, right-click the agent instance that you created and click **Start** to start the agent.

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the console, see "Starting the Cloud APM console" on page 957.

For help with troubleshooting, see the Troubleshooting section.

Configuring the agent on Linux systems

To configure the agent on Linux operating systems, you must run the script and respond to prompts.

Procedure

1. On the command line, change the path to the agent installation directory.
   Example `/opt/ibm/apm/agent/bin`
2. Run command **`/sterling_connect_direct-agent.sh config instance_name`**.

   Note: The *instance_name* is the name that you want to give to the agent instance.
3. Command line displays the message `Edit 'Monitoring Agent for Sterling Connect Direct' setting? [1=Yes, 2=No]`.
4. Enter 1 to edit the settings.
5. Specify values for the configuration parameters when you are prompted. For information about the configuration parameters, see "Configuration parameters of the agent" on page 779.
6. Run the command to start the agent **`./sterling_connect_direct-agent.sh start instance_name`**

What to do next

Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. Get more information on using the Cloud APM console, at "Starting the Cloud APM console" on page 957.

For help with troubleshooting, see the Troubleshooting section.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

You can use the silent response file to configure the Monitoring Agent for Sterling Connect Direct on Linux and Windows systems. After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

1. Open silent response file present at *`install_dir`*`/samples/sterling_connect_direct_silent_config.txt` in a text editor.
2. Enter server name, username, password, installation directory in the file and save the file.
3. In the command prompt, go to *`install_dir`*`/bin` and run the command

   Linux AIX `./sterling_connect_direct-agent.sh config <Instance_name>` *install_dir*`/samples/sterling_connect_direct_silent_config.txt`.

   Windows `./sterling_connect_direct-agent.bat config <Instance_name>` *install_dir*`/samples/sterling_connect_direct_silent_config.txt`.

What to do next

Log in to the Cloud APM console to view data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

For help with troubleshooting, see the Troubleshooting section.

Configuration parameters of the agent

While configuring the Monitoring Agent for Sterling Connect Direct you can set the values for configuration parameters.

The following table contains detailed descriptions of the configuration parameters of the Monitoring Agent for Sterling Connect Direct.

Table 218. Names and descriptions of the configuration parameters

Parameter name	**Description**	**Mandatory field**
Instance Name	The default value for this field is identical to the value that you specify in the **Enter a unique instance name** field.	Yes
Server name	The host name or IP of Sterling Connect Direct server.	Yes
Server port	The Port of Sterling Connect Direct server. Default value for Sterling Connect Direct is 1363.	Yes
Username	The username to connect to Sterling Connect Direct server.	Yes
Password	The password to connect to Sterling Connect Direct server.	Yes
Java Home	Path to the folder where Java is installed.	No
Java trace level	The trace level used by Java providers. Default value for Sterling Connect Direct is Error	Yes
JVM arguments	This parameter allows you to specify an optional list of arguments to the java virtual machine.	No
Class Path for external jars	The path for jars required by Java API data provider that are not included with the agent.	No

Configuring Sterling File Gateway monitoring

The Monitoring Agent for Sterling File Gateway monitors the IBM Sterling File Gateway application by using the business-to-business (B2B) REST APIs and file gateway database. You must configure the Sterling File Gateway agent so that the agent can collect data from the data sources and monitor the statistics and health of the Sterling File Gateway application. You can configure the agent on Windows and Linux systems.

Before you begin

- Review the hardware and software prerequisites, see Software Product Compatibility Reports for Sterling File Gateway agent.
- Ensure that the B2B REST APIs are installed on your file gateway node. For more information about the B2B REST API installation, see “Installing the B2B REST API” on page 780.

About this task

The Sterling File Gateway agent is a multiple instance agent. You must create the first instance and start the agent manually.

- To configure the agent on Windows systems, you can use the **IBM Performance Management** window or the silent response file.
- To configure the agent on Linux systems, you can run the script and respond to prompts, or use the silent response file.

Installing the B2B REST API

You can install and configure the business-to-business (B2B) REST APIs on your Sterling File Gateway node. The B2B REST APIs are available in the B2B Integrator installer (V5.2.6.2).

Procedure

1. Navigate to the `<install_dir>/bin` directory.

   Where, *install_dir* is the agent installer directory for the B2B integrator.
2. Run the following command:

   - Linux `./InstallService.sh/`*`install_dir`*`/bin/b2bAPIs_10000602.jar`

     Where, *<install_dir>* is the location where you extracted the media file content.
   - Windows `./InstallService.cmd/`*`install_dir`*`/bin/b2bAPIs_10000602.jar`

     Where, *<install_dir>* is the B2B installer folder.

Configuring the Sterling File Gateway agent on Windows systems

You can configure the Sterling File Gateway agent on Windows operating systems by using the **IBM Cloud Application Performance Management** window. After you update the configuration values, you must start the agent to save the updated values.

About this task

The Sterling File Gateway agent provides default values for some parameters. You can specify different values for these parameters.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Monitoring Agent for Sterling File Gateway**, and then click **Configure agent**.

   Remember: After you configure the agent for the first time, the **Configure agent** option is not available. To configure the agent again, click **Reconfigure**.
3. In the Sterling File Gateway agent window, complete the following steps:

   a) Enter a unique name for the Sterling File Gateway agent instance, and click **OK**.

   b) On the **B2B API Details** tab, specify values for the configuration parameters, and then click **Next**.

   c) On the **Database Details** tab, specify values for the configuration parameters, and then click **Next**.

   d) On the **Java API** tab, specify values for the configuration parameters, and then click **OK**.

   For more information about the configuration parameters in each tab of the Sterling File Gateway agent window, see the following topics:

   - "Configuration parameters for the B2B API details" on page 784
   - "Configuration parameters for database details" on page 785
   - "Configuration parameters for the Java API" on page 785
4. In the **IBM Performance Management** window, right-click **Sterling File Gateway agent**, and then click **Start**.

Configuring the Sterling File Gateway agent on Linux systems

You can run the configuration script and respond to prompts to configure the Sterling File Gateway agent on the Linux operating systems.

Procedure

1. Go to the command line and run the **`<install_dir>/bin/sterling_file_gateway-agent.sh config instance_name`** command.

   Where, the *instance_name* is the name that you want to give to the instance and *install_dir* is the agent installation directory path.
2. You are prompted to provide values for all the mandatory configuration parameters. You can modify the default values of configuration parameters.

   For more information about the configuration parameters, see the following topics:

   - "Configuration parameters for the B2B API details" on page 784
   - "Configuration parameters for database details" on page 785
   - "Configuration parameters for the Java API" on page 785
3. To start the agent, run the **`<install_dir>/bin/sterling_file_gateway-agent.sh start instance_name`** command.

Configuring Sterling File Gateway agent by using the silent response file

You can use the silent response file to configure the Sterling File Gateway agent without responding to prompts when you run the configuration script. You can configure the agent that uses the silent response file on both Windows and Linux systems. The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

The silent response file contains the agent configuration parameters with default values that are defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

- To configure the Sterling File Gateway agent in the silent mode, complete the following steps:
  a) In a text editor, open the `sterling_file_gatway_silent_config.txt` file that is available at the following path:

     – Linux *`install_dir`*`/samples/sterling_file_gatway_silent_config.txt`

       Example **`/opt/ibm/apm/agent/samples/sterling_file_gateway_silent_config.txt`**

     – Windows *`install_dir`*`\samples\sterling_file_gateway_silent_config.txt`

       Example **`C:\IBM\APM\samples\sterling_file_gateway_silent_config.txt`**

  b) In the `sterling_file_gateway_silent_config.txt` file, specify values for all the mandatory parameters. You can also modify the default values of other parameters.

     For more information about the configuration parameters, see the following topics:

     – "Configuration parameters for the B2B API details" on page 784

     – "Configuration parameters for database details" on page 785

- "Configuration parameters for the Java API" on page 785

c) Save and close the `sterling_file_gateway_silent_config.txt` file, and run the following command:

- Linux *install_dir*/bin/sterling_file_gateway-agent.sh config *instance_name* *install_dir*/samples/sterling_file_gateway_silent_config.txt

  Example **/opt/ibm/apm/agent/bin/sterling_file_gateway-agent.sh config instance_name /opt/ibm/apm/agent/samples/sterling_file_gateway_silent_config.txt**

- Windows *install_dir*/bin/sterling_file_gateway-agent.bat config *instance_name* *install_dir*/samples/sterling_file_gateway_silent_config.txt

  Example **C:\IBM\APM\bin\sterling_file_gateway-agent.bat config instance_name C:\IBM\APM\samples\sterling_file_gateway_silent_config.txt**

  Where *instance_name* is the name that you want to give to the instance and *install_dir* is the path where the agent is installed.

Important: Ensure that you include the absolute path to the silent response file. Otherwise, the agent data is not shown in the dashboards.

d) Run the following command to start the agent:

- Linux *install_dir*/bin/sterling_file_gateway-agent.sh start *instance_name*

  Example **/opt/ibm/apm/agent/bin/sterling_file_gateway-agent.sh start instance_name**

- Windows *install_dir*\bin\sterling_file_gateway-agent.bat start *instance_name*

  Example **C:\IBM\APM\bin\sterling_file_gateway-agent.bat start instance_name**

Configuring agent environment variables for the data provider on Linux

You can configure the Sterling File Gateway agent environment variables for the data provider on Linux operating systems.

About this task

The Sterling File Gateway agent provides environment variables that you can configure for the data provider.

Procedure

1. Go to the `<install_dir>/agent/config` directory.
2. Open the `.fg.environment` file in an editor and edit the environment variables.

   For more information about the agent environment variables that you can configure, see "Environment variables for the data provider" on page 783.

Configuring agent environment variables for the data provider on Windows

You can configure the Sterling File Gateway agent environment variables for the data provider on Windows operating systems by using the **IBM Performance Management** window.

About this task

The Sterling File Gateway agent provides environment variables that you can configure for the data provider.

Procedure

1. Click **Start** > **All Programs** > > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click the agent instance and click **Advanced** > **Edit ENV File** and edit default values for the environment variables.

   For more information about the agent environment variables that you can configure, see "Environment variables for the data provider" on page 783.

Environment variables for the data provider

After you configure the Sterling File Gateway agent, you can modify some threshold duration values related to the agent data collection. You can specify these values in the agent environment file.

The following table contains detailed description of the environment variables for the data provider.

Table 219. Name and description of the environment variables for the data provider

Parameter name	Description
Collection duration for files transfer (in hours) (**KFG_FILE_ARRIVED_INTERVAL**)	The duration in hours for which the agent collects data for file transfers. The default value is 24 hours.
Collection intervals for files transfer activities that are displayed as a line chart (in hours) (**KFG_FILE_ACTIVITY_INTERVAL**)	The duration in hours for which the agent collects data for file transfer activities. The default value is 1 hour. For example, the agent collects file transfer activities that occurred in last 1 hour. This data is visible in terms of line charts on the instance page. The default value is 1 hour.
Threshold interval for inactive partners (in days) (**KFG_INACTIVE_PARTNERS_INTERVAL**)	The threshold duration when the partner is inactive or not received or uploaded any file. The default value is 10 days. For example, if any partner that does not receive or transfer a file in last 10 days, displays as "Inactive" on the agent.
Maximum number of data provider log files (**KFG_LOG_FILE_MAX_COUNT**)	The maximum number of log files that the data provider creates before it overwrites the previous log files. The default value is 10.
Maximum size in KB of each data provider log (**KFG_LOG_FILE_MAX_SIZE**)	The maximum size in KB that a data provider log file must reach before the data provider creates a new log file. The default value is 5190 KB.

Table 219. Name and description of the environment variables for the data provider (continued)

Parameter name	Description
Level of detail in data provider log (**KFG_LOG_LEVEL**)	The level of details that are included in the log file that the data provider creates. The default value is 4 (Info). The following values are valid: • 1 (Off): No messages are logged. • 2 (Severe): Only errors are logged. • 3 (Warning): All errors and messages that are logged at the severe level and potential errors that might result in undesirable behavior. • 4 (Info): All errors and messages that are logged at the warning level and high-level informational messages that describe the state of the data provider when it is processed. • 5 (Fine): All errors and messages that are logged at the information level and low-level informative messages that describe the state of the data provider when it is processed. • 6 (Finer): All errors and messages that are logged at the fine level plus detailed informative messages, such as performance profiling information and debug data. Selecting this option can adversely affect the performance of the monitoring agent. This setting is intended only as a tool for problem determination along with the IBM support staff. • 7 (Finest): All errors and messages that are logged at the Fine level and the most detailed informative messages that include low-level programming messages and data. Selecting this option might adversely affect the performance of the monitoring agent. This setting is intended only as a tool for problem determination along with the IBM support staff. • 8 (All): All errors and messages are logged.
Fetching events for all file transfers (**KFG_ALL_FGEVENTS**)	The flag for fetching events for all file transfers. The valid values are, Yes or No. The default value is No. If the value is set to No, then agent fetches events for failed file transfers for a user configurable duration. If the value is set to Yes, then the agent fetches events for all file transfers for a user configurable duration.

Configuration parameters for the B2B API details

When you configure the Sterling File Gateway agent, you must specify values of the configuration parameters for the business-to-business (B2B) API details.

The following table contains detailed description of the configuration parameters for the B2B API details.

Table 220. Name and description of the configuration parameters for the B2B API details

Parameter name	Description
Instance Name (**KFG_Instance_Name**)	The name of the instance. **Restriction:** The Instance Name field displays the name of the instance that you specify when you configure the agent for the first time. When you configure the agent again, you cannot change the instance name of the agent.
Server Name (**KFG_API_SERVICES_Node_ ADDRESS**)	The host name or IP address of the B2B API service.

Table 220. Name and description of the configuration parameters for the B2B API details (continued)

Parameter name	Description
Server Port (**KFG_API_SERVICES_PORT**)	The port of the B2B API.
User Name (**KFG_API_SERVICES_USERNAME**)	A user name to connect to the B2B API service.
Password (**KFG_API_SERVICES_PASSWORD**)	The password for the user name that you use to connect to the B2B API service.

Configuration parameters for database details

When you configure the Sterling File Gateway agent, you must specify values of the configuration parameters for the database details.

The following table contains detailed description of the configuration parameters for the database details.

Table 221. Name and description of the configuration parameters for the database details

Parameter name	Description
Database Server Name (**KFG_DB_Node_ADDRESS**)	The host name or IP address of the Sterling File Gateway database server.
Database User (**KFG_DB_USERNAME**)	The name of the database user.
Database Password (**KFG_DB_PASSWORD**)	The password of the database.
Database Port (**KFG_DB_PORT**)	The port of the database.
Database Type (**KFG_DB_TYPE**)	The type of the database.

Configuration parameters for the Java API

When you configure the Sterling File Gateway agent, you must specify values of the configuration parameters for the Java API.

The following table contains detailed description of the configuration parameters for the Java API.

Table 222. Name and description of the configuration parameters for the Java API

Parameter name	Description
Class path for the external JAR (**KFG_CLASSPATH**)	The database driver JAR path that you want to specify for the corresponding database.

Configuring Sybase Server monitoring

The Sybase agent offers a central point of management for distributed databases. It collects the required information for database and system administrators to examine the performance of the Sybase server system, detect problems early and prevent them. Database and system administrators can set the required threshold levels and flags to trigger alerts when the system reaches these thresholds.
You must configure the Monitoring Agent for Sybase Server to monitor Sybase server.

Before you begin

Review the hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Sybase agent.

About this task

The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see "Change history" on page 54.

The Sybase agent is a multiple instance agent, you must configure and start each agent instance manually.

Procedure

1. Configure the monitoring agent.
   - "Configuring the agent by using the command line interface" on page 787
   - "Configuring the agent by using the silent response file" on page 789
2. Start and stop the monitoring agent by using agent command **sybase-agent**.

   For more information about the **sybase-agent**, see *Using agent commands* in https://www.ibm.com/support/knowledgecenter/SSHLNR_8.1.4/com.ibm.pm.doc/welcome.htm.
3. Connect the monitoring agent to the Performance Management server by using the command **agent2server**.

   For more information about the **agent2server**, see *Using agent commands* in https://www.ibm.com/support/knowledgecenter/SSHLNR_8.1.4/com.ibm.pm.doc/welcome.htm.

Granting permissions

You must grant permissions to the user ID that is used to monitor the Sybase server.

Before you begin

Install the Sybase agent.

You must have the database administrator role to grant permissions.

About this task

The user ID that is used by the monitoring agent must have access to Sybase tables and the installed monitor tables.

You can perform the following tasks:

- Create a user ID for the monitoring agent.
- Grant permission to the new user ID and the installed monitor tables.

If you are not running the Sybase agent as root user, make sure that the user ID belongs to the Sybase group and has read-access to the Sybase log files.

Procedure

1. Enter the command for the operating system you are using.
   - Windows

     ```
     cd install_dir\tmaitm6\SQLLIB
     ```
   - UNIX

     ```
     cd install_dir/misc
     ```

   Where, *install_dir* is the home directory where the Sybase server is installed.
2. Use the **isql** command to log in to the Sybase server as user sa.

3. Run the following command to configure the ID that is used by the Sybase agent to communicate with Sybase server:

   ```
   1>sp_addlogin user_name, password 2>g
   ```

   Where:

   - *user_name* is the user ID. By default, it is `tivoli`.

     If the user ID is not `tivoli`, edit the `koygrant.sql` file and change the `tivoli` to the correct user ID.
   - *password* is the password of the user.

   Note:

   Location of the `koygrant.sql` file:

   - Windows `\opt\ibm\apm\agent\misc\`
   - UNIX `/opt/ibm/apm/agent/misc/`
4. Run the following command to grant permission to the tables in the database:

   ```
   isql -U sa -P password -S servername -i koygrant_filepathkoygrant.sql
   ```

   Where:

   - *password* is the password of user sa.
   - *servername* is the database server name.
   - *koygrant_filepath* is at the following location:

     Note:

     - Windows `\opt\ibm\apm\agent\misc\`
     - UNIX `/opt/ibm/apm/agent/misc/`
5. Run the following command to create proxy tables that are used for the installed monitor tables:

   ```
   isql -U sa -P password -S servername
        -i $SYBASE/ASE-12_5/scripts/installmontables
   ```

   Where:

   - *password* is the password of user sa.
   - *servername* is the database server name.

What to do next

When the permissions are successfully granted, you can configure the monitoring agent.

Configuring the agent by using the command line interface

You can configure the Monitoring Agent for Sybase Server by using the command line interface.

Before you begin

The Sybase agent does not support remote configuration. Hence, you need to ensure that the Sybase server is installed on the same host where the Sybase agent is installed.

The Sybase agent supports Sybase Server version 15.7 and 16.0 only.

The user ID that is used to connect to the database server is created.

About this task

The Sybase agent is a multiple instance agent, you must configure and start each agent instance manually.

Procedure

1. Run the following command to configure the agent.

   - Windows

     ```
     install_dir\bin\sybase-agent.bat instance_name
     ```

   - UNIX

     ```
     install_dir/bin/sybase-agent.sh instance_name
     ```

   Where:

   - *install_dir* is the agent installation directory.
   - *instance_name* is the Sybase server instance name.

2. When you are prompted to provide values for the following parameters, press Enter to accept the default value, or specify a value and press Enter.

   a) For the `Home Directory` parameter, enter the path of the Sybase server home directory path.

   - Windows

     The example of `Home Directory` is `\opt\sybase`.

   - UNIX

     The example of `Home Directory` is `/opt/sybase`.

   b) For the `ASE Directory` parameter, enter the path of the database server ASE.

   - Windows

     The example of `ASE Directory` is `\opt\sybase\ASE-12_5`.

   - UNIX

     The example of `ASE Directory` is `/opt/sybase/ASE-12_5`.

   c) For the `Open Client Directory` parameter, enter the Sybase open client installation location.

   - Windows

     The example of `Open Client Directory` is `\opt\sap\ocs-16_0`.

   - UNIX

     The example of `Open Client Directory` is `/opt/sap/ocs-16_0`.

   d) For the `USER ID` parameter, enter the user ID that is used by the monitoring agent to connect to the Sybase server.

   The default `USER ID` is `tivoli`.

   e) For the `PASSWORD` parameter, enter the password of the user ID that is used by the monitoring agent to connect to the Sybase server.

   f) For the `VERSION` parameter, enter the Sybase server version.

   The Sybase agent supports Sybase server versions 15.7 and 16.0 only.

   g) For the `ERROR LOG FILE` parameter, enter the fully qualified file name of the error log file for the Sybase server.

   - Windows

     The example of the `ERROR LOG FILE` is `\opt\sap\ASE-16_0\install\`*`servername`*`.log`.

- UNIX

  The example of the ERROR LOG FILE is `/opt/sap/ASE-16_0/install/`*`servername`*`.log`.

  Where *servername* is the Sybase server name.

h) For the EXTENDED parameter, enter the extended parameter that is used by support to exclude certain cursor execution. Optionally, press Enter without specifying any values to execute all cursors.

   The options for the EXTENDED parameter are `DBD2,DBD15,KOYSEGD`.

   - DBD2 will exclude cursor execution for datasets Sybase_Database_Detail and Sybase_Database_Summary.
   - DBD15 will exclude cursor execution for dataset Sybase_Database_Detail.
   - KOYSEGD will exclude cursor execution for dataset Sybase_Segment_Detail.

What to do next

When the configuration is completed, you can start the monitoring agent and connect the monitoring agent to the Performance Management server.

To start the Sybase agent, use the agent command `sybase-agent` command.

To connect the Sybase agent to the Performance Management server, use the `agent2server` command.

For more information about the `sybase-agent` and `agent2server` , see *Using agent commands* in https://www.ibm.com/support/knowledgecenter/SSHLNR_8.1.4/com.ibm.pm.doc/welcome.htm.

Configuring the agent by using the silent response file

You can configure the Monitoring Agent for Sybase Server by using the silent response file.

Before you begin

The Sybase agent does not support remote configuration. Hence, you need to ensure that the Sybase server is installed on the same host where the Sybase agent is installed.

The Sybase agent supports Sybase server version 15.7 and 16.0 only.

The user ID that is used to connect to the database server is created.

About this task

The Sybase agent is a multiple instance agent, you must configure and start each agent instance manually.

You must edit the silent response file and run the agent command to configure the monitoring agent.

Procedure

1. Edit the silent response file.

   - Windows

     Silent response file is at: *`install_dir`*`\samples\sybase_silent_config.txt`.

   - UNIX

     Silent response file is at: *`install_dir`*`/samples/sybase_silent_config.txt`.

   Where *install_dir* is the agent installation directory.

   a) For the `Home Directory` parameter, specify the path of the Sybase server home directory.

      - Windows

        The example of `Home Directory` is `\opt\sybase`.

- UNIX

  The example of Home Directory is /opt/sybase.

b) For the ASE Directory parameter, specify the path of the database server ASE.

- Windows

  The example of ASE Directory is \opt\sybase\ASE-12_5.

- UNIX

  The example of ASE Directory is /opt/sybase/ASE-12_5.

c) For the Open Client Directory parameter, specify the Sybase open client installation location.

- Windows

  The example of Open Client Directory is \opt\sap\ocs-16_0.

- UNIX

  The example of Open Client Directory is /opt/sap/ocs-16_0.

d) For the USER ID parameter, specify the user ID that is used by the monitoring agent to connect to the Sybase server.

The default USER ID is tivoli.

e) For the PASSWORD parameter, specify the password of the user ID that is used by the monitoring agent to connect to the Sybase server.

f) For the VERSION parameter, specify the Sybase server version.

The Sybase agent supports Sybase server versions 15.7 and 16.0 only.

g) For the ERROR LOG FILE parameter, specify the fully qualified file name of the error log file for the Sybase server.

- Windows

  The example of the ERROR LOG FILE is \opt\sap\ASE-16_0\install\*servername*.log.

- UNIX

  The example of the ERROR LOG FILE is /opt/sap/ASE-16_0/install/*servername*.log.

Where *servername* is the Sybase server name.

h) For the EXTENDED parameter, specify the extended parameter that is used by support to exclude certain cursor execution. Optionally, leave it blank to execute all cursors.

The options for the EXTENDED are DBD2,DBD15,KOYSEGD.

- DBD2 will exclude cursor execution for datasets Sybase_Database_Detail and Sybase_Database_Summary.
- DBD15 will exclude cursor execution for dataset Sybase_Database_Detail.
- KOYSEGD will exclude cursor execution for dataset Sybase_Segment_Detail.

2. Save the silent response file.
3. Run the following agent command to configure the monitoring agent.

   - Windows

     ```
     install_dir\bin\sybase-agent.bat config instance_name
      install_dir\samples\sybase_silent_config.txt
     ```

   - UNIX

     ```
     install_dir/bin/sybase-agent.sh config instance_name
      install_dir/samples/sybase_silent_config.txt
     ```

   Where:

- *install_dir* is the agent installation directory.
- *instance_name* is the Sybase server name.

What to do next

When the configuration is completed, you can start the monitoring agent and connect the monitoring agent to the Performance Management server.

To start the Sybase agent, use the **sybase-agent** command.

To connect the Sybase agent to the Performance Management server, use the **agent2server** command.

For more information about the **sybase-agent** and **agent2server** , see *Using agent commands* in https://www.ibm.com/support/knowledgecenter/SSHLNR_8.1.4/com.ibm.pm.doc/welcome.htm.

Disabling dirty reads for query

The Sybase agent enables `dirty reads` for its query execution by default to prevent locking.
The variable `COLL_USE_NOLOCK` is used to enable or disable the query `dirty reads`.
When `dirty reads` is enabled, query is executed with isolation level zero to avoid locking.
If you wish to disable the `dirty reads` for agent query, you can set the variable `COLL_USE_NOLOCK` to zero.

Before you begin

To disable `dirty reads` for agent query, ensure that the agent is installed.

About this task

The Sybase agent enables `dirty reads` by default. To disable the `dirty reads` for the agent query, complete the following steps.

Procedure

1. Stop the agent.
2. Set the variable `COLL_USE_NOLOCK` to zero.
   - UNIX
     a. Add **COLL_USE_NOLOCK=0** in *CANDLEHOME*`/config/.oy.environment` file.
     b. Save and close the file.
   - Windows
     a. Locate the agent instance file *CANDLEHOME*`\TMAITM6_x64\KOYENV_`*INSTANCENAME*.
     b. Add the following line in the file.

        COLL_USE_NOLOCK=0

     c. Save and close the file.

   The *CANDLEHOME* is the agent installation directory.
   The *INSTANCENAME* is the agent instance name.
3. Start the agent.

Configuring Synthetic Playback monitoring

You must configure the Synthetic Playback agent so that the agent can collect data on the availability and performance of internal web applications. This data is displayed in the Application Performance Dashboard.

About this task

Configure the Synthetic Playback agent by running a script and by responding to prompts. Then, start the script and verify that the script is running.

Important: Only existing users of the IBM Website Monitoring on Cloud add-on can install, configure, and run the Synthetic Playback agent. Website Monitoring has been replaced by IBM Cloud Availability Monitoring. For more information, see "About Availability Monitoring" on page 1024.

Procedure

- To configure the agent by running the script and responding to prompts, complete the following steps:
  a) Enter `*install_dir*/bin/synthetic_playback-agent.sh config` where *install_dir* is the Synthetic Playback agent installation directory.
  b) When prompted to `Edit Monitoring Agent for Synthetic Playback settings`, enter 1 to continue.
  c) When prompted to enter a data center name for your playback point of presence, enter a name that identifies the location of your agent.

  Important: Choose a descriptive name for your playback point of presence. When you complete the installation of your agent, you can then select that location by name as a playback location for your synthetic transactions and view transaction data from that location in the Application Performance Dashboard.

  d) When prompted for `Java parameters`, choose a `Java trace level`. Press Enter to choose the default parameter, or enter a number from 1 - 8 to specify a trace level.
  e) When prompted for `Class path for external jars`, press Enter to leave blank, or else specify the location of an external jar.
- To configure the agent by using the silent response file, complete the following steps:
  a) In a text editor, open the `synthetic_playback_silent_config.txt` file that is available at the *install_dir*/samples path.
  For example

  Linux /opt/ibm/apm/agent/samples

  b) In the `synthetic_playback_silent_config.txt` file, uncomment and assign values to the following properties:
  - For `LOCATION`, equate this parameter to the name of your data center or a name describing where your agent is installed.
  - For `JAVA_TRACE_LEVEL`, equate this parameter to one of the listed trace levels, such as `JAVA_TRACE_LEVEL=ERROR`.

  Save the file.

  c) On the command line, change the path to *install_dir*/bin.
  d) Run the following command to configure the agent in silent mode:

```
synthetic_playback-agent.sh config install_dir/samples/
synthetic_playback_silent_config.txt
```

- To start the Synthetic Playback agent, enter: *install_dir*/bin/synthetic_playback-agent.sh start.

- To verify that the Synthetic Playback agent is running, enter: `install_dir/bin/synthetic_playback-agent.sh status`. For more information, see Table 12 on page 180.

What to do next

To view the performance of internal web applications, you must create synthetic transactions in the Synthetic Script Manager. For more information, see "Managing synthetic transactions and events with Website Monitoring" on page 1006.

Enabling upstream proxy support for the Synthetic Playback agent

Enable upstream proxy support for the Synthetic Playback agent to monitor HTTP requests from internal web applications to external web applications.

Before you begin

Ensure that you are running Synthetic Playback agent version 01.00.05.08 or later. To check what agent version you are running, enter `install_dir/bin/cinfo -t` in the command line, where *install_dir* is the installation location of the agent. If you are running any other version of the Synthetic Playback agent, you must download and install IBM Cloud Application Performance Management, Private 8.1.4.0 Synthetic Playback agent interim fix 08 from IBM Fix Central (Enter `Synthetic` in the **Search** field and the list of Synthetic Playback agent interim fixes are displayed). For installation instructions, see 8.1.4.0-IBM-IPM-SYNTHETIC-PLAYBACK-AGENT-IF0008 Readme.

About this task

Internal web applications behind an enterprise firewall require an upstream proxy to access external web resources. Configure the proxy setting of the Synthetic Playback agent to allow your agent to support your upstream proxy so that you can monitor HTTP requests from internal web applications to external web applications.

Procedure

- To configure and enable upstream proxy support for your agent, complete the following steps.

  a) As the root user, configure the proxy settings by running the following commands in the command line.

  ```
  cd install_dir/agent/lx8266/sn/bin
  #./set_proxy.sh
  ```

  When you are prompted, enter the agent installation path, the default path is `/opt/ibm/apm/agent`. Enter the number for the proxy type that you want to configure for your Synthetic Playback agent.

  For example:

  ```
  # cd /install_dir/agent/lx8266/sn/bin/
  #./set_proxy.sh
  please input the agent install path, default is (/opt/ibm/apm/agent)
  agent install path is:/opt/ibm/apm/agent
  please input the number of proxy type:
  1 system proxy
  2 manual proxy
  3 pac proxy
  4 no proxy
  ```

  b) Enter `install_dir/bin/synthetic_playback-agent.sh start` to restart your agent.

- Linux To disable upstream proxy support for your agent, run the `./set_proxy.sh` command again, and select `4 no proxy`. Then, restart your agent.

Configuring Tomcat monitoring

You can configure the Monitoring Agent for Tomcat with the default or custom settings to monitor the resources of Tomcat application servers. The agent can be configured on Windows and Linux systems.

Before you begin

Review the hardware and software prerequisites. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the Tomcat agent.

About this task

The Tomcat agent is a multiple instance agent; you must create the first instance and start the agent manually. The managed system name includes the instance name that you specify, for example, `instance_name:host_name:pc`, where *pc* is your two character product code. The managed system name is limited to 32 characters. The instance name that you specify is limited to 28 characters that excludes the length of your host name. For example, if you specify `TOMCAT2` as your instance name, your managed system name is `TOMCAT2:hostname:OT`. If you specify a long instance name, the managed system name is truncated, and the agent code is not displayed completely.

To avoid permission issues when you configure the agent, be sure to use the same root user or non-root user ID that was used for installing the agent. If you installed your agent as a selected user and want to configure the agent as a different user, see "Configuring agents as a non-root user" on page 183. If you installed and configured your agent as a selected user and want to start the agent as a different user, see "Starting agents as a non-root user" on page 992.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the "Change history" on page 54.

Configuring the Tomcat agent with the default settings

You can use the default settings of the Tomcat agent to monitor the Tomcat server. You do not need to provide more configuration information other than the new instance name.

Before you begin

Before you configure the agent with the default settings, ensure that the following prerequisites are met:

- The agent is installed in the default directory.
- The JMX service URL uses the 8686 port.
- The Tomcat server is configured without the JMX authorization.

About this task

Remember: When you configure the agent with the default settings, the collection of transaction tracking and deep-dive diagnostics data is not enabled.

Procedure

1. Run the following command:

   Linux
   ```
   install_dir/bin/tomcat-agent.sh config instance_name install_dir/samples/tomcat_silent_config.txt
   ```
   Windows
   ```
   install_dir/bin/tomcat-agent.bat config instance_name install_dir/samples/tomcat_silent_config.txt
   ```
   Where

   install_dir

   The installation directory of the Tomcat agent.

instance_name
The name that you want to give to the instance.

2. Run the following command to start the agent:

   Linux
   ```
   install_dir/bin/tomcat-agent.sh start instance_name
   ```
   Windows
   ```
   install_dir/bin/tomcat-agent.bat start instance_name
   ```

What to do next

Log in to the Cloud APM console to view data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring the agent on Windows systems

You can configure the agent on Windows operating systems by using the **IBM Performance Management** window.

Before you begin

Ensure that the following prerequisites are met:

- Java is installed on the Tomcat Server where the agent is installed.
- JDK 1.6 or later version is set on the prompt from where the agent installer is installed.
- JMX Remote is enabled for the Tomcat Server. For details, see Enabling JMX Remote.
- The Tomcat Server is up and running.

About this task

You can configure the agent from the command prompt. For details, follow the steps that are given in the "Configuring the Tomcat agent on Linux systems" on page 798 topic, and run the commands with the .bat extension instead of .sh extension. The following procedure explains configuring the agent by using the agent configuration panel.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Monitoring Agent for Tomcat**.
3. Click **Configure agent**.

   ⚠ **Attention:** If **Configure agent** is unavailable, click **Reconfigure**.
4. In the **Instance Name** window, specify a unique name for the Tomcat agent instance, and click **OK**.

   Restriction: The MSN must not exceed 32 characters.
5. In the **SERVER NAME** field, enter a unique name to identify the Tomcat Server that is being monitored.
6. In the **Java Parameter Settings** window, complete one of the following steps:
   - Click **Next** to accept the default location where Java is installed. The default installation path is `C:\IBM\APM\java\java80_x64\jre`.
   - In the **Java Home** field, specify the path when IBM Java is installed at a different path.
7. In the **JSR-160-Complaint Server** window, specify the details of the following parameters:

   a) In the **JMX user ID** field, specify the ID of the user that is used to connect to the Tomcat MBean server when the JMX authorization is enabled in Tomcat.

   b) In the **JMX password** field, specify the password of the JMX user when the JMX authorization is enabled in Tomcat.

   c) In the **JMX service URL** field, enter the URL that is used for connecting to the Tomcat MBean server.

The format of the URL is `service:jmx:rmi:///jndi/rmi://host_name:port_number/jmxrmi`. The default URL is valid when the server runs on the local host and uses the 8686 port as a JMX port. You can modify the host name and port number in the URL, keeping the same format.

d) From the **Data Collector Configuration** list, select Yes if you want to enable collection of transaction tracking and deep dive data.

8. In the **Monitoring Agent for Tomcat** window, right-click the Tomcat agent instance, and click **Start**.
9. Enable the collection of Transaction Tracking and Deep Dive data and restart the Tomcat Server.

What to do next

If Tomcat agent is running as a service, after configuring the agent on Windows, configure Tomcat Data Collector. For more information, see "Configuring Tomcat Data Collector " on page 797.

Log in to the Cloud APM console to view data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Providing Local Security Policy for running Monitoring Agent for Tomcat on Windows by a Non-Admin user

Local security policies are available to run a Monitoring Agent for Tomcat on Windows by a non-admin user.

About this task

A combination of following two local security policies works to run the Tomcat agent on Windows by a non-admin-user. For a Tomcat agent to start/stop, Configure, and Data verification these two policies work.

1. Debug Programs.
2. Log on as Service.

Follow the procedure that is given to avail the Local Security permissions for a non-administrator user.

Procedure

1. Go to TEMA and change the Tomcat agent startup with non-admin user.
2. Add non-admin user under Tomcat Agent Installation Folder and Give Full Permissions to it.
3. Add non-admin user under Registry key HKEY_LOCAL_MACHINE and click **Full Permissions**.
4. Run the **`secpol.msc`** command in **startmenu** to open the Local Security policies
5. Then, to add Non-admin user in the policies refer "Local Security Policy permissions" on page 796
6. Restart the Tomcat Agent.
7. Check Tomcat Agents status and verify the data on APM portal.

Local Security Policy permissions

Granting Debug Programs permission

About this task

To grant the Debug Programs permission, follow the procedure on Tomcat agent as described here:

Procedure

1. Click **Start > Administrative Tools > Local Security Policy**. The **Local Security Settings** window opens
2. Expand **Local Policies** and click **User Rights Assignment**.The list of user rights opens.
3. Double-click **Debug Programs** policy. The **Debug Programs Properties** window opens.
4. Click **Add User or Group**. The **Select Users or Groups** window opens.

5. In the Enter the object names to select field, enter the user account name to whom you want to assign permissions, and then click **OK**.
6. Click **OK**.

Granting Log on as Service permission

About this task

To grant the Log-on as service permission, follow the procedure on Tomcat agent as described here.

Procedure

1. Click **Start > Administrative Tools > Local Security Policy**. The **Local Security Settings** window opens
2. Expand **Local Policies** and click **User Rights Assignment**.The list of user rights opens.
3. Double-click **Log-on as service** policy. The **Log-on as service Properties** window opens.
4. Click **Add User or Group**. The **Select Users or Groups** window opens.
5. In the Enter the object names to select field, enter the user account name to whom you want to assign permissions, and then click **OK**.
6. Click **OK**.

Configuring Tomcat Data Collector

If Tomcat agent is running as a service, after configuring the agent on Windows, configure Tomcat Data Collector with the instructions given here.

About this task

After configuring and starting Tomcat agent instance, it generates or updates `setenv.bat` file in `/CANDLEHOME/setenv_<instance_name>.bat`. This file contains data collector configuration parameters necessary for configuration of Tomcat Data Collector.

Procedure

1. Open **Apache Tomcat Properties** window, and click **Java**
2. Open `setenv_<instanceName>.bat` from the location `/CANDLEHOME/setenv_<instance_name>.bat`
3. Copy the value of **JAVA_OPTS** parameter from `setenv_<instance_name>.bat` shown in the block:

```
agentlib:am_ibm_16=C:\IBM\APM\otdchome\7.3.0.13.0\runtime\TOMTKWIN1
-Xbootclasspath/p:C:\IBM\APM\otdchome\7.3.0.13.0\toolkit\lib\bcm-bootstrap.jar
-Djava.security.policy=C:\IBM\APM\otdchome\7.3.0.13.0\itcamdc\etc\datacollector.policy
-Dcom.ibm.tivoli.itcam.ai.runtimebuilder.inputs=
C:\IBM\APM\otdchome\7.3.0.13.0\runtime\TOMTKWIN1\TOM_TK_1_DCManual.txt
-Dcom.ibm.tivoli.itcam.serverHome=C:\TOMCAT_9\apache-tomcat-9.0.5\apache-tomcat-9.0.5
-Dam.home=C:\IBM\APM\otdchome\7.3.0.13.0\itcamdc
-Dcom.ibm.tivoli.itcam.toolkit.runtime.dir=C:\IBM\APM\otdchome\7.3.0.13.0\runtime
```

4. Paste this value into the text box labeled **Java Options** in the **Java** tab of **Apache Tomcat Properties**
5. Click **Apply**
6. Go to **Control Panel**, click **System** > **Advanced** > **Environment Variables**
7. In **System variables**, edit the variable *PATH* by appending file path `<OTDC_home>\toolkit\lib\win64;<OTDC_HOME>/ toolkit\lib\win64\ttapi` and click **OK**

   Note: Replace <OTDC_home> with the real path of the toolkit installation directory. For instance, `C:\IBM\APM\otdchome\7.3.0.13.0\toolkit\lib\win64;C:\IBM\APM\otdchome\7.3.0.13.0\toolkit\lib\win64\ttapi`
8. Click **NEW** to add a variable *RUNTIME_DIR*.
9. Add **Variable Name** as *RUNTIME_DIR* and **Variable Path** as `C:\IBM\APM\otdchome\7.3.0.13.0\runtime`. This path is available in `setenv_<instancename>.bat`

10. Restart Windows. Make sure that the Tomcat service startup is set to Automatic

Configuring the Tomcat agent on Linux systems

You run the configuration script and respond to prompts to configure the Tomcat agent on Linux systems.

Before you begin

- JMX Remote is enabled for the Tomcat Server. For details, see Enabling JMX Remote.
- The Tomcat Server is up and running.

Procedure

1. Run the following command:
   `install_dir/bin/tomcat-agent.sh config instance_name`
   Where *instance_name* is the name that you want to give to the instance.
2. When you are prompted to specify a value for SERVER, specify a unique name to identify the Tomcat Server that is being monitored, and press Enter.
3. When you are prompted to specify a value for `Java home`, press Enter to accept the default location where the Java virtual machine is installed. The default location is `/opt/ibm/apm/agent/JRE/lx8266/jre`. If the agent is not installed in the default directory, specify `install_dir/JRE/lx8266/jre`.
4. When you are prompted to specify a value for `JMX user ID`, specify the ID of the user who connects to the Tomcat MBean server. If the JMX authorization is not enabled, press Enter.
5. When you are prompted to specify a value for `JMX password`, specify the password of the JMX user and confirm it. If JMX authorization is not enabled, press Enter.
6. When you are prompted to specify a value for `JMX service URL`, press Enter to accept the default URL or specify another service URL for connecting to the Tomcat MBean server.
   The format of the URL is `service:jmx:rmi:///jndi/rmi://host_name:port_number/jmxrmi`. The default URL is valid when the server runs on the local host and uses the 8686 port as a JMX port. You can modify the host name and the port in the URL, keeping the same format.
7. When you are prompted to specify a value for `Data Collector Configuration`, specify 1 and press Enter to enable collection of transaction tracking and deep dive data.
8. Run the following command to start the agent:
   `install_dir/bin/tomcat-agent.sh start instance_name`
9. Enable the collection of Transaction Tracking and Deep Dive data, restart the Tomcat Server.

What to do next

Log in to the Cloud APM console to view data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Configuring the Tomcat agent by using the silent response file

You can use the silent response file to configure the Tomcat agent without responding to prompts.

Procedure

1. In a text editor, open the `tomcat_silent_config.txt` file that is available at the following path: `install_dir/samples`
2. For the **KOT_SERVER** parameter, specify a unique name to identify the Tomcat Server that is being monitored.
3. For the **Java home** parameter, specify the path where the Java virtual machine is installed. The default location is `/opt/ibm/apm/agent/JRE/lx8266/jre`. If the agent is not installed in the default directory, specify `install_dir/JRE/lx8266/jre`.

4. For the **`JMX user ID`** parameter, specify the ID of the user that is used to connect to the Tomcat MBean server. You must specify a value for this parameter when the JMX authorization is enabled in Tomcat.
5. For the **`JMX password`** parameter, specify the password of the JMX user. You must specify a value for this parameter when the JMX authorization is enabled in Tomcat.
6. For the **`JMX service URL`** parameter, specify the service URL for connecting to the Tomcat MBean server. The format of the URL is `service:jmx:rmi:///jndi/rmi://`*`host_name`*`:`*`port_number`*`/jmxrmi`. The default URL is valid when the server runs on the local host and uses the 8686 port as a JMX port. You can modify the host name and the port number in the URL, keeping the same format.
7. For the **`KOT_DCCONFIGURATION`** parameter, specify Yes if you want to enable collection of transaction tracking and deep dive data.
8. Save and close the `tomcat_silent_config.txt` file, and run the following command to update the agent configuration settings:

   Linux *`install_dir`*`/bin/tomcat-agent.sh config` *`instance_name install_dir`*`/samples/tomcat_silent_config.txt`

   Windows *`install_dir`*`/bin/tomcat-agent.bat config` *`instance_name install_dir`*`/samples/tomcat_silent_config.txt`

   Where *instance_name* is the name that you want to give to the instance, and *install_dir* is the installation directory of the Tomcat agent.
9. Run the following command to start the agent:

   Linux *`install_dir`*`/bin/tomcat-agent.sh start` *`instance_name`*

   Windows *`install_dir`*`/bin/tomcat-agent.bat start` *`instance_name`*
10. If you enable the collection of transaction tracking and deep dive data, restart the Tomcat Server.

What to do next

Log in to the Cloud APM console to view data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

Enabling the collection of transaction tracking and diagnostics data

On the **Agent Configuration** page, you can enable or disable the collection of transaction tracking and diagnostics data.

About this task

When you enable the collection of transaction tracking data, the agent collects data of the following components:

- Servlet JSP
- EJB applications
- JMS

Procedure

Complete the following steps to configure the data collection for each managed system.

1. Log in to the Cloud APM console.
2. From the navigation bar, click **System Configuration** > **Agent Configuration**.

   The **Agent Configuration** page is displayed.
3. Click the **Tomcat** tab.
4. Select the check boxes of the managed systems for which you want to configure the data collection and complete any of the following actions from the **Actions** list.

   - To enable transaction tracking, click **Set Transaction Tracking** > **Enabled**. The status in the **Transaction Tracking** column is updated to Enabled for each selected managed system.

- To enable the diagnostic data collection, select **Set Diagnostic Mode** > **Enabled Diagnostic Mode Only**. The status in the Diagnostic Mode column is updated to Enabled for each selected managed system.
- To enable the diagnostic data collection and method trace, select **Set Diagnostic Mode** > **Enabled Diagnostic Mode and Method Trace**. The status in the Diagnostic Mode and Method Trace columns is updated to Enabled for each selected managed system.
- To disable transaction tracking, click **Set Transaction Tracking** > **Disabled**. The status in the **Transaction Tracking** column is updated to Disabled for each selected managed system.
- To disable diagnostic data collection, click **Set Diagnostic Mode** > **Disabled Diagnostic Mode and Method Trace**. The status in the **Diagnostic Mode and Method Trace** columns is updated to Disabled for each selected managed system.

What to do next

- Log in to the Cloud APM console to view the transaction tracking and diagnostics data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.
- Review the data that is presented in the Tomcat agent dashboard pages. For more information, see "Group and Instance - Application Performance Dashboard" on page 1063.
- The **Method traces** widget lists the method names, their stack trace, and their response times. To optimize response time, the number of methods are limited by the threshold value for `response time` (default is 50 milliseconds) taken by each method call. Thus, you do not see the method traces for the responses that complete within 50 milliseconds. If you prefer to see all method calls in the Method traces widget, you can update the agent configuration:

  1. Open the /<*agent_installation*>/otdcchome/<*toolkit_version*>/gdc/etc/appMethods.xml file in a text editor.
  2. Locate the **`collectContextData`** parameter. These are the default values:

     ```
     <collectContextData>false</collectContextData>
        <collectStackTrace>ifThresholdExceeded</collectStackTrace>
        <perfThreshold>50</perfThreshold>
        <secondaryPerfThreshold>200</secondaryPerfThreshold>
        <dataToCollect>instanceAndSummary</dataToCollect>
        <createDataRow>ifThresholdExceeded</createDataRow>
        <showParentMethodId>true</showParentMethodId>
     ```
  3. Set `<collectStackTrace>` to `true`.
  4. Set `<createDataRow>` to `true`.
  5. Restart the Tomcat server.

  The threshold limit affects the behavior of the Method traces widget. Therefore, the total response time might not be the sum of the method trace that is displayed on the dashboard.

  Note: Diagnostics must be enabled for the `<collectStackTrace>` true setting to take effect. For more information, see "Configuring collection of detailed diagnostic information" on page 800 and "Customizing the request thresholds" on page 802.

Configuring collection of detailed diagnostic information

If you have IBM Cloud Application Performance Management, Advanced, you can use the data collector to collect detailed diagnostic information on the monitored application server instance. To configure the behavior of the diagnostic data collection, including the amount of diagnostic information that the data collector stores, customize the `gdc_custom.properties` configuration file.

About this task

You can find the `gdc_custom.properties` file in the *dc_home*`/runtime/`*<server_name>*`/custom/gdc` directory.

The following examples describe how to use the properties in the `gdc_custom.properties` configuration file to do the following things:

- Setting limits for the size and number of detailed information files
- Setting full or partial collection of request and method diagnostic data

You can also set other properties in the `gdc_custom.properties` file to customize collection of diagnostic data. Refer to the comments in the file that describe the properties.

Remember: After you complete editing the `gdc_custom.properties` file, you must restart the monitored application server instance for the changes to take effect.

Setting limits for the size and number of detailed information files

About this task

The data collector stores diagnostic information in a number of files. By default, it stores 100 files; if 100 files are already stored and a new file is created, the oldest file is deleted. The data collector creates a new file every 15 minutes, or when the size of the current file exceeds 200 megabytes. When the total size of the directory that contains the files exceeds 2 gigabytes, the data collector deletes the oldest file.

Procedure

You can change the following settings in the *dc_home*`/runtime/`*<server_name>*`/custom/gdc/gdc_custom.properties` file:

- To set the maximum number of files with diagnostic information, set the `com.ibm.itcam.gdc.dfe.filelimit` property.
  For example:

  ```
  com.ibm.itcam.gdc.dfe.filelimit=100
  ```

- To set the time, in minutes, after which the data collector creates a new diagnostic data file, set the `com.ibm.itcam.gdc.dfe.frequency` property.
  For example:

  ```
  com.ibm.itcam.gdc.dfe.frequency=15
  ```

- To set the maximum diagnostic data file size, in megabytes, set the `dfe.file.maxlimit` property.
  For example:

  ```
  dfe.file.maxlimit=200
  ```

  If the current diagnostic data file reaches this size, the data collector creates a new diagnostic data file.

- To set the maximum total size of all data files, in bytes, set the `trace.dir.size.limit` property.
  For example:

  ```
  trace.dir.size.limit=2147483648
  ```

  If the sum of the sizes of all the diagnostic data files exceeds this value, the data collector deletes the oldest data file. The minimum total size is 25 megabytes.

Setting full or partial collection of request and method diagnostic data

About this task

The data collector has the following default settings:

- The data collector collects diagnostic data only for the selected requests. The selection (sampling) of the requests aims to include all errors and some good requests.
- Method data collection is disabled at server startup.
- When method data collection is enabled, the data collector gathers method data only for some requests (of those for which diagnostic data is collected). This further selection (sampling) again aims to include all errors and some good requests.

Important: Changing these settings affects performance of the application server. On production servers, the performance degradation might be critical.

Procedure

You can change these settings by using properties in the *dc_home*/runtime/*<server_name>*/custom/gdc/gdc_custom.properties file.

- To enable method data collection, set the property as follows:

  ```
  dfe.enable.methoddata=true
  ```

  Tip: You can also use the **Agent Configuration** page to dynamically enable or disable the method trace data collection.
- To collect diagnostic data for every request, disable the sampling by setting the property as follows:

  ```
  dc.sampling.enable=false
  ```

- To enable method data collection for every request for which diagnostic data is collected, set the property as follows:

  ```
  dc.sampling.enable=false
  dc.sampling.methsampler.enabled=false
  ```

  Remember: The dc.sampling.methsampler.enabled property takes effect only when method data collection is enabled on the Agent Configuration page or by the dfe.enable.methoddata property.

Customizing the request thresholds

Some of the requests might not have enough information if the default thresholds are high. You can customize the request thresholds so that more requests or request context data can be captured by the data collector.

About this task

Each request type has two threshold types, which are named **perfThreshold** and **secondaryPerfThreshold**. A request is captured by the data collector only when it takes more time than what is specified for the **perfThreshold** threshold. Context data, such as stack trace and SQL statement, is captured only when the request takes more time than what is specified for the **secondaryPerfThreshold** threshold. You can adjust these threshold values to suit your needs.

Procedure

1. Go to the etc directory of the Tomcat agent.
   - Linux AIX /<dc_home>/gdc/etc
   - Windows \<dc_home>\gdc\etc

   where *<dc_home>* is the installation directory of the Tomcat agent. The default is C:\IBM\APM\otdchome on Windows systems and /opt/ibm/apm/agent/otdchome on Linux systems.
2. In a text editor, open the XML file for the request type that you want to customize. You can tell which file is for which request type from the XML file name.

For example, the `servlet.xml` file is for servlet requests, the `custom.xml` file is for custom requests, and the `appMethods.xml` file is for class and methods when method trace is enabled.

3. Set the `<collectContextData>`, `<collectStackTrace>`, and `<createDataRow>` tags to `ifThresholdExceeded`.

   ```
   <collectContextData>ifThresholdExceeded</collectContextData>
   <collectStackTrace>ifThresholdExceeded</collectStackTrace>
   <createDataRow>ifThresholdExceeded</createDataRow>
   ```

4. Set the `<perfThreshold>` tags to the threshold value that you desire. The unit of the threshold is millisecond.
   For example, the `servlet.xml` file has the following settings for servlet requests. As a result, only the servlet requests that take more than 2000 milliseconds are captured by the data collector if *createDataRow* is set to `ifThresholdExceeded`.

   ```
   <collectContextData>true</collectContextData>
   <collectStackTrace>ifThresholdExceeded</collectStackTrace>
     <perfThreshold>2000</perfThreshold>
   <dataToCollect>instanceAndSummary</dataToCollect>
   <createDataRow>ifThresholdExceeded</createDataRow>
   ```

5. You can update the configuration for threshold check and it will force all method calls to be visible on APM dashboard.

   Edit the `dc_home/gdc/etc/appMethods.xml` file as given below:

   a. Set **`<collectContextData>`** to true
   b. Set **`<collectStackTrace>`** to true
   c. Set **`<createDataRow>`** to true

   Updating these configurations will result in an overall performance impact on web-applications which are deployed on Tomcat server.

6. Save your desired changes and restart the application server.

Update or Change Tomcat Application Server

To update or change Tomcat application server after Tomcat agent configuration, follow the steps given in this topic. These steps are common for both, Tomcat configured through Windows and Linux.

Procedure

1. Stop Tomcat agent instance and Tomcat Server
2. Go to `<TOMCAT_SERVER>/bin` and open `setenv.sh` file in editor
3. Remove all the startup parameters for Data Collector from `setenv.sh`. Remove following lines from the file

   ```
   export LD_LIBRARY_PATH="<CANDLE_HOME>/otdchome/7.3.0.13.0/toolkit/lib/lx8266"
   export RUNTIME_DIR="<CANDLE_HOME>/otdchome/7.3.0.13.0/runtime"
   export JAVA_OPTS="-agentlib:am_ibm_16=<CANDLE_HOME>/otdchome/7.3.0.13.0/runtime/
   <Tomcat_Application_
   Server> -
   Xbootclasspath/p:<CANDLE_HOME>/otdchome/7.3.0.13.0/toolkit/lib/bcm-bootstrap.jar -
   Djava.security.policy=<CANDLE_HOME>/otdchome/7.3.0.13.0/itcamdc/etc/datacollector.policy -
   Dcom.ibm.tivoli.itcam.ai.runtimebuilder.inputs=<CANDLE_HOME>/otdchome/7.3.0.13.0/runtime/
   <Tomcat_
   Application_Server>/<Agent_Instance>_DCManual.txt -
   Dcom.ibm.tivoli.itcam.serverHome=<TOMCAT_HOME> -
   Dam.home=<CANDLE_HOME>/otdchome/7.3.0.13.0/itcamdc -
   Dcom.ibm.tivoli.itcam.toolkit.runtime.dir=<CANDLE_HOME>/otdchome/7.3.0.13.0/runtime"
   ```

4. Save the changes and start Tomcat Server
5. Reconfigure the Tomcat agent to update or change Tomcat Application Server
6. Update or Change only the Tomcat Application Server and do not change any the other configuration setting
7. Start Tomcat agent instance

8. Check `setenv.sh` file is updated with new Tomcat Application Server in startup parameters for Data Collector
9. Restart Tomcat Server
10. Verify the changes made to Tomcat Application Server are reflected in Agent Machine and IBM Cloud Application Performance Management dashboard

    - Verify Tomcat Application Server change at `<CANDLE_HOME>/otdchome/7.3.0.13.0/runtime/<Tomcat_Application_Server>` location on Agent machine
    - Verify Tomcat Application Server change on **Aggregate Transaction Topology** page and *appserver* attribute under *KOT_Server* Attribute Group on IBM Cloud Application Performance Management dashboard

Prerequisites for upgrading Tomcat APM agent to latest toolkit build 7.3.0.15.0

To upgrade the Tomcat agent to toolkit build 7.3.0.15.0, you must ensure the prerequisites given in this topic are complete.

Procedure

1. Before agent upgrade, remove the DC startup parameters from `setenv.sh/setenv.bat` file. Remove following DC startup parameters from `setenv.sh/setenv.bat` file.

```
export LD_LIBRARY_PATH="/opt/ibm/apm/agent/otdchome/7.3.0.13.0/toolkit/lib/
lx8266":$LD_LIBRARY_PATH
export RUNTIME_DIR="/opt/ibm/apm/agent/otdchome/7.3.0.13.0/runtime":$RUNTIME_DIR
export JAVA_OPTS="-agentlib:am_ibm_16=<Agent_Install_Dir>/otdchome/7.3.0.13.0/runtime/
<Tomcat_Application_Server>
-Xbootclasspath/p:<Agent_Install_Dir>/otdchome/7.3.0.13.0/toolkit/lib/bcm-bootstrap.jar
-Djava.security. policy=<Agent_Install_Dir>/otdchome/7.3.0.13.0/itcamdc/etc/
datacollector.policy
-Dcom.ibm.tivoli.itcam.ai.runtimebuilder.inputs=<Agent_Install_Dir>/otdchome/7.3.0.13.0/
runtime/<Tomcat_Application_Server>/<Tomcat_Agent_Instance>_DCManual.txt
-Dcom.ibm.tivoli. itcam.serverHome=<Tomcat_Server>/apache-tomcat-<Version>
-Dam.home=<Agent_Install_Dir>/otdchome/7.3.0.13.0/itcamdc
-Dcom.ibm.tivoli.itcam.toolkit.runtime.dir=<Agent_Install_Dir>/otdchome/7.3.0.13.0/
runtime":$JAVA_OPTS
```

2. Restart the server.
3. Stop the server.
4. Upgrade the agent with new upgraded toolkit build (7.3.0.15.) build
5. Verify the agent version and required `/otdchome` and `otdchome_backup` folders.
6. Start the Tomcat Server
7. Restart the Tomcat agent
8. Verify that the `setenv.sh/setenv.bat` has the upgraded toolkit 7.3.0.15.0 DC startup parameters.
9. Restart the Tomcat Server. Server should stop properly without any issues
10. Verify the End-To-End agent functionality.

Configuring VMware VI monitoring

After installing the Monitoring Agent for VMware VI, you must create the first instance, and manually start the agent so that the agent can collect data of the VMware Virtual Infrastructure that is being monitored.

Before you begin

- Review the hardware and software prerequisites.
- Create a user ID in your VMware Virtual Infrastructure. The agent uses this user ID to connect to the VMware vCenter for monitoring the VMware Virtual Infrastructure. Ensure that you have the "System.View" and "System.Read" privileges on all the vCenters and ESX servers that are being monitored. For information about how to create the user ID, see the VMware documentation for managing users, groups, permissions, and roles.

- Determine whether the vCenter is configured for SSL communication. If it is configured, then you must configure the VMware VI agent to use SSL for communicating with the vCenter.
  - To determine whether the vCenter uses SSL for communication, use the `https://` *`vCenterIPaddress`* URL to access the vCenter. If you can access the vCenter, then it indicates that the vCenter uses SSL to communicate over the network.
  - To configure the VMware VI agent to use SSL for communicating with the vCenter, complete the steps that are described in "Enabling SSL communication with VMware VI data sources" on page 806.
- Decide the number of agent instances that you need to monitor your VMware Virtual Infrastructure. For information about sizing the agent instances according to your monitoring environment, see "Sizing and planning the VMware VI agent deployment" on page 805.

About this task

The VMware VI agent is a multiple instance agent. Unlike a single instance agent, for which you can configure the agent to monitor and collect data for only one monitored application, the VMware VI agent can have multiple configured instances that connect to multiple vCenter servers and remotely monitor your VMware Virtual Infrastructure.

The configuration parameters define the VMware VI data sources that are monitored and define a connection to either the VMware vCenter, vCenter Server Appliance, or to an individual VMware ESX server. To know the supported versions of these applications, see the Software Product Compatibility Reports for the VMware VI agent.

The product version and the agent version often differ. The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see the "Change history" on page 54.

You must manually configure the agent to view data for all the agent attributes.

- To configure the agent on Windows operating systems, you can use the **IBM Performance Management** window or the silent response file.
- To configure the agent on Linux operating systems, you can run the script and respond to prompts, or use the silent response file.

Sizing and planning the VMware VI agent deployment

The number of agent instances that you can configure on a single system depends on the availability and utilization of resources on the system.

The following table categorizes the VMware environment into various sizes with the required Java heap size:

Table 223. VMware environment and Java heap size

VMware environment size	Number of ESX servers	Java heap size
Small environment	A vCenter server that manages up to 125 ESX(i) servers and 300 - 1500 guests.	**`-Xmx2048m`** (2 GB)
Medium environment	A vCenter server that manages between 125 - 250 ESX(i) servers and 1500 - 4000 guests.	**`-Xmx4096m`** (4 GB)
Large environment	A vCenter server that manages between 250 - 500 ESX(i) servers and 4000 - 7500 guests.	**`-Xmx8192m`** (8 GB)

Table 223. VMware environment and Java heap size (continued)

VMware environment size	Number of ESX servers	Java heap size
Very large environment	A vCenter server that manages more than 500 ESX(i) servers and more than 7500 guests.	**`-Xmx16384m`** (16 GB)

To increase the heap size for the Java data provider, complete the steps that are described in "Increasing the Java heap size" on page 811.

For the agent instances to successfully monitor the environment, the server on which you install the agent, must have adequate memory resources to accommodate the data that is collected by these agent instances. A single instance of the VMware VI agent requires approximately 300 - 400 MB to monitor a small environment. See the following guidelines about the number of agent instances to be configured:

- Use a single instance to monitor a single vCenter. Do not use the same instance to monitor multiple vCenters.
- In a non-cluster environment, use a single instance to monitor a maximum of 8 small ESX servers (100 - 200 virtual machines in one ESX server). Do not configure multiple individual ESX servers under the single agent instance.
- Use multiple agent instances of the VMware VI agent to monitor an environment that contains multiple vCenters. Before you configure multiple instances, ensure that you have adequate memory resources on the system where you install the agent.

Enabling SSL communication with VMware VI data sources

Before you configure the agent to securely communicate with its VMware VI data sources by using SSL, you must add a data source SSL certificate to the certificate truststore of the agent.

About this task

Important: The following information applies only if the agent is configured to validate SSL certificates.

If the SSL certificate validation is turned off, the VMware VI agent connects to VMware data sources even if their SSL certificates are expired, untrusted, or invalid. However, turning off SSL certificate validation is potentially not secure and must be done with care.

If a VMware data source uses an SSL certificate that is signed by a common Certificate Authority (for example, Verisign, Entrust, or Thawte), then it is not necessary to add certificates to the VMware VI agent certificate truststore. However, if the data source uses a certificate that is not signed by a common Certificate Authority, as is the case by default, you must add the certificate to the truststore to allow the agent to successfully connect and collect data.

Note:

1. The default VMware certificate file is named `rui.crt`.
2. For a Virtual Center, the SSL certificate file is located by default in the following path:

   ```
   C:\Documents and Settings\All Users\Application Data\VMware\VMware
   VirtualCenter\SSL
   ```
3. For an ESX server, the SSL certificate file is located by default in the `/etc/vmware/ssl` directory.

Procedure

1. Copy the certificate file from your data source to the agent computer.
2. On the agent computer, place the certificate file in a directory of your choice. Do not overwrite the certificate files. Use a unique file name and a label for each certificate that you add.
3. Use the *keytool* command to add the data source certificate to the certificate truststore of the agent:

```
keytool -import -noprompt -trustcacerts -alias CertificateAlias -file
CertificateFile -keystore Truststore -storepass TruststorePassword
```

Where

CertificateAlias

Unique reference for each certificate added to the certificate truststore of the agent, for example, an appropriate alias for the certificate from *datasource.example.com* is *datasource*.

CertificateFile

Complete path and file name to the VMware data source certificate to add to the truststore.

Truststore

Complete path and file name to the VMware VI agent certificate database. Use the following path and file name:

- Windows (64 bit): `install_dir\tmaitm6_x64\kvm.truststore`
- Linux (64 bit): `install_dir/lx8266/vm/etc/kvm.truststore`

TruststorePassword

ITMVMWAREVI is the default password for the VMware VI agent truststore. To change this password, consult the Java Runtime documentation for information about the tools to use.

Important: To use the *keytool* command, the Java Runtime bin directory must be in your path. Use the following commands:

- Windows (64 bit): `set PATH=%PATH%;install_dir\java\java70_x64\jre\bin`
- Linux (64 bit): `PATH="$PATH":/opt/ibm/apm/agent/JRE/lx8266/bin`

4. After you add all the data source certificates, start the monitoring agent.

What to do next

Complete the agent configuration.

Configuring the agent on Windows systems

You can configure the agent on Windows operating systems by using the **IBM Performance Management** window. After you update the configuration values, you must start the agent to save the updated values.

About this task

The VMware VI agent provides default values for some parameters. You can specify different values for these parameters.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**.
2. In the **IBM Performance Management** window, right-click **Monitoring Agent for VMware VI**, and then click **Configure agent**.

   Remember: After you configure the agent for the first time, the **Configure agent** option is disabled. To configure the agent again, click **Reconfigure**.
3. In the Monitoring Agent for VMware VI window, complete the following steps:

   a) Enter a unique name for the VMware VI agent instance, and click **OK**.

   b) On the **Data Provider** tab, specify values for the configuration parameters, and then click **Next**.

   c) On the **Data Source** tab, specify values for the configuration parameters, and then click **Next**.

   The VMware VI agent is a multi-data source agent. You can monitor multiple data sources from the same agent.

    - If you want to configure a new data source, click **New**.
    - If you want to delete an existing data source, click **Delete**.

    For information about the configuration parameters in each tab of the Monitoring Agent for VMware VI window, see the following topics:

    - Configuration parameters for the data provider
    - Configuration parameters for the data source

4. In the **IBM Performance Management** window, right-click the instance that you configured, and then click **Start**.

What to do next

- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

  If you need help with troubleshooting, see the Troubleshooting section.
- If you are monitoring a large VMware environment with more than 500 ESX hosts, you might need to increase the heap size for the Java data provider. For more information, see "Increasing the Java heap size" on page 811.

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

The silent response file contains the agent configuration parameters with default values that are defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

- To configure the VMware VI agent in the silent mode, complete the following steps:

  a) In a text editor, open the `vmware_vi_silent_config.txt` file that is available at the following path:

    - Linux *install_dir*`/samples/vmware_vi_silent_config.txt`

      Example `/opt/ibm/apm/agent/samples/vmware_vi_silent_config.txt`
    - Windows *install_dir*`\samples\vmware_vi_silent_config.txt`

      Example `C:\IBM\APM\samples\vmware_vi_silent_config.txt`

  b) In the `vmware_vi_silent_config.txt` file, specify values for all mandatory parameters. You can also modify the default values of other parameters.

  For information about the configuration parameters, see the following topics:

    - "Configuration parameters for the data provider" on page 811
    - "Configuration parameters for the data source" on page 810

  c) Save and close the `vmware_vi_silent_config.txt` file, and run the following command:

    - Linux *install_dir*`/bin/vmware_vi-agent.sh config` *instance_name* *install_dir*`/samples/vmware_vi_silent_config.txt`

Example **/opt/ibm/apm/agent/bin/vmware_vi-agent.sh config instance_name /opt/ibm/apm/agent/samples/vmware_vi_silent_config.txt**

- Windows *install_dir*\bin\vmware_vi-agent.bat config *instance_name* *install_dir*\samples\vmware_vi_silent_config.txt

  Example **C:\IBM\APM\bin\ vmware_vi-agent.bat config instance_name C:\IBM \APM\samples\vmware_vi_silent_config.txt**

  Where

  instance_name
  Name that you want to give to the instance.

  install_dir
  Path where the agent is installed.

Important: Ensure that you include the absolute path to the silent response file. Otherwise, the agent data is not shown in the dashboards.

d) Run the following command to start the agent:

- Linux *install_dir*/bin/vmware_vi-agent.sh start *instance_name*

  Example **/opt/ibm/apm/agent/bin/vmware_vi-agent.sh start instance_name**

- Windows *install_dir*\bin\vmware_vi-agent.bat start *instance_name*

  Example **C:\IBM\APM\bin\vmware_vi-agent.bat start instance_name**

What to do next

- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

  If you need help with troubleshooting, see the Troubleshooting section.

- If you are monitoring a large VMware environment with more than 500 ESX hosts, you might need to increase the heap size for the Java™ data provider. For more information, see "Increasing the Java heap size" on page 811.

Configuring the agent by responding to prompts

To configure the agent on Linux operating systems, you must run the script and respond to prompts.

Procedure

- To configure the agent by running the script and responding to prompts, complete the following steps:

  a) On the command line, run the following command:

  install_dir/bin/vmware_vi-agent.sh config *instance_name*

  Example **/opt/ibm/apm/agent/bin/vmware_vi-agent.sh config instance_name**

  Where

  instance_name
  Name that you want to give to the instance.

  install_dir
  Path where the agent is installed.

  b) Respond to the prompts by referring to the following topics:

  - "Configuration parameters for the data provider" on page 811
  - "Configuration parameters for the data source" on page 810

  c) Run the following command to start the agent:

  install_dir/bin/vmware_vi-agent.sh start *instance_name*

Example **`/opt/ibm/apm/agent/bin/vmware_vi-agent.sh start instance_name`**

What to do next

- Log in to the Cloud APM console to view the data that is collected by the agent in the dashboards. For information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

  If you need help with troubleshooting, see the Troubleshooting section.
- If you are monitoring a large VMware environment with more than 500 ESX hosts, you might need to increase the heap size for the Java™ data provider. For more information, see "Increasing the Java heap size" on page 811.

Configuration parameters for the data source

When you configure the VMware VI agent, you can change the default values of the parameters for the data source, such as the address, user id, and password of the data source.

The following table contains detailed descriptions of the configuration parameters for the data source.

Table 224. Names and descriptions of the configuration parameters for the data source

Parameter name	Description	Mandatory field
Data Source ID	The ID of the data source.	Yes
Data Source Address	Address of the data source. If you do not want the agent to validate the SSL certificates, set the value to the host name or IP address of the VMware Virtual Center or ESX server that is being monitored. If you want the agent to validate the SSL certificates when using SSL to communicate over the network, configure the agent by using the Subject Alternative Name that is provided in the certificate. To view the subject alternative name of the data center, complete the following steps: 1. Open the certificate. 2. In the **Certificate** window, click the **Details** tab. 3. Select **Subject Alternative Name**, and use the value of DNS Name. For example, if the value of DNS Name is "`ibmesx3v3vc.ITMfVS.com`", then use the "`ibmesx3v3vc.ITMfVS.com`" value for the host name.	Yes
Use SSL Connection to Data Source	Indicates whether the agent uses an SSL connection to connect to the data sources of the VMware Virtual Infrastructure. Specify `Yes` if the agent uses an SSL connection to connect to the data sources. Otherwise, specify No. The default value is `Yes`.	Yes
Data Source User ID	The user ID that has sufficient privileges to collect monitoring data, and is known to the data source.	Yes
Data Source Password	The password of the user ID that is configured for accessing the data source.	Yes
Confirm Data Source Password	The same password that you specified in the **Data Source Password** field.	

Configuration parameters for the data provider

When you configure the VMware VI agent, you can change the default values of the parameters for the data provider, such as the maximum number of data provider log files, the maximum size of the log file, and the level of detail that is included in the log file.

The following table contains detailed descriptions of the configuration parameters for the data provider.

Table 225. Names and descriptions of the configuration parameters for the data provider

Parameter name	Description	Mandatory field
Instance Name	The name of the instance. **Restriction:** The **Instance Name** field displays the name of the instance that you specify when you configure the agent for the first time. When you configure the agent again, you cannot change the instance name of the agent.	Yes
Valid SSL Certificates	Indicates whether the agent validates SSL certificates when the agent uses SSL to communicate over the network. Set the value to `Yes` if you want the agent to validate SSL certificates when the agent uses SSL to communicate over the network. Set the value to No to prevent the agent from validating SSL certificates. The default value is `Yes`. For information about adding a data source SSL certificate to the certificate truststore of the agent, see "Enabling SSL communication with VMware VI data sources" on page 806.	Yes
Maximum number of Data Provider Log Files	The maximum number of log files that the data provider creates before it overwrites the previous log files. The default value is 10.	Yes
Maximum Size in KB of Each Data Provider Log	The maximum size in KB that a data provider log file must reach before the data provider creates a new log file. The default value is 5190 KB.	Yes
Level of Detail in Data Provider Log	The level of detail that can be included in the log file that the data provider creates. The default value is INFO. The following values are valid: OFF, SEVERE, WARNING, INFO, FINE, FINER, FINEST, and ALL.	Yes
`KEY_STORE_PASSWORD`	The **`KEY_STORE_PASSWORD`** allows user to configure the agent with new key-store password set for the agent JRE. Please note that this Java key-store has no relation with the key-store of vCenter. It is not mandatory to enter the password on each configuration. If this field is left blank, the agent assumes that the default Java key-store password should be used while using the agent JRE.	No

Increasing the Java heap size

After you configure the VMware VI agent, if you are monitoring a large VMware Virtual Infrastructure environment, then you might need to increase the heap size for the Java™ data provider.

About this task

The default maximum heap size for the Java data provider is 256 megabytes. You must set the maximum heap size to an appropriate value that depends on the size of the VMware environment. For information about the heap sizes that are required for the various VMware environments, see Table 223 on page 805.

Important: The system, on which you install and configure the VMware VI agent, must have adequate memory space to accommodate the required heap size.

If any of the following problems arise, then you might need to increase the heap size:

- The Java data provider stops because of a `javacore` problem, and creates a file that is named `javacore.`*`date.time.number`*`.txt` in the `CANDLEHOME\tmaitm6_x64` directory.
- The `javacore.`*`date.time.number`*`.txt` file contains the string `java/lang/OutOfMemoryError`.

Procedure

- Windows

  Complete the following steps to set a value of 1 GB as the heap size:

  1. Open the `%CANDLE_HOME%\TMAITM6_x64\kvm_data_provider.bat` file.
  2. Add the following line before the line that starts with `KVM_JVM_ARGS="%KVM_CUSTOM_JVM_ARGS...`:

     ```
     SET KVM_CUSTOM_JVM_ARGS=-Xmx1024m
     ```

  3. Restart the agent.

- Linux

  Complete the following steps to set a value of 1 GB as heap size:

  1. Open the `$CANDLEHOME/lx8266/vm/bin/kvm_data_provider.sh` file.
  2. Add the following line before the line that starts with `KVM_JVM_ARGS="$KVM_CUSTOM_JVM_ARGS...`:

     ```
     KVM_CUSTOM_JVM_ARGS=-Xmx1024m
     ```

  3. Restart the agent.

Configuring Environment Variables

You can configure environment variables to change the behavior of the agent.

About this task

Note: For Windows platform, user can edit the environment variable that will change the behavior of the specific agent instance.
For non-Windows platform, environment variable setting will impact all the running agent instances.

Procedure

1. Stop all the agent instances.
2. Locate the environment variable file.
   - Windows:

     Locate the KVMENV_*instance_name* file by navigating to the agent folder, where *instance_name* is the agent instance name.

     - Agent of 32-bit system: `%CANDLEHOME%\TMAITM6`
     - Agent of 64-bit system:`%CANDLEHOME%\TMAITM6_x64`
   - Non-Windows:

     Locate the `.vm.environment` file by navigating to the agent folder.

     - Agent of 32-bit system: `$CANDLEHOME/config`
     - Agent of 64-bit system: `$CANDLEHOME/config`
3. Edit environment variable according to the requirement and save the file.

Example:

```
KVM_DATA_PROVIDER_CONNECTION_RETRY_COUNT=0
```

4. Start the agent instance.

Configuring WebLogic monitoring

The Monitoring Agent for WebLogic provides you with a central point of monitoring for the health, availability, and performance of your WebLogic server environment. The agent displays a comprehensive set of metrics to help you make informed decisions about your WebLogic resources, including Java virtual machines (JVMs), Java messaging service (JMS), Java Database Connectivity (JDBC).

Before you begin

- The directions here are for the most current release of the agent, except as indicated.
- Make sure that the system requirements for the WebLogic agent are met in your environment. For the up-to-date system requirement information, see the Software Product Compatibility Reports (SPCR) for the WebLogic agent.
- Before you configure the WebLogic agent, the Oracle WebLogic server first must be configured by completing the following tasks:

  Note: Most of the Oracle WebLogic server configuration is done by using the administrative console, typically at `http://`*`weblogic-server`*`:7001/console`.

  1. Set up a monitor user in the Monitors group.
     a. Select the domain to monitor/edit.
     b. Select **Security Realms**.
     c. Select your security realm (or create one if one does not exist).
     d. Create a user that will be used to communicate with WebLogic over JMX.
     e. Add this user the Monitors group.
     f. Save the changes to the domain.
  2. Enable the Listen Ports.
     a. Select the domain to monitor/edit.
     b. On each server that you want to monitor, click **Environment** > **Servers** > ***Select a server*** .
     c. Ensure that the **Listen Port** is enabled and note its port number.
     d. If you want to enable SSL, then ensure that the **SSL Listener Port** is enabled and set a port for SSL as well.
  3. Enable the JMX MBean Server Connections.
     a. Select the domain that you want to monitor/edit.
     b. Select **Configure** > **Advanced**.
     c. Check **Platform Mbean Server Enabled**.
     d. Save the change.
  4. Enable the IIOP Protocol option.
     a. Select the domain that you want to monitor/edit.
     b. On each server that you would like to monitor, click **Environment** > **Servers** then select a server.
     c. Select the **Protocol Tab** > ***Select IIOP***.
     d. Under the **Advanced** section, enter the default IIOP user name and password.
     e. Save the change.
  5. Enable SSL.

a. Enable HTTP Tunneling.

   1) Go to **Environment** > **Servers** > ***Select a server*** > **Protocol** > **General**.

   2) Check **Enable HTTP Tunneling**.

b. Enable SSL Listen Port.

   1) Go to **Environment** > **Servers** > ***Select a server*** > **Configuration** > **General**.

   2) Configure a port number.

About this task

The WebLogic agent is both a multiple instance agent and also a multiple subnode agent. You can create one agent instance with multiple subnodes – one for each WebLogic server, or you can create an agent instance for each WebLogic server with one subnode for that server. Or you can create a combination of each type of configuration. After you configure agent instances, you must start each agent instance manually.

Procedure

1. To configure the agent on Windows systems, use the **IBM Performance Management** window or the silent response file with the agent configuration batch file.
   - "Configuring the agent on Windows systems" on page 815.
   - "Configuring the agent by using the silent response file" on page 819.
2. To configure the agent on Linux and UNIX systems, run the agent configuration script and respond to prompts, or use the silent response file.
   - "Configuring the agent by responding to prompts" on page 819.
   - "Configuring the agent by using the silent response file" on page 819.
3. Optional: To configure transaction tracking, configure individual agent instances to provide transaction tracking data and configure your Application Performance Dashboard to display transaction tracking data.

   a) Follow the procedure for "Configuring transaction tracking for the WebLogic agent" on page 822.

   b) Follow the procedure for "Configuring your Application Performance Dashboard to display transaction tracking data for the WebLogic agent" on page 827.

   Note: Transaction tracking capability is available for the WebLogic agent in the Cloud APM, Advanced offering. For the WebLogic agent with basic resource monitoring capability, which is in the Cloud APM, Base offering, skip this step.

What to do next

In the Cloud APM console, go to your Application Performance Dashboard to view the data that was collected. For more information about using the Cloud APM console, see "Starting the Cloud APM console" on page 957.

If you are unable to view the data in the agent dashboards, first check the server connection logs and then the data provider logs. The default paths to these logs are as follows:

- Linux AIX `/opt/ibm/apm/agent/logs`
- Windows `C:\IBM\APM\TMAITM6_x64\logs`

For help with troubleshooting, see the Cloud Application Performance Management Forum.

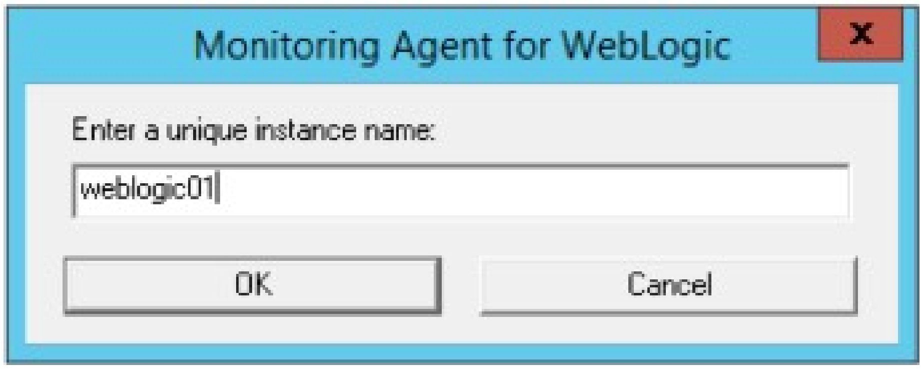
Configuring the agent on Windows systems

You can configure the WebLogic agent on Windows operating systems by using the IBM Cloud Application Performance Management window. After you update the configuration values, you must start the agent to save the updated values.

Procedure

1. Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Cloud Application Performance Management**.
2. In the **IBM Performance Management** window, right-click the **Monitoring Agent for WebLogic** template, and then click **Configure agent**.

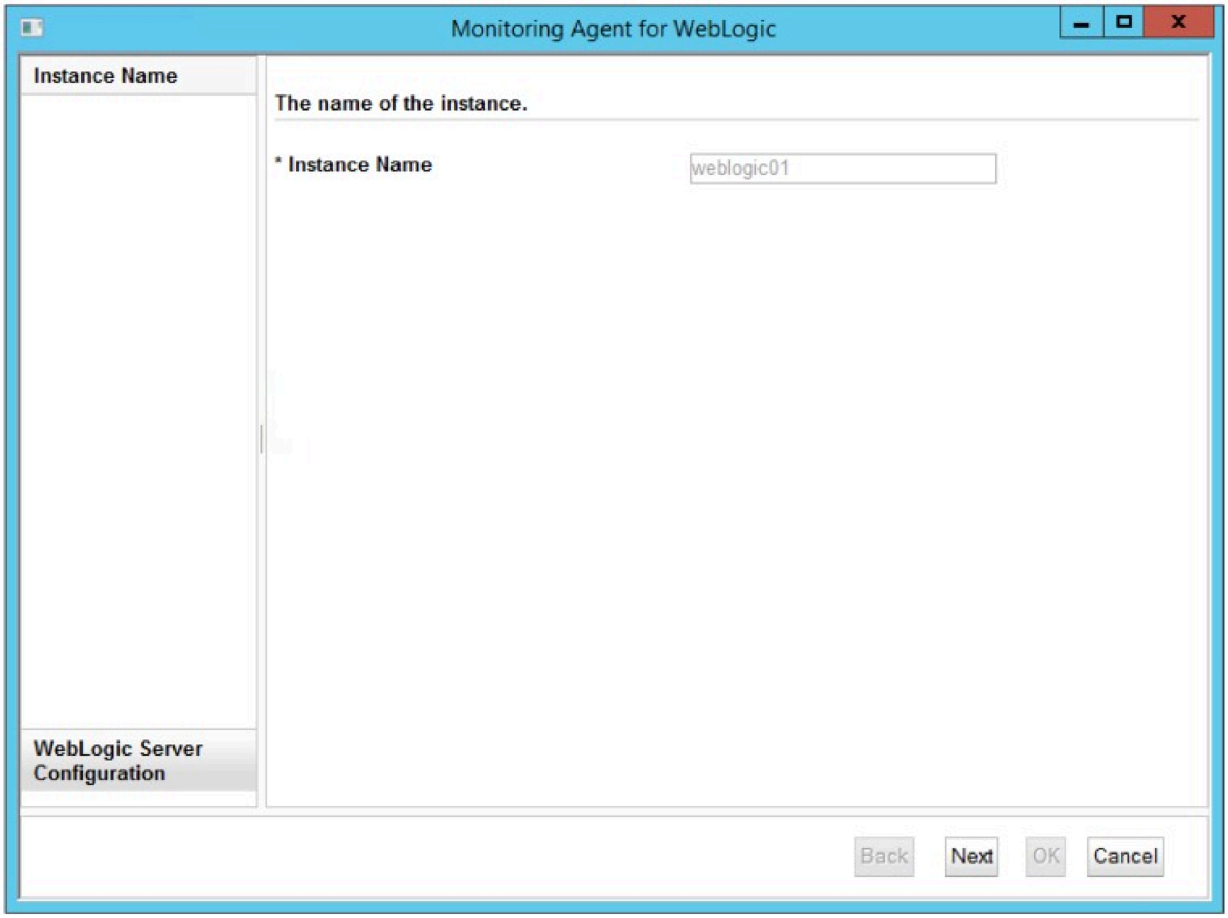
   Remember: After you configure an agent instance for the first time, the **Configure agent** option is disabled. To configure the agent instance again, right-click on it and then click **Reconfigure**.
3. Enter a unique instance name then click **OK**. Use only letters, Arabic numerals, the underline character, and the minus character in the instance name. For example: `weblogic01`.

   Monitoring Agent for WebLogic

   Enter a unique instance name: weblogic01

   Figure 27. The window to enter a unique instance name
4. Click **Next** on the **Instance Name** agent configuration panel.



The name of the instance.

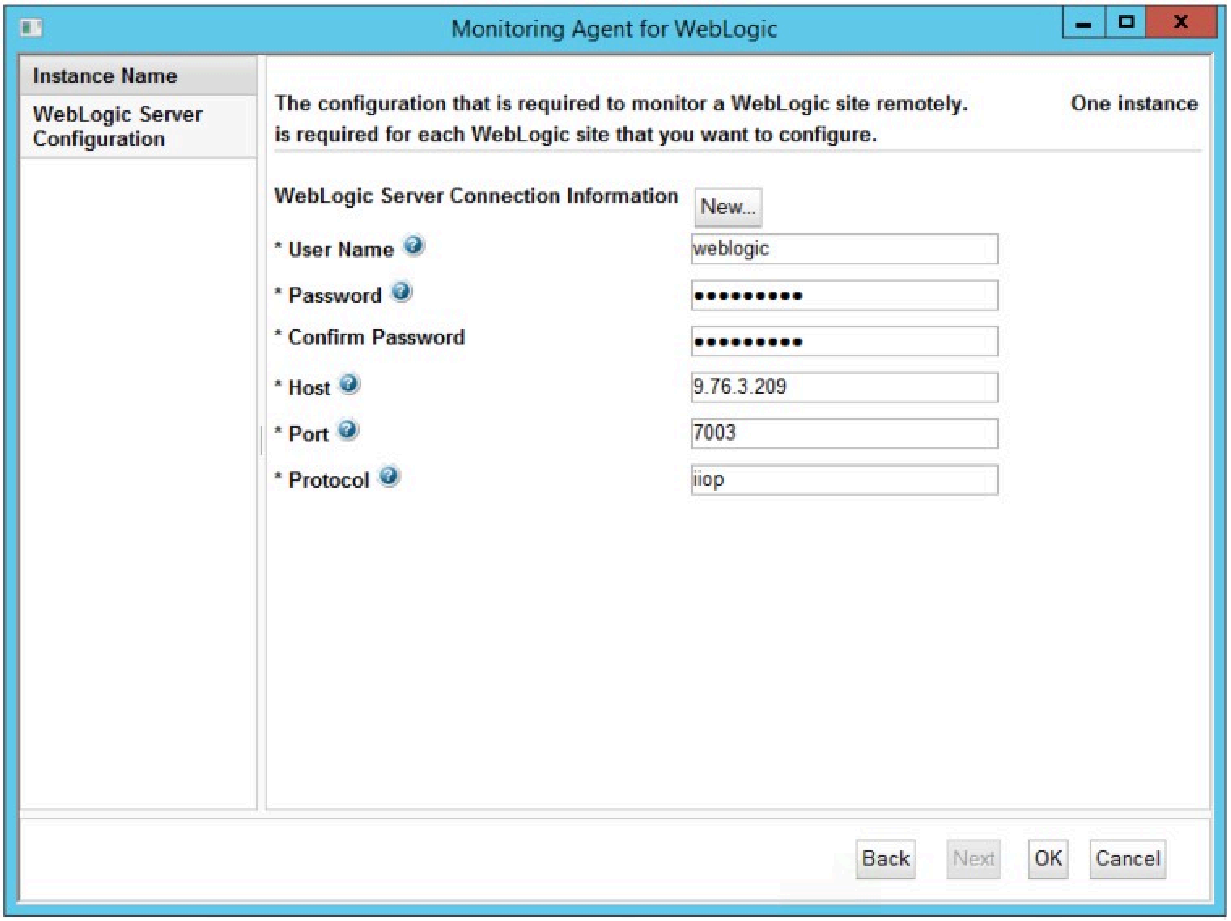
* Instance Name: weblogic01

Figure 28. The window displaying the agent instance name

5. Enter the **WebLogic Server Configuration** instance template settings.

   Note: This section is not the WebLogic server connection instance configuration. It is a template section for setting what is used as the default values when you add the actual WebLogic server connection instance configurations beginning in step 6.

   See Table 226 on page 821 for an explanation of each of the configuration parameters.



The configuration that is required to monitor a WebLogic site remotely. One instance is required for each WebLogic site that you want to configure.

WebLogic Server Connection Information

* User Name: weblogic

* Password: ●●●●●●●●●●

* Confirm Password: ●●●●●●●●●●

* Host: 9.76.3.209

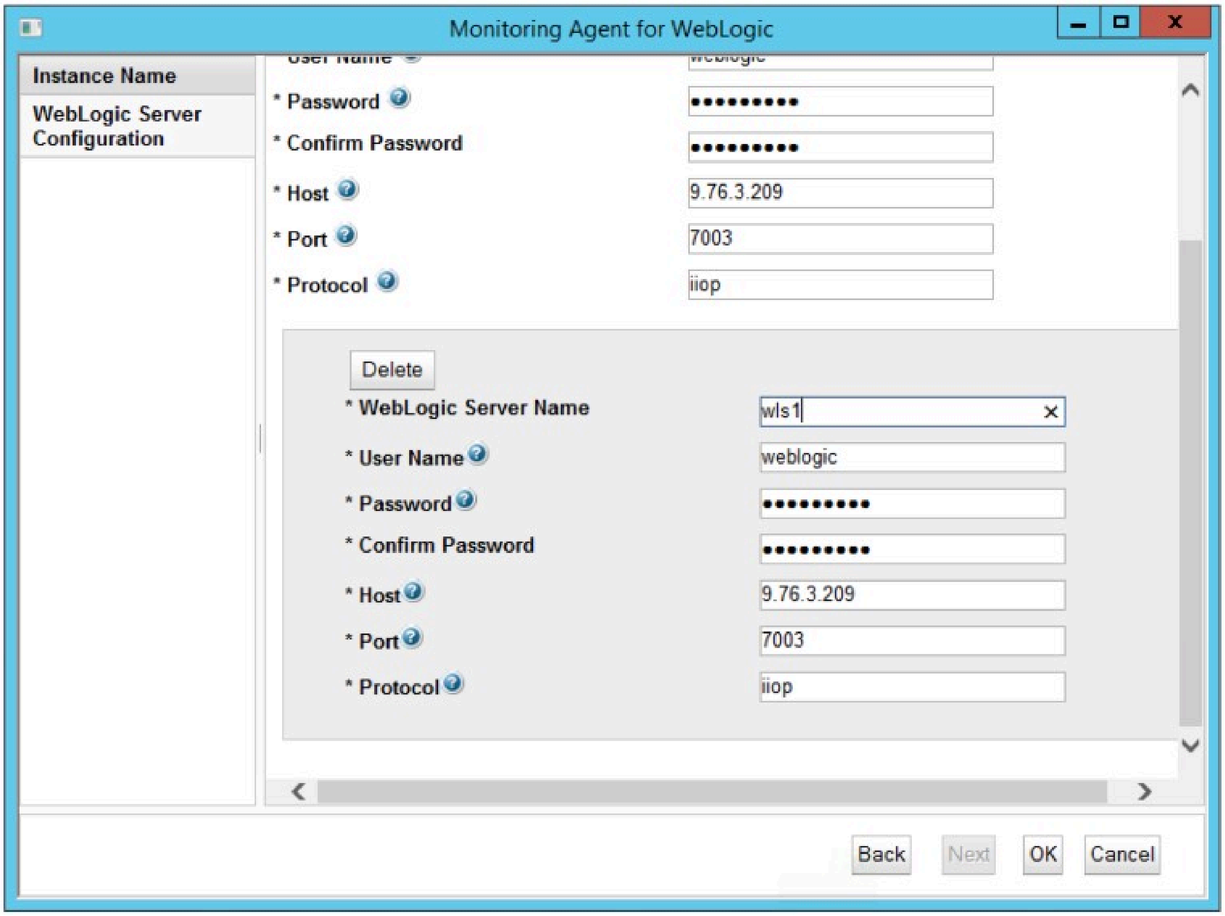
* Port: 7003

* Protocol: iiop

Figure 29. The window to specify WebLogic server connection instance template settings

6. Press **New** and enter WebLogic server connection instance settings, then click **Next**.

   See Table 226 on page 821 for an explanation of each of the configuration parameters.



* User Name: weblogic
* Password: •••••••••
* Confirm Password: •••••••••
* Host: 9.76.3.209
* Port: 7003
* Protocol: iiop

* WebLogic Server Name: wls1
* User Name: weblogic
* Password: •••••••••
* Confirm Password: •••••••••
* Host: 9.76.3.209
* Port: 7003
* Protocol: iiop

Figure 30. The window to specify WebLogic server connection instance settings

7. Click **OK** to complete the configuration.
8. Copy the WebLogic security files into the WebLogic agent binary directory.

   a. Locate the `wlclient.jar` and `wljmxclient.jar` files under `ORACLE_HOME`. For example, `C:\Oracle\Middleware\Oracle_Home\wlserver\server\lib`.

   b. Copy the files from step “8.a” on page 818 to the WebLogic agent binary directory.

   - Linux AIX *`install_dir`*`/bin`.
   - Windows *`install_dir`*`\TMAITM6_x64`

   where *install_dir* is the path where the agent is installed. The default *install_dir* paths are listed here:

   - Linux AIX `/opt/ibm/apm/agent`
   - Windows `C:\IBM\APM\TMAITM6_x64`

9. In the IBM Cloud Application Performance Management window, right-click the instance that you configured, and then click **Start**.

Configuring the agent by responding to prompts

After installation of the WebLogic agent, you must configure it before you start the agent. If the WebLogic agent is installed on a local Linux or UNIX computer, you can follow these instructions to configure it interactively through command line prompts.

About this task

Remember: If you are reconfiguring a configured agent instance, the value that is set in the last configuration is displayed for each setting. If you want to clear an existing value, press the space key when the setting is displayed.

Procedure

Follow these steps to configure the WebLogic agent by running a script and responding to prompts.

1. Run the following command.

   ```
   install_dir/bin/weblogic-agent.sh config instance_name
   ```

   where *install_dir* is the path where the agent is installed and *instance_name* is the name that you want to give to the agent instance.

   Example

   ```
   /opt/ibm/apm/agent/bin/weblogic-agent.sh config example-inst01
   ```

2. Respond to the prompts to set configuration values for the agent.

   See “Configuration parameters for the WebLogic agent” on page 821 for an explanation of each of the configuration parameters.

3. Copy the WebLogic client library files into the WebLogic agent binary directory.

   a) Locate the `wlclient.jar` and `wljmxclient.jar` files under `ORACLE_HOME`.

   b) Copy the files from step “3.a” on page 819 to the WebLogic agent binary directory.

      ```
      install_dir/bin
      ```

   where *install_dir* is the path where the agent is installed.

   Example

   ```
   /opt/ibm/apm/agent/bin
   ```

4. Run the following command to start the agent:

   ```
   install_dir/bin/weblogic-agent.sh start instance_name
   ```

   where *install_dir* is the path where the agent is installed and *instance_name* is the name of the agent instance.

   Example

   ```
   /opt/ibm/apm/agent/bin/weblogic-agent.sh start example-inst01
   ```

Configuring the agent by using the silent response file

The silent response file contains the agent configuration parameters. You can edit the silent response file to modify the values of these parameters, and run the configuration script to create an instance and update the agent configuration values. This mode of configuration is called the silent mode.

About this task

The silent response file contains the agent configuration parameters with default values defined for some parameters. You can edit the silent response file to specify different values for the configuration parameters.

After you update the configuration values in the silent response file, you must run the configuration script to configure the agent with these updated values.

Procedure

Configure the WebLogic agent in the silent mode by completing the following steps.

1. In a text editor, open the `weblogic_silent_config.txt` file that is available at the following path:

   - Linux AIX *install_dir*`/samples/weblogic_silent_config.txt`
   - Windows *install_dir*`\samples\weblogic_silent_config.txt`

   where *install_dir* is the path where the agent is installed.

   Example

   - Linux AIX `/opt/ibm/apm/agent/samples/weblogic_silent_config.txt`
   - Windows `C:\IBM\APM\samples\weblogic_silent_config.txt`

2. In the `weblogic_silent_config.txt` file, specify values for all mandatory parameters. You can also modify the default values of other parameters.

   See "Configuration parameters for the WebLogic agent" on page 821 for an explanation of each of the configuration parameters.

3. Save and close the `weblogic_silent_config.txt` file, and run the following command:

   - Linux AIX *install_dir*`/bin/weblogic-agent.sh config` **instance_name** *install_dir*`/samples/weblogic_silent_config.txt`
   - Windows *install_dir*`\bin\weblogic-agent.bat config` **instance_name** *install_dir*`\samples\weblogic_silent_config.txt`

   where *install_dir* is the path where the agent is installed and *instance_name* is the name that you want to give to the agent instance.

   The default *install_dir* paths are listed here:

   - Linux AIX `/opt/ibm/apm/agent`
   - Windows `C:\IBM\APM\TMAITM6_x64`

   Important: Ensure that you include the absolute path to the silent response file. Otherwise, the agent data is not shown in the dashboards.

   Example

   - Linux AIX `/opt/ibm/apm/agent/bin/weblogic-agent.sh config example-inst01 /opt/ibm/apm/agent/samples/weblogic_silent_config.txt`
   - Windows `C:\IBM\APM\bin\weblogic-agent.bat config example-inst01 C:\IBM\APM\samples\weblogic_silent_config.txt`

4. Copy the WebLogic client libraries into the WebLogic agent binary directory.

   a. Locate the `wlclient.jar` and `wljmxclient.jar` files under `ORACLE_HOME`.

   b. Copy the files from step "5.a" on page 820 to the WebLogic agent binary directory.

      - Linux AIX *install_dir*`/bin`.
      - Windows *install_dir*`\TMAITM6_x64`

   where *install_dir* is the path where the agent is installed. The default *install_dir* paths are listed here:

   - Linux AIX `/opt/ibm/apm/agent`
   - Windows `C:\IBM\APM\TMAITM6_x64`

5. Run the following command to start the agent:

- Linux AIX *install_dir*/bin/weblogic-agent.sh start **instance_name**
- Windows *install_dir*\bin\weblogic-agent.bat start **instance_name**

where *install_dir* is the path where the agent is installed and *instance_name* is the name of the agent instance.

The default *install_dir* paths are listed here:

- Linux AIX `/opt/ibm/apm/agent`
- Windows `C:\IBM\APM\TMAITM6_x64`

Example

- Linux AIX `/opt/ibm/apm/agent/bin/weblogic-agent.sh start example-inst01`
- Windows `C:\IBM\APM\bin\weblogic-agent.bat start example-inst01`

Configuration parameters for the WebLogic agent

The configuration parameters for the WebLogic agent are displayed in a table.

1. WebLogic Agent Settings - WebLogic agent environment settings.

Table 226. WebLogic Agent Settings

Parameter name	Description	Silent configuration file parameter name
WebLogic Server Name	Provide a name to identify the WebLogic server agent instance. Example: *wls1* **Note:** This alias can be anything you choose to represent the WebLogic server agent instance with the following restrictions. Only letters, Arabic numerals, the underline character, and the minus character can be used in the connection name. The maximum length of a connection name is 25 characters.	Each of the following parameters must have an instance name suffix which will be the same for each parameter of an agent instance. New agent instances must use a unique instance name for its set of parameters. For example, one agent instance can use *.wls1* and another agent instance can use *.wls*2 in place of *.instance_name* in the parameter names below.
User Name	Username that is used to authenticate with the WebLogic server.	**KWB_WLS_USERNAME.instance_name**
Password	Password that is used to authenticate with the WebLogic server.	**KWB_WLS_PASSWORD.instance_name**
Host	Host that is used to authenticate with the WebLogic server. Type either the fully qualified host name or the IP address of the WebLogic server.	**KWB_WLS_HOST.instance_name**
Port	Port that is used to authenticate with the WebLogic server.	**KWB_WLS_PORT.instance_name**
Protocol	Protocol that is used to authenticate with the WebLogic server. Supported protocols are *iiop* and *https*.	**KWB_WLS_PROTOCOL.instance_name**

Configuring transaction tracking for the WebLogic agent

The transaction tracking capability of the WebLogic agent requires changes to the agent instance environment settings file and the WebLogic server startup file. A script is provided to help you make the changes.

Before you begin

Linux AIX Ensure that the resource limit for open files is greater than 5,000 for the transaction tracking toolkit to work properly.

- Display the current open file limit setting. **`ulimit -n`**
- Example setting the open file limit to 5,056. **`ulimit -n 5056`**

Perform "Configuring WebLogic monitoring" on page 813 Windows step 1 or Linux AIX step 2 before you follow this procedure.

Note: Transaction tracking capability is available for the WebLogic agent in the Cloud APM, Advanced offering. For the WebLogic agent with basic resource monitoring capability, which is in the Cloud APM, Base offering, skip this step.

The WebLogic agent must be installed locally to the WebLogic server that is monitored with the transaction tracking capability.

The user account that runs this script must have write permission to the following directories and files:

1. The *`WEBLOGIC_HOME`* directory.
2. The *`WEBLOGIC_HOME`*`/bin` directory and files.
3. The *`install_dir`*`/config` directory.
4. The *`install_dir`*`/config/`*`hostname`*`_wb_`*`instance_name`*`.cfg` file.

where

WEBLOGIC_HOME
WebLogic server installation directory.

install_dir
Path where the agent is installed. The default paths to these logs are as follows.

- Linux AIX `/opt/ibm/apm/agent`
- Windows `C:\IBM\APM\TMAITM6_x64`

hostname
Name of the host computer where the agent is installed.

instance_name
Name of the agent instance that is assigned in the agent configuration method topic:

- Configuring the agent on Windows systems, step "3" on page 815
- Configuring the agent by responding to prompts, step "1" on page 819
- Configuring the agent by using the silent response file, step "2" on page 820

Procedure

Run the **`simpleConfig`** script.

1. Log in to the WebLogic server with the WebLogic agent installed.
2. Change directory to the agent installation directory.
   - Linux AIX *`install_dir`*
   - Windows *`install_dir`*`\TMAITM6_x64`

   where *install_dir* is the path where the agent is installed.

The default *install_dir* paths are listed here:

- Linux AIX `/opt/ibm/apm/agent`
- Windows `C:\IBM\APM\TMAITM6_x64`

3. Change directory to `wbdchome/8.1.4.0.0/bin`.
4. Run the setup script.
   - Linux AIX `./simpleConfig.sh`
   - Windows `simpleConfig.bat`
5. Follow the prompts to enter parameters for your environment:
   a) Choose the WebLogic agent *instance_name* and subnode to configure from the list of detected agent instance and subnode combinations where *instance_name* is the name of the agent instance.

   b) Type the number of the WebLogic server startup method.

   c) Type the WebLogic domain root search path.

   This path is used as the base from which to search for WebLogic domains. If the *WEBLOGIC_HOME* environment variable is set, its value is offered as the default value.

   d) Type the number of the WebLogic domain for the WebLogic server to configure.

   e) Type the number of the WebLogic server name to configure.

   Linux Example configuration with a WebLogic startup method of `WebLogic startup script`.

```
./simpleconfig.sh

The following agents and subnodes are not yet configured for transaction tracking:

1) wlinst1 Server1
2) wlinst1 Server2

Type the number that corresponds to the agent instance and subnode that you want to
configure.

Type your selection here (For example: 1): 1

The following WebLogic startup methods are supported:

1) WebLogic startup script
2) WebLogic Node Manager

Type your selection here (default is 1): 1

The path to begin looking for WebLogic domains.
WebLogic domain search root (default is: ): /home/wlsadmin

The found WebLogic domain paths are:

1) /home/wlsadmin/oracle/user_projects/domains/ttdd

Type the number that corresponds to the WebLogic domain containing the WebLogic server that
you want to configure.

Type your selection here (For examble: 1): 1

The following WebLogic servers are available for configuration:

1) AdminServer
2) Server1

Select a WebLogic server name (default is: 2): 2
INFO: [2000] Automatic configuration of agent environment file succeeded.
INFO: [3000] Automatic configuration of WebLogic start script succeeded.
INFO: [9000] Restart the WebLogic agent and WebLogic server for configuration to take effect.
```

6. Follow these steps if `WebLogic Node Manager` is selected as the **`WebLogic server startup method`** in step “5.b” on page 823. Otherwise, proceed to step “7” on page 825.
   a) Open the `weblogic_nodemanager_dc_opts` file that is listed in information message number 3011 of the output text from step “5” on page 823.

Windows Example configuration output with a WebLogic startup method of `WebLogic Node Manager`.

```
INFO: [2000] Automatic configuration of agent environment file succeeded.
INFO: [3010] Automatic configuration of WebLogic start script skipped.
INFO: [3011] Please review C:\IBM\APM\TMAITM6_x64\wbdchome\8.1.4.0.0\runtime\ttdd_win
\win_Server1\
staging\weblogic_nodemanager_dc_opts.win for required WebLogic JVM start options.
INFO: [9000] Restart the WebLogic agent and WebLogic server for configuration to take
effect.
```

b) Log in to the WebLogic console and select **Environment** > **Servers**.

c) Select the server to configure.

d) Select the **Configuration** > **Server Start** tab.

e) Copy the server start arguments from the `weblogic_nodemanager_dc_opts` file to the server's **Start Server** Arguments in the WebLogic console and save the changes.

The server start arguments are all the lines that follow the comment line `# Add the following lines to the server start arguments` in the `weblogic_nodemanager_dc_opts` file.

f) Ensure that the transaction tracking toolkit is in the shared library path at run time.

Choose a method.

- Update the Node Manager start script.

  Note: All WebLogic servers started by the Node Manager have this library path set with the transaction tracking toolkit object file libraries included.

  1) Open the `weblogic_nodemanager_dc_opts` file that is listed in information message number 3011 of the output text from step "5" on page 823.

  2) Set the transaction tracking toolkit path in the Node Manager start script. The command to set the path is the line that follows the comment line `# Make sure that the executable path available to the WebLogic server includes the toolkit lib directory` in the `weblogic_nodemanager_dc_opts` file.

     – Linux Copy the LD_LIBRARY_PATH line from the generated `weblogic_nodemanager_dc_opts.linux` file and paste it below the `export` JAVA_OPTIONS line in the Node Manager start script. Example, `*WEBLOGIC_HOME*/user_projects/domains/*domain_name*/bin/startNodeManager.sh`.

     – Windows Copy the PATH from the generated `weblogic_nodemanager_dc_opts.win` file and paste it below the `export JAVA_OPTIONS` line in the Node Manager start script. Example, `*WEBLOGIC_HOME*\user_projects\domains\*domain_name*\bin\startNodeManager.bat`.

     where *WEBLOGIC_HOME* is the WebLogic server installation directory and *domain_name* is the name of the WebLogic domain.

- Update the environment for the user account that starts the Node Manager.

  Note: All applications started by the user account have this library path set with the Toolkit object file libraries included.

  1) Edit environment settings for the user that starts the Node Manager.

     – Linux AIX Edit the shell resource file or shell profile file. For example, in the bash shell, `.bashrc` or `.bash_profile`.

     – Windows Edit **Control Panel** > **System and Security** > **System** > **Advanced system settings** > **Environment Variables...** > **User variables for *user_name*** > **Path** where *user_name* is the name of the user account that is used to start the WebLogic server.

  2) Set the transaction tracking toolkit path in the user account environment. The command to set the path is the line that follows the comment line `# Make sure that the executable`

path available to the WebLogic server includes the toolkit lib directory in the weblogic_nodemanager_dc_opts file.

  - Linux AIX Copy the `export LD_LIBRARY_PATH` line from the generated `weblogic_nodemanager_dc_opts.linux` file. If an `export LD_LIBRARY_PATH` line does not exist, add it. If it does exist, edit it to add only the path from the right of the equals sign to the existing path with the correct path delimiter.
  - Windows Copy the `set PATH` line from the generated `weblogic_nodemanager_dc_opts.win` file. If a **Path** variable does not exist in the **User variables for *user_name*** section where *user_name* is the name of the user account that is used to start the WebLogic server, add it by entering `Path` as the variable name and the path from the right of the equals sign as the value. If it does exist, edit the value to add only the path from the right of the equals sign to the existing path with the correct path delimiter.

3) Reload the environment.

Warning: The `startNodeManager` scripts are generated by the WebLogic configuration utility. So you might lose your changes when the WebLogic configuration is run again.

7. If the WebLogic server and the agent are running, restart them.

Results

WebLogic server files that are changed during transaction tracking configuration:

- The `startManagedWebLogic` script.
  - Linux AIX `*WEBLOGIC_HOME*/bin/startManagedWebLogic.sh`
  - Windows `*WEBLOGIC_HOME*\bin\startManagedWebLogic.cmd`

  where *WEBLOGIC_HOME* is the WebLogic server installation directory.

  This file is updated with the configuration settings for the transaction tracking capability. Configuration markers are inserted into the file for use when you disable the transaction tracking capability. A backup file is saved in the `*WEBLOGIC_HOME*/bin/bak/` directory before the script adds or removes the transaction tracking capability changes.

Agent files that are changed during transaction tracking configuration:

- Agent instance configuration file
  - Linux AIX `*install_dir*/config/*hostname*_wb_*instance_name*.cfg`
  - Windows `*install_dir*\TMAITM6_x64\*hostname*_WB_*instance_name*.cfg`
- Agent environment settings file
  - Linux AIX `*install_dir*/config/wb_*instance_name*.environment`
  - Windows `*install_dir*\TMAITM6_x64\KWBENV_*instance_name*`

where

install_dir
Path where the agent is installed. The default paths to these logs are as follows.

- Linux AIX `/opt/ibm/apm/agent`
- Windows `C:\IBM\APM\TMAITM6_x64`

hostname
Name of the host computer where the agent is installed.

instance_name
Name of the agent instance that is assigned in the agent configuration method topic:

- Configuring the agent on Windows systems, step “3” on page 815
- Configuring the agent by responding to prompts, step “1” on page 819
- Configuring the agent by using the silent response file, step “2” on page 820

Disabling transaction tracking for a WebLogic agent instance

The transaction tracking capability of the WebLogic agent can be removed. A script is provided to remove the transaction tracking capability for an agent instance.

Before you begin

Ensure that the WebLogic server and the WebLogic agent are shut down.

The user account that runs this script must have write permission to the following directories and files:

1. The *WEBLOGIC_HOME* directory.
2. The *WEBLOGIC_HOME*/bin directory and files.
3. The *install_dir*/config directory.
4. The *install_dir*/config/*hostname*_wb_*instance_name*.cfg file.

Procedure

Run the **unconfig** script with the **remove** option.

1. Log in to the WebLogic server with the WebLogic agent installed.
2. Change directory to the agent installation directory.
   - Linux AIX *install_dir*
   - Windows *install_dir*\TMAITM6_x64
3. Change directory to wbdchome/8.1.4.0.0/bin.
4. Run the **unconfig** script with the **remove** option and the agent instance name the subnode name.
   - To disable one subnode for an agent instance, use the *subnode_name* parameter.
     - Linux AIX ./unconfig.sh **remove** *instance_name subnode_name*
     - Windows unconfig.bat **remove** *instance_name subnode_name*
   - To disable all subnodes for an agent instance, omit the *subnode_name* parameter.
     - Linux AIX ./unconfig.sh **remove** *instance_name*
     - Windows unconfig.bat **remove** *instance_name*
5. Start the WebLogic server and the agent.

Uninstalling transaction tracking for the WebLogic agent

The transaction tracking capability of the WebLogic agent can be uninstalled. A script is provided to remove the transaction tracking capability from all agent instances and also to remove the transaction tracking toolkit.

Before you begin

Ensure that the WebLogic server and all WebLogic agent instances are shut down.

The user account that runs this script must have write permission to the following directories and files:

1. The *WEBLOGIC_HOME* directory.
2. The *WEBLOGIC_HOME*/bin directory and files.
3. The *install_dir*/config directory.
4. The *install_dir*/config/*hostname*_wb_*instance_name*.cfg file.

Procedure

Run the **unconfig** script with the **uninstall** option.

1. Log in to the WebLogic server with the WebLogic agent installed.
2. Change directory to the agent installation directory.
   - Linux AIX *install_dir*
   - Windows *install_dir*\TMAITM6_x64
3. Change directory to `wbdchome/8.1.4.0.0/bin`.
4. Run the **unconfig** script with the **uninstall** option.
   - Linux AIX `./unconfig.sh` **`uninstall`**
   - Windows `unconfig.bat` **`uninstall`**
5. Start the WebLogic server and all agent instances.

Configuring your Application Performance Dashboard to display transaction tracking data for the WebLogic agent

Viewing data that is gathered by the transaction tracking capability of the WebLogic agent requires configuration changes to your Application Performance Dashboard.

Before you begin

Perform "Configuring transaction tracking for the WebLogic agent" on page 822 before you follow this procedure.

Procedure

1. Enable the transaction tracking data in the Application Performance Dashboard if you have the WebLogic agent with transaction tracking capability, which is in the Cloud APM, Advanced offering, and you want to enable the transaction tracking capability.
   a) From the navigation bar, click **System Configuration** > **Agent Configuration**.

      The **Agent Configuration** page is displayed.

   b) Select the **WebLogic** tab.

   c) Select the check boxes for the WebLogic server agent instances that you want to monitor and take one of the following actions from the **Actions** list:
      - To enable transaction tracking, click **Set Transaction Tracking** > **Enabled**. The status in the **Transaction Tracking** column is updated to *Enabled*.
      - To disable transaction tracking, click **Set Transaction Tracking** > **Disabled**. The status in the **Transaction Tracking** column is updated to *Disabled*.
2. To view the WebLogic agent transaction tracking data dashboards, add the WebLogic agent instance to an application in your Application Performance Dashboard.

   For more information about the Applications editor, see Managing applications.
3. Ensure that user accounts are assigned to a role that includes the Diagnostic Dashboard permission to have access to the following WebLogic agent transaction tracking Application Dashboard buttons.

   Otherwise, these buttons are disabled for that user in the Application Dashboard.
   a. The **Diagnose** drill-down button on the **Slowest 5 Response Time** widget.
   b. The **Inflight Requests** button on the **Applications** widget.

Configuring WebSphere Applications monitoring

Configuring WebSphere Applications monitoring involves configuring a data collector for your application servers. The data collector can be either stand-alone or embedded with the WebSphere Applications agent.

Embedded data collector

Most of the WebSphere application servers can be monitored by the embedded data collector, except for the Liberty profile on IBM Cloud. The embedded data collector can provide all available monitoring features.

To configure the embedded data collector, you must first install the WebSphere Applications agent on the system where the application server is running. After that, use the provided configuration utilities to configure the data collector interactively or silently.

Stand-alone data collector

The stand-alone data collector is applicable only for WebSphere Application Server Liberty on Linux for System x and for WebSphere Liberty profile on IBM Cloud.

If you choose to configure a stand-alone data collector, you can skip the agent installation procedure and directly configure the data collector in Liberty.

However, some on-demand diagnostics data will not be collected by the stand-alone data collector, such as heap dump at the current time or in-flight request information. It means you can only enable the data collector to automatically collect heap dump information at specified intervals but you cannot take heap snapshot whenever you want by using the **Take Snapshot** button from the Cloud APM console. All in-flight request related dashboards, which can be provided by the embedded data collector, are not available to the stand-alone data collector.

Use Table 227 on page 828 to determine the appropriate data collector for your application server.

Table 227. WebSphere applications and applicable data collectors

Application to be monitored	Applicable data collector	Documentation
WebSphere Application Server traditional	Embedded data collector	"Configuring the data collector for WebSphere Applications agent" on page 828
WebSphere Liberty profile on IBM Cloud	Stand-alone data collector	"Configuring the Liberty data collector in IBM Cloud environment (Liberty V18.* and older versions)" on page 468
WebSphere Liberty profile in Docker container	Embedded data collector	"Monitoring WebSphere Application Server Liberty inside a Docker container" on page 860

Configuring the data collector for WebSphere Applications agent

The WebSphere Applications agent does not need any configuration after agent installation, unless you want to change the default port. However, you must configure the data collector, which is a component of the agent, to set up monitoring for your WebSphere environment.

About this task

The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see "Change history" on page 54.

Procedure

- (Fast track) If you only use the WebSphere Applications agent without the legacy product ITCAM Agent for WebSphere Applications in your environment, to quickly set up the environment for monitoring, see "Fast track: Configuring the data collector for WebSphere Applications agent" on page 829 for a simplified configuration flow.

- (Simple configuration) For a complete configuration flow for a pure IBM Cloud Application Performance Management environment, see "Configuring the data collector with the simple configuration utility" on page 832.
- (Full configuration) To configure the data collector with more customization options, use the full configuration utilities. For instructions, see "Configuring or reconfiguring the data collector with full configuration utilities" on page 834.
- (Silent configuration) To deploy the same monitoring for many application server instances, configure the data collector in silent mode. For instructions, see "Configuring the data collector in silent mode" on page 842.
- (WebSphere Portal Server) To monitor WebSphere Portal Server instances, use the advanced configuration procedure. For instructions, see "Configuring or reconfiguring the data collector with full configuration utilities" on page 834.
- (Manual configuration) If you cannot use the provided configuration utilities to configure the data collector for WebSphere Applications agent, manually configure the data collector in the WebSphere Administrative Console. For instructions, see "Manually configure the data collector if the configuration utilities fail" on page 849.
- (Agent coexistence) If you want to configure the data collector to work in an agent coexistence environment where both WebSphere Applications agent and ITCAM Agent for WebSphere Applications are installed, see "(Agent coexistence) Configuring the WebSphere Applications agent and the data collector" on page 853.
- (Docker monitoring) To monitor WebSphere Application Server Liberty running inside a Docker container, see "Monitoring WebSphere Application Server Liberty inside a Docker container" on page 860.

Fast track: Configuring the data collector for WebSphere Applications agent

The WebSphere Applications agent does not need any configuration after agent installation. However, you must configure the data collector, which is a component of the agent, to set up monitoring for your WebSphere environment.

Before you begin

1. Install the WebSphere Applications agent on the system where the application server to be monitored is installed and running.
2. Check the user access requirements.
   - Windows Use the administrator ID that is used to install the application server to configure the data collector. Make sure that this user ID has full write permission the data collector home directory, *install_dir*`\dchome\7.3.0.14.08`.
   - Linux AIX Use the user ID that is used to install the application server to configure the data collector. Make sure that this user ID has read and write permissions to the following subdirectories within *install_dir*`/yndchome/7.3.0.14.08`:
     - `bin`
     - `data`
     - `runtime`

About this task

A simple configuration utility, `simpleconfig`, is used in this procedure to provide the basic configuration of data collector.

The `simpleconfig` utility configures the data collector with default settings. To configure the data collector with more customization options, use the full configuration utility, `config`, in the same directory. For instructions, see "Configuring or reconfiguring the data collector with full configuration utilities" on page 834.

In most cases, the `simpleconfig` utility is sufficient. For more complex environment, you can use the `config` configuration utility to configure the data collector. If the `simpleconfig` utility fails, use the `config` utility instead.

Procedure

1. Log in to the system with the user ID that is used to install the application server.
2. Change to the `bin` directory within the data collector home directory.
   - Windows *install_dir*\dchome\7.3.0.14.08\bin
   - Linux AIX *install_dir*/yndchome/7.3.0.14.08/bin
3. Run the following simple configuration utility:
   - Windows **simpleconfig.bat**
   - Linux AIX **./simpleconfig.sh**
4. Follow the prompts to continue with the data collector configuration.

   You are required to do some or all of the following things, depending on the application server settings:

   - For traditional WebSphere Application Server:
     - Select the auto-discovered WebSphere installation directory or manually specify the installation directory.
     - Select the WebSphere Application Server profile to monitor.
     - Select the security properties profile to use or provide the user name and password of the WebSphere administrative console (if security is enabled for the application server).
   - For WebSphere Application Server Liberty:
     - Specify the full path of the Liberty home directory that contains `bin` and `servers` directories. For example, `/opt/ibm/wlp`.
     - Specify the home directory of the JRE that is used by Liberty.
5. After the data collector configuration completes, restart the application server.

   a) Go to the `bin` directory under the home directory for the application server profile. For example, opt/IBM/WebSphere/AppServer/profiles/*profile_name*/bin.

   b) Stop the application server by entering the **stopServer** command in the command console.
      - Linux AIX ./stopServer.sh *server_name*
      - Windows stopServer.bat *server_name*

   c) When prompted, enter the user ID and password of WebSphere administrative console administrator.

   d) Start the application server again by entering the **startServer** command in the command console.
      - Linux AIX ./startServer.sh *server_name*
      - Windows startServer.bat *server_name*
6. Log in to the Cloud APM console to view data in the dashboards.

   a) Access the console by using the link that is provided in the email alerting you that your service is ready. Alternatively, access the console from the IBM Marketplace website. For detailed instructions, see "Starting the Cloud APM console" on page 957.

b) Use the Applications editor to add the monitored application server to the Application Performance Dashboard. You can add it as a new component to your existing application or create an application to contain this component.

   For more information about the Applications editor, see "Managing applications" on page 1073.

Results

The data collector is configured to monitor the application server instance. Remember that data collection can increase the application server overhead. You can control the data collection with more advanced configuration options for tuning.

Checking user access requirements

The WebSphere Applications agent has some user access requirements for the user ID that is to configure the data collector.

About this task

Use the ID that is used to install the application server to configure the data collector after you grant appropriate permissions for the application server installation ID.

Procedure

- Windows Use the administrator ID that is used to install the application server to configure the data collector. Make sure that this user ID has full write permission the data collector home directory, *install_dir*\dchome\7.3.0.14.08.
- Linux AIX Use the user ID that is used to install the application server to configure the data collector. Make sure that this user ID has read and write permissions to the following subdirectories within *install_dir*/yndchome/7.3.0.14.08:
  - bin
  - data
  - logs
  - runtime

  Remember: If you use different user IDs to install application servers, you might need to use different user IDs to configure the data collector. After you configure the data collector for the first time, grant the write permission to the following files every time you use a different user ID to configure the data collector, where *profile_name* is the application server profile name:

  - *install_dir*/yndchome/7.3.0.14.08/data/findservers.inputlist
  - *install_dir*/yndchome/7.3.0.14.08/data/*profile_name*.findservers.progress
  - *install_dir*/yndchome/7.3.0.14.08/data/config_inputlist
  - *install_dir*/yndchome/7.3.0.14.08/runtime/custom/connections.properties

Handling other existing data collector in the application server

If a data collector already exists in the application server, you must decide what to do with the existing data collector, so that it does not conflict with the WebSphere Applications agent data collector.

About this task

The following types of data collector might already exist in the application server that is to be monitored:

- The data collector of WebSphere Applications agent, which is installed in a previous version of IBM Cloud Application Performance Management
- The data collector of ITCAM Agent for WebSphere Applications, which is installed in the old IBM® Tivoli® Monitoring infrastructure
- Any other data collector that is not provided by IBM

Procedure

Take appropriate actions to avoid the data collector conflicts.

- For a previous version of the WebSphere Applications agent data collector, which is installed in a previous version of IBM Cloud Application Performance Management, you have the following options:
  - Migrate the data collector with the migration utility from the latest data collector home directory. For instructions, see "WebSphere Applications agent: Migrating the data collector" on page 1118.
  - Unconfigure the previous version of the data collector and then configure the data collector again with the configuration utility from latest data collector home directory. For information about unconfiguring the data collector, see "WebSphere Applications agent: Unconfiguring the data collector" on page 149.
- For the data collector of ITCAM Agent for WebSphere Applications, complete the following steps if you want to deploy monitoring in an agent coexistence environment:
  a) Uninstall the data collector of ITCAM Agent for WebSphere Applications.
  b) Configure only one data collector to send data to both WebSphere Applications agent and ITCAM Agent for WebSphere Applications. For instructions, see "(Agent coexistence) Configuring the WebSphere Applications agent and the data collector" on page 853.
- For other data collectors that are not provided by IBM, evaluate whether it is necessary to remove these data collectors. The WebSphere Applications agent data collector uses the Java Byte Code manipulation to collect data. Other data collectors that use the same way to collect data might conflict with the WebSphere Applications agent data collector.

Configuring the data collector with the simple configuration utility

The WebSphere Applications agent starts automatically after installation, but you must manually configure the data collector, which is a component of the agent, to monitor application server instances.

Before you begin

- Make sure that the user access requirements are met in your environment. For instructions, see "Checking user access requirements" on page 831.
- If other data collector exists in the application server that is to be monitored, take appropriate actions to avoid data collector conflicts. For instructions, see "Handling other existing data collector in the application server" on page 831.

About this task

Important:

- If you want to configure the data collector only for resource monitoring or to set extra options, use the full configuration procedure. For instructions, see "Configuring or reconfiguring the data collector with full configuration utilities" on page 834.
- If you want to change the name of the server in the monitoring user interface, reconfigure the data collector and specify a server alias. For instructions, see "Configuring or reconfiguring the data collector with full configuration utilities" on page 834.

For the WebSphere Applications agent, the *dc_home* variables refer to the home directory of the data collector. The location of the *dc_home* variable on each operating system is as follows:

- Windows `install_dir\dchome\7.3.0.14.08`
- Linux AIX `install_dir/yndchome/7.3.0.14.08`

Procedure

1. Log in to the system with the user ID that is used to install the application server.
2. Change to the `bin` directory within the data collector home directory.
   - Windows `install_dir\dchome\7.3.0.14.08\bin`

- Linux AIX *install_dir*/yndchome/7.3.0.14.08/bin

3. Run the following simple configuration utility:

   - Windows `simpleconfig.bat`
   - Linux AIX `./simpleconfig.sh`

   The **`simpleconfig`** utility automatically discovers the home directories of the application servers.
4. Follow the prompts to continue with the data collector configuration.

   You are required to do the following things, depending on the application server settings:

   - For traditional WebSphere Application Server:
     - Select the auto-discovered WebSphere installation directory or manually specify the installation directory.
     - Select the WebSphere Application Server profile to monitor.
     - Select the security properties profile to use or provide the user name and password of the WebSphere administrative console (if security is enabled for the application server).
   - For WebSphere Application Server Liberty:
     - Specify the full path of the Liberty home directory that contains the `bin` and `servers` directories (for example, `/opt/ibm/wlp`).
     - Specify the home directory of the JRE that is used by Liberty.
5. If possible, restart the application server instance after the data collector configuration completes.
   a) Go to the `bin` directory under the home directory for the application server profile. For example, `opt/IBM/WebSphere/AppServer/profiles/`*`profile_name`*`/bin`.

   b) Stop the application server by entering the **`stopServer`** command in the command console.

   - Linux AIX `./stopServer.sh` *`server_name`*
   - Windows `stopServer.bat` *`server_name`*

   c) When prompted, enter the user ID and password of WebSphere administrative console administrator.

   d) Start the application server again by entering the **`startServer`** command in the command console.

   - Linux AIX `./startServer.sh` *`server_name`*
   - Windows `startServer.bat` *`server_name`*

Results

- The data collector is configured to monitor all instances in a profile, or, for WebSphere Application Server Liberty, a single instance or multiple instances in the same directory. To monitor more profiles or instances, repeat the configuration.
- The data collector is configured within the server instances, providing maximum monitoring.
- For Cloud APM, Base, resource monitoring is enabled.
- For Cloud APM, Advanced, resource monitoring, transaction tracking, and diagnostic data is enabled.

Known limitation: When monitoring WebSphere Application Server Liberty, the data collector cannot generate Java Naming and Directory Interface (JNDI) events.

What to do next

- Log in to the Cloud APM console and use the Applications editor to add the monitored application server to the Application Performance Dashboard. For instructions on how to start the Cloud APM console, see

"Starting the Cloud APM console" on page 957. For information about using the Applications editor, see "Managing applications" on page 1073.

Remember: If the WebSphere Applications agent is configured to monitor WebSphere Portal Server, the agent is related to the WebSphere Portal Server application component on the Application Performance Dashboard, not WebSphere Application Server.

- If the monitoring user interface on the Application Performance Dashboard shows no information for the application server instance, restart the monitoring agent component of the WebSphere Applications agent by completing the following steps:
  - Windows

    ```
    cd install_dir\bin
    was-agent.bat stop
    was-agent.bat start
    ```

  - Linux AIX

    ```
    cd install_dir/bin
    ./was-agent.sh stop
    ./was-agent.sh start
    ```

Configuring or reconfiguring the data collector with full configuration utilities

To set additional configuration options, you can use the full configuration utilities (interactive or silent) to configure the data collector instead of the simple configuration utility. You can also use full configuration utilities to reconfigure the data collector when it is already configured. Also, you need to use the full configuration utility to configure monitoring for WebSphere Portal Server instances.

Before you begin

- Make sure that the user access requirements are met in your environment. For instructions, see "Checking user access requirements" on page 831.
- If other data collector exists in the application server that is to be monitored, take appropriate actions to avoid data collector conflicts. For instructions, see "Handling other existing data collector in the application server" on page 831.

About this task

The configuration and reconfiguration utilities can be found in the following directories:

- Windows *install_dir*\dchome\7.3.0.14.08\bin
- Linux AIX *install_dir*/yndchome/7.3.0.14.08/bin

Procedure

- The configuration utility is named **config**. You might need to configure the data collector with the full configuration utility in the following cases:
  - The **simpleconfig** configuration utility fails.
  - You want to configure monitoring for WebSphere Portal Server instances.
  - You want to specify a server alias that is displayed in the monitoring user interface during the data collector configuration.
  - You want to have more control of what data to be collected. For example, you want to use resource monitoring only and disable diagnostics data and transaction tracking.
  - You do not want to configure all application servers within the same profile at one time.
  - The data collector is not configured within the application server and you want to reconfigure it.

  For information about the interactive full configuration utility, see "Configuring the data collector interactively" on page 835.

- The reconfiguration utility is named **reconfig**. You might need to reconfigure the data collector in the following cases:

  – You want to reconfigure the data collector after it is configured either interactively or silently.

  For information about interactive reconfiguration utility, see “Reconfiguring the data collector interactively” on page 839.
- For silent configuration, see “Configuring the data collector in silent mode” on page 842.

Configuring the data collector interactively

Use the interactive configuration utility (`config.sh` or `config.bat`) to configure the data collector when the `simpleconfig` utility fails. You can use the `config.sh` or `config.bat` utility to configure the data collector for each application server instance that you want to monitor.

Before you begin

If you will configure the data collector to monitor WebSphere Application Server Liberty, set the **JAVA_HOME** system environment variable to the same JVM as the one used for the application server. For example, on a Windows system, set **JAVA_HOME** value to `C:\Program Files\IBM\java`. Or on a Linux system, run `export JAVA_HOME=/opt/IBM/java`.

About this task

Use the following full configuration utility to configure the data collector:

- Windows *install_dir*\dchome\7.3.0.14.08\bin\config.bat
- Linux AIX *install_dir*/yndchome/7.3.0.14.08/bin/config.sh

Procedure

To configure the data collector by responding to prompts, complete these steps:

1. Log in to the system with the user ID that is used to install the application server.
2. Go to the `bin` directory within the *dc_home* data collector home directory.
3. Start the configuration utility by issuing the following command:

   - Windows `config.bat`
   - Linux AIX `./config.sh`

   The configuration utility displays the IP addresses and host names of all network cards that are found on the local computer system.
4. Enter the number that corresponds to the IP address and host name. If the IP address and host name that you want to use are not on the list, enter the IP address or host name.
5. Specify the home directory of the application server that is to be monitored.

   - For traditional WebSphere Application Server, enter the number that corresponds to an auto-discovered application server home directory or specify a full path to an application server home directory.
   - For WebSphere Application Server Liberty, enter the full path to the WebSphere Application Server Liberty home directory that contains the `bin` and `servers` directories, for example `/opt/ibm/wlp`.
6. If you are configuring the data collector for WebSphere Application Server Liberty, you are prompted for the Java home directory. Specify the Java home directory that is used for the application server. For example, `/opt/IBM/java`.
7. When the configuration utility lists all profiles under the specified application server home directory, enter the number that corresponds to the application server profile that you want to configure.

- For traditional WebSphere Application Server, the configuration utility then indicates whether WebSphere Global Security is enabled for the WebSphere Application Server profile that you specified. If global security is not enabled, proceed to the Step "9" on page 836.
- For WebSphere Application Server Liberty, proceed to Step "10" on page 836.

8. If global security is enabled for the WebSphere Application Server profile, specify whether to retrieve security settings from a client properties file. Enter 1 to allow the configuration utility to retrieve the user name and password from the appropriate client properties file. Otherwise, enter 2 to enter the user name and password.

   The data collector communicates with the WebSphere Administrative Services by using the Remote Method Invocation (RMI) or the SOAP protocol. If global security is enabled for a profile, you must specify the user ID and password of a user who is authorized to log in to the WebSphere Application Server administrative console for the application server profile. Alternatively, you can encrypt the user name and password and store them in client properties files before you configure the data collector. You must use the `sas.client.props` file for an RMI connection, or the `soap.client.props` file for an SOAP connection.
9. When you are prompted for the host name of WebSphere administrative console, press Enter to accept the default or specify the host name or IP address of the WebSphere administrative console. The default value is `localhost`.

   Remember: For a Network Deployment environment, enter the host name or IP address of the Deployment Manager.
10. When the configuration utility lists all the server instances that are not configured yet for data collection, select one or more application server instances from the list. Enter the number that corresponds to the application server instance to configure for data collection or enter an asterisk (*) to configure all application server instances for data collection. To specify a subset of servers, enter the numbers that represent the servers, separated by commas.
    For example, `1,2,3`.

    Remember:

    - For a stand-alone environment, application server instances must be running during the configuration. (A WebSphere Application Server Liberty instance does not need to be running).
    - For a Network Deployment environment, the Deployment Manager must be running.
    - Ensure that the application server instances that you select are the actual servers that host the applications or services that you want to monitor.
11. In the **Integration with Agent for WebSphere Applications** section, specify that you want to integrate the data collector with the WebSphere Applications agent. You must enter 1 to select this integration option, and then press Enter.

    The selected server will be registered for PMI resource monitoring.
12. If you are configuring the data collector for traditional WebSphere Application Server, specify whether you want to configure the data collector within the application server instance.

    - Enter 1 to configure the data collector within the application server. With this option, the data collector is integrated with the application server, which is required for the full range of operational monitoring and diagnostics data collection. However, configuring the data collector within the application server requires restarting the application server. Also, the data collector might affect server performance.
    - Enter 2 to not to configure the data collector within the application server and proceed to Step "14" on page 836. With this option, the data collector runs as a stand-alone process and only resource monitoring can be enabled.
13. When prompted, specify whether to enable the data collector for diagnostics data. Enter 1 to enable diagnostics data collection. The default is 2.
14. When you are prompted for the host name of the V8 monitoring agent, enter the host name or IP address of the WebSphere Applications agent or press Enter to accept the default. The default value corresponds to your choice in Step 3.

The V8 monitoring agent refers to the WebSphere Applications agent, which is installed with IBM Cloud Application Performance Management.

15. When you are prompted for the port number of the V8 monitoring agent, enter the port number of the WebSphere Applications agent or press Enter to accept the default. The default is 63335.
16. When you are asked whether to configure V6 monitoring agent for WebSphere Applications, press Enter to accept the default for No.

    The V6 monitoring agent refers to the ITCAM Agent for WebSphere Applications, which is installed in the old IBM® Tivoli® Monitoring infrastructure. Configuring V6 monitoring agent is required only for agent coexistence environment.
17. When you are prompted for the server alias, press Enter to accept the default or enter another alias. If you are configuring several application server instances, the configuration utility prompts you for an alias for every instance.

    Important: The alias can contain only the following characters: `A-Z`, `a-z`, underbar (`_`), dash (`-`), and period (`.`). Do not use other characters in the alias.

    The server alias is the first qualifier of the agent instance name (also known as MSN) that is displayed on the Cloud APM console. The default is the node name combined with the server name. For example, the **node1server1** alias indicates the server named **server1** in the node named **node1**.
18. When you are prompted for a port number for PMI resource monitoring, press Enter to accept the default or enter a new number. The default port is 63355.

    This port is used for internal communication between components that are running on the same host. If the default is in use, you can set a different number.
19. In the **Support for transaction tracking** section, specify whether to enable transaction tracking. Enter 1 to enable support for transaction tracking. Otherwise, enter 2 and skip to Step "22" on page 837.
20. When you are prompted for host name or IP address of the Transaction Framework Extension, press Enter to accept the default or enter another host name or IP address.

    Transaction Framework Extension is an internal component of the WebSphere Applications agent that gathers metrics from the data collector.
21. When you are prompted for the port number that the data collector uses to connect to the Transaction Framework Extension, press Enter to accept the default or enter another port number. The default is 5457.
22. Specify whether to integrate the data collector with Application Performance Diagnostics Lite. Press Enter to accept the default for no.
23. In the **Advanced settings** section, specify whether to change the garbage collection log path. Enter 1 to select a garbage collection log path. Otherwise, enter 2 and skip to Step "25" on page 837. To use the log path that is already specified in the JVM argument of the application server, enter 2.
24. Specify the garbage collection log path. Enter a file name with its full path. For WebSphere Application Server Liberty, do not use variables in the path. The data collector automatically modifies the log file name, adding the server instance information to it.

    For example, if you specify `gc.log` as the file name, the actual name is set to *`profile_name.cell_name.node_name.server_name`*`.gc.log` for every configured application server instance.

    Important: In the garbage collection log path, you can use WebSphere variables such as `${SERVER_LOG_ROOT}`. However, do not use templates, such as `%pid`.
25. Review the summary of the data collector configuration that is to be applied to the specified application server instances. If necessary, reconfigure parts of the data collector configuration before you apply the changes.
26. Enter a to accept your changes.
27. When prompted, specify whether you want to create a backup of your current configuration. Enter 1 to create a backup of the current configuration. Otherwise, enter 2.

The configuration utility applies the changes and presents a status message to indicate that the configuration of the data collector for the profile is completed.

28. If you are configuring the data collector for traditional WebSphere Application Server, restart the application server instances or restart the agent, depending on your choice in Step "12" on page 836.

    - If you have enabled the data collector within the application server, restart the application server instances as indicated by the configuration utility.
    - If you have enabled PMI resource monitoring without enabling the data collector within the application server, restart the WebSphere Applications agent by running the following commands:

      – Windows

      ```
      cd install_dir\bin
      was-agent.bat stop
      was-agent.bat start
      ```

      – Linux AIX

      ```
      cd install_dir/bin
      ./was-agent.sh stop
      ./was-agent.sh start
      ```

    The data collector configuration takes effect after the application server or agent restart.

29. Log in to the Cloud APM console to view data in the dashboards.

    a) Access the console by using the link that is provided in the email alerting you that your service is ready. Alternatively, access the console from the IBM Marketplace website. For detailed instructions, see "Starting the Cloud APM console" on page 957.

    b) Use the Applications editor to add the monitored application server to the Application Performance Dashboard. You can add it as a new component to your existing application or create an application to contain this component.

       For more information about the Applications editor, see "Managing applications" on page 1073.

What to do next

- If the current user ID that is used to configure the data collector is not the same ID of the user running the application server, verify that the user ID for configuring the data collector has read and write permissions to the `runtime` and `logs` directories within the data collector home directory. These two subdirectories are created by the ID of the user running the application server when the server is restarted.
- Log in to the Cloud APM console to view the monitoring data in the dashboards. If monitoring data are not available immediately, restart the WebSphere Applications agent by running the following commands:

  – Windows

  ```
  cd install_dir\bin
  was-agent.bat stop
  was-agent.bat start
  ```

  – Linux AIX

  ```
  cd install_dir/bin
  ./was-agent.sh stop
  ./was-agent.sh start
  ```

- Changing the server alias changes the agent instance name that is registered with the Cloud APM console. If this is not the first time that you configure the data collector and you changed the server alias, you must clear some cache files by completing the following steps:

  1. Stop the monitoring agent if it is running.

2. Open the *hostname*_yn.xml file in the following directory with a text editor, where *hostname* is the name of the host where the WebSphere Applications agent is installed.
   - Windows *install_dir*\TMAITM6_x64 (Default is C:\IBM\APM\TMAITM6_x64)
   - Linux AIX *install_dir*/config (Default is /opt/ibm/apm/agent/config)
3. Locate the line that starts with the following string and contains the previous server name. For example, was85.win4net01Cell02.win4net01Node02.AppSrv01.server1, where server1 is previous name of the application server.

   ```
   <!ENTITY was_product_code.cellname.nodename.profilename.servername
   ```

   where *was_product_code* is the product code of WebSphere Application Server; *cellname* is the name of the cell; *nodename* is the node name; *profilename* is the name of the application server profile; *servername* is the previous name of the application server.
4. Locate the .XML file that is indicated in the line within the current directory and delete the file.
5. Remove the line that you located in Step 3 from the *hostname*_yn.xml file.
6. At the end of the *hostname*_yn.xml file, remove the line that contains the previous server names.
7. Save and close the file.
8. Restart the monitoring agent.

Reconfiguring the data collector interactively

If you configured the data collector to monitor one or more application server instances, you can reconfigure the data collector by using the reconfiguration utility (reconfig.sh or reconfig.bat).

Before you begin

If you will configure the data collector to monitor WebSphere Application Server Liberty, set the **JAVA_HOME** system environment variable to the same JVM as the one used for the application server. For example, on a Windows system, set **JAVA_HOME** value to C:\Program Files\IBM\java. Or on a Linux system, run export JAVA_HOME=/opt/IBM/java.

About this task

Use the following full reconfiguration utility to configure the data collector:

- Windows *install_dir*\dchome\7.3.0.14.08\bin\reconfig.bat
- Linux AIX *install_dir*/yndchome/7.3.0.14.08/bin/reconfig.sh

Remember: The **reconfig** utility is not applicable in the following cases. Use the **config** configuration utility instead. Although the **config** utility warns that the server is already configured, but it still can make any required changes.

- The data collector is already configured for resource monitoring only and you want to reconfigure the data collector.
- You want to reconfigure the data collector for WebSphere Portal Server.

Tip: In the prompts asking for agent configuration settings, the reconfiguration utility offers the currently configured values as defaults.

Procedure

To reconfigure the data collector by responding to prompts, complete these steps:

1. Log in to the system with the user ID that is used to install the application server.
2. Go to the bin directory within the *dc_home* data collector home directory.
3. Start the reconfiguration utility by issuing the following command:
   - Windows reconfig.bat

- Linux AIX `./reconfig.sh`

  Tip: Running this reconfiguration utility has the same effect as running the `config.bat` script with the `-reconfig` argument on Windows systems or the `config.sh` script with the `-reconfig` argument on Linux or AIX systems.

  The reconfiguration utility displays the IP addresses of all network cards that are found on the local computer system.

4. Enter the number that corresponds to the IP address to use.

   The reconfiguration utility displays all application server instances for which the data collector is configured on this host, and prompts you to select one or more application server instances from the list.
5. Select one or more application server instances from the list. Enter the number that corresponds to the application server instance to reconfigure for data collection or enter an asterisk (`*`) to reconfigure all application server instances for data collection. To specify a subset of servers, enter the numbers, separated by commas, that represent the servers. For example: `1,2,3`.

   Remember:

   - For a stand-alone environment, application server instances must be running during the configuration. (A WebSphere Application Server Liberty instance does not need to be running).
   - For a Network Deployment environment, the Deployment Manager must be running.
   - Ensure that the application server instances that you select are the actual servers that host the applications or services that you want to monitor.
6. In the **Integration with Agent for WebSphere Applications** section, specify that you want to integrate the data collector with the WebSphere Applications agent. You must enter 1 to select this integration option, and then press Enter.
7. If you are configuring the data collector for traditional WebSphere Application Server, specify whether you want to configure the data collector within the application server instance.

   - Enter 1 to configure the data collector within the application server. With this option, the data collector is integrated with the application server, which is required for the full range of operational monitoring and diagnostics data collection. However, configuring the data collector within the application server requires restarting the application server. Also, the data collector might affect server performance.
   - Enter 2 to not to configure the data collector within the application server and process to Step "9" on page 840. With this option, the data collector runs as a stand-alone process and only PMI resource monitoring can be enabled.
8. When prompted, specify whether to enable diagnostics data collection for the data collector. Enter 1 for yes, or enter 2 for no.
9. When you are prompted for the host name, enter the host name or IP address of the WebSphere Applications agent or press Enter to accept the default. The default value corresponds to your choice in Step "4" on page 840.
10. When you are prompted for the port number, enter the port number of the monitoring agent or press Enter to accept the default. The default is 63335.
11. When you are asked whether to configure V6 monitoring agent for WebSphere Applications, press Enter to accept the default for No.

    The V6 monitoring agent refers to the ITCAM Agent for WebSphere Applications, which is installed in the old IBM® Tivoli® Monitoring infrastructure. Configuring V6 monitoring agent is required only for agent coexistence environment.
12. When you are prompted for the server alias, press Enter to accept the default or enter another alias. If you are configuring several application server instances, the configuration utility prompts you for an alias for every instance.

    Important: The alias can contain only the following characters: `A-Z`, `a-z`, underbar (_), dash (-), and period (`.`). Do not use other characters in the alias.

The server alias is the first qualifier of the agent instance name (also known as MSN) that is displayed on the Cloud APM console. The default is the node name combined with the server name. For example, the **node1server1** alias indicates the server named **server1** in the node named **node1**.

13. When you are prompted for a port number for PMI resource monitoring, press Enter to accept the default or enter a new number. The default port is 63355.

    This port is used for internal communication between components that are running on the same host. If the default is in use, you can set a different number.
14. In the **Support for transaction tracking** section, specify whether to enable transaction tracking. Enter 1 to enable support for transaction tracking. Otherwise, enter 2 and skip to Step "17" on page 841.
15. When you are prompted for host name or IP address of the Transaction Framework Extension, press Enter to accept the default or enter another host name or IP address.

    Transaction Framework Extension is an internal component of the WebSphere Applications agent that gathers metrics from the data collector.
16. When you are prompted for the port number that the data collector uses to connect to the Transaction Framework Extension, press Enter to accept the default or enter another port number. The default is 5457.
17. Specify whether to integrate the data collector with Application Performance Diagnostics Lite. Press Enter to accept the default for No.
18. In the **Advanced settings** section, specify whether to change the garbage collection log path.

    Enter 1 to select a garbage collection log path. Otherwise, enter 2 and skip to Step "20" on page 841. To use the log path that is already specified in the JVM argument of the application server, enter 2.
19. Specify the garbage collection log path. Enter a file name with its full path. For WebSphere Application Server Liberty, do not use variables in the path. The data collector automatically modifies the log file name, adding the server instance information to it.

    For example, if you specify `gc.log` as the file name, the actual name is set to *`profile_name.cell_name.node_name.server_name`*`.gc.log` for every configured application server instance.

    Important: In the garbage collection log path, you can use WebSphere variables such as `${SERVER_LOG_ROOT}`. However, do not use templates, such as `%pid`.
20. Review the summary of the data collector configuration that is to be applied to the specified application server instances. Reconfigure parts of the data collector configuration before you apply the changes, if required.
21. Enter a to accept your changes.
22. When prompted, specify whether you want to create a backup of your current configuration. Enter 1 to create a backup of the current configuration. Otherwise, enter 2.

    The configuration utility applies the changes and presents a status message to indicate that the configuration of the data collector for the profile is completed.
23. If you are configuring the data collector for traditional WebSphere Application Server, restart the application server instances or restart the agent, depending on your choice in Step "7" on page 840.

    - If you have enabled the data collector within the application server, restart the application server instances as indicated by the configuration utility.
    - If you have enabled PMI resource monitoring without enabling the data collector within the application server, restart the WebSphere Applications agent by running the following commands:

        – Windows

        ```
        cd install_dir\bin
        was-agent.bat stop
        was-agent.bat start
        ```

        – Linux AIX

```
cd install_dir/bin
./was-agent.sh stop
./was-agent.sh start
```

The data collector configuration takes effect after the application server or agent restart.

What to do next

- Changing the server alias changes the agent instance name that is registered with the Cloud APM console. If you changed the server alias during the reconfiguration procedure, you must clear some cache files by completing the following steps:

  1. Stop the monitoring agent if it is running.
  2. Open the `*hostname*_yn.xml` file in the following directory with a text editor, where *hostname* is the name of the host where the WebSphere Applications agent is installed.
     – Windows *`install_dir`*`\TMAITM6_x64` (Default is `C:\IBM\APM\TMAITM6_x64`)
     – Linux AIX *`install_dir`*`/config` (Default is `/opt/ibm/apm/agent/config`)
  3. Locate the line that starts with the following string and contains the previous server name. For example, `was85.win4net01Cell02.win4net01Node02.AppSrv01.server1`, where `server1` is previous name of the application server.

     ```
     <!ENTITY was_product_code.cellname.nodename.profilename.servername
     ```

     where *was_product_code* is the product code of WebSphere Application Server; *cellname* is the name of the cell; *nodename* is the node name; *profilename* is the name of the application server profile; *servername* is the previous name of the application server.
  4. Locate the `.XML` file that is indicated in the line within the current directory and delete the file.
  5. Remove the line that you located in Step 3 from the `*hostname*_yn.xml` file.
  6. At the end of the `*hostname*_yn.xml` file, remove the line that contains the previous server names.
  7. Save and close the file.
  8. Restart the monitoring agent.

Configuring the data collector in silent mode

If you want to configure many application server instances, it might be more convenient to configure the data collector in silent mode.

About this task

When you configure the data collector in silent mode, you first specify configuration options in a properties file. A sample properties file, `sample_silent_config.txt`, is packaged with the configuration utility. The file is available in the following directories, where *dc_home* is the directory where the data collector is installed. For the full path of the *dc_home* directory, see the introduction in Configuring the data collector for WebSphere Applications agent.

- Windows *`dc_home`*`\bin`
- Linux AIX *`dc_home`*`/bin`

For detailed information about each available configuration property in this file, see “Properties file for silent configuration of data collector” on page 844.

Procedure

Complete the following steps to perform a silent configuration:

1. Specify configuration options in the properties file. You can copy the sample properties file and change the required options.
2. Set the location of the Java home directory before you run the utility.

For example:

- Windows

  ```
  set JAVA_HOME=C:\Program Files\IBM\WebSphere\AppServer80\java
  ```

- Linux AIX

  ```
  export JAVA_HOME=/opt/IBM/AppServer80/java
  ```

Important: If you are configuring monitoring for WebSphere Application Server Liberty, you must use same JVM version as the one used for the application server. Otherwise, the monitoring might fail.

3. Go to the following directory:
   - Windows *dc_home*\bin
   - Linux AIX *dc_home*/bin
4. Run the command to configure the data collector in silent mode.
   - Windows Run the following command as the administrator who installed the WebSphere Application Server.

     ```
     config.bat -silent [dir_path]\silent file
     ```

   - Linux AIX Run the following command with root user privileges.

     ```
     config.sh -silent [dir_path]/silent file
     ```

   Tip: If the `wsadmin` user was used to install the application server, run the `config` utility either as the `wsadmin` user or with root user privileges.
5. After configuring the data collector to monitor application server instances, if you have enabled the data collector within the application server, you must restart the instances. The data collector configuration takes effect when the application server instances are restarted.
6. If you have enabled PMI resource monitoring without enabling the data collector within the application server, you might need to restart the WebSphere Applications agent to start the monitoring. If monitoring data is not available immediately, restart the monitoring agent by running the following commands:
   - Windows

     ```
     cd install_dir\bin
     was-agent.bat stop
     was-agent.bat start
     ```

   - Linux AIX

     ```
     cd install_dir/bin
     ./was-agent.sh stop
     ./was-agent.sh start
     ```

What to do next

After silent configuration, to reconfigure the data collector, you have two options:

- Reconfigure it interactively by using the **`reconfig`** reconfiguration utility. For instructions, see "Reconfiguring the data collector interactively" on page 839.
- Unconfigure it silently and then use the same procedure to configure it silently again. For instructions, see "Unconfiguring the data collector in silent mode" on page 151.

Related reference

"Properties file for silent configuration of data collector" on page 844

To silently configure the data collector, you first specify configuration options in a properties file and then run the configuration utility.

Properties file for silent configuration of data collector

To silently configure the data collector, you first specify configuration options in a properties file and then run the configuration utility.

When you create your properties file, keep in mind the following considerations:

- A line in the file that starts with a number sign (#) is treated as a comment, and is not processed. If the number sign is used elsewhere in the line, it is not considered to be the start of a comment.
- Each property is described on a separate line, in the following format: *property* = *value*.

  property

  Name of property. The list of valid properties that you can configure is shown in Table 228 on page 844.

  value

  Value of the property. Default values for some properties are already provided. You can delete default values to leave property values blank or empty. An empty value is treated as if the property is not specified, as opposed to using the default value. If you want to use default values, you can comment out the property in the file.
- Passwords are in plain text.
- Properties and their values are case-sensitive.

Table 228 on page 844 describes the properties that are available when configuring the data collector in silent mode.

Important: If you are configuring the data collector for a WebSphere Application Server Liberty instance, some of the properties are not used.

Table 228. Available properties for running the configuration utility in silent mode

Property	**Comment**
default.hostip	If the computer system uses multiple IP addresses, specify the IP address for the data collector to use.
Integration of the data collector with the ITCAM for Application Diagnostics Managing Server **Important:** The Managing Server is only available if you have ITCAM for Application Diagnostics. For a WebSphere Application Server Liberty instance or in a Cloud APM environment, these properties are not used.	
ms.connect	Specifies whether the data collector is configured to connect to the managing server in an ITCAM for Application Diagnostics environment. Valid values are `True` and `False`.
ms.kernel.host	Specifies the fully qualified host name of the managing server.
ms.kernel.codebase.port	Specifies the codebase port on which the managing server is listening.
ms.am.home	Specifies the managing server home directory.
ms.am.socket.bindip	Specifies the IP address or host name to be used by the data collector to communicate with the managing server. If more than one network interface or IP address is configured on data collector computer system, choose one of them.

Table 228. Available properties for running the configuration utility in silent mode (continued)

Property	Comment
ms.probe.controller.rmi.port	If the data collector is behind a firewall or you have special requirements to change the Controller RMI port of data collector, set this port number range. Configure this port number as permitted by the firewall for the data collector host. For example: `ms.probe.controller.rmi.port=8300-8399` or ms.probe.controller.rmi.port=8300.
ms.probe.rmi.port	If the data collector is behind a firewall, or you have special requirements to change the RMI port of data collector, set this port number range. Configure this port number as permitted by the firewall for the data collector host. For example: `ms.probe.rmi.port=8200-8299` or `ms.probe.rmi.port=8200`.
Support for transaction tracking To view transaction tracking information, you must have topology views available in the Cloud APM console and enable transaction tracking in the Agent Configuration window.	
ttapi.enable	Specifies whether the data collector supports transaction tracking. Valid values are `True` and `False`.
ttapi.host	Specifies the host of the Transaction Framework Extension, which is the component of the Monitoring Agent for WebSphere Applications that gathers metrics from the data collector. use the local host value, `127.0.0.1`.
ttapi.port	Specifies the port of the Transaction Framework Extension. Use 5457.
Integration of the data collector with ITCAM for SOA **Important:** For a WebSphere Application Server Liberty instance or in a Cloud APM environment, this property is not used.	
soa.enable	Specifies whether to integrate the data collector with ITCAM for SOA. The ITCAM for SOA agent must be installed to complete the configuration.
Integration of the data collector with the Tivoli Performance Monitoring **Important:** For a WebSphere Application Server Liberty instance or in a Cloud APM environment, this property is not used.	
tpv.enable	Specifies whether to integrate the data collector with the Tivoli Performance Monitoring when the data collector is included as part of ITCAM for WebSphere Application Server version 8.5. Tivoli Performance Monitoring is accessed with the WebSphere Application Server administrative console. Valid values are *True* and *False*.
Integration of the data collector with Application Performance Diagnostics Lite **Important:** For a WebSphere Application Server Liberty instance, this property is not used.	

Table 228. Available properties for running the configuration utility in silent mode (continued)

Property	Comment
de.enable	Specifies whether to collect diagnostics data, required for Application Performance Diagnostics and Application Performance Diagnostics Lite. Valid values are `True` and `False`. Enable this integration if you have Application Diagnostics or might have it in the future. In this case, collection of diagnostic data is enabled at server startup. Otherwise, it is disabled at startup; you can enable it using the Agent Configuration page in the user interface, but if the server is restarted, collection of diagnostic data is disabled again. This setting also enables integration with Application Performance Diagnostics Lite, which is a tool for diagnostic investigation of applications running on WebSphere Application Server and WebSphere Portal Server. Using this tool, you can analyze data in real time or you can save diagnostic information to a file for later analysis.
PMI resource and data collector monitoring The selected server is always configured for resource (PMI) monitoring, without any changes to the application server. This monitoring option provides limited metrics and works only with WebSphere Applications agent, but does not require restarting the application server and can not affect performance.	
tema.appserver	Specifies whether you want to configure the data collector within the application server instance. The data collector within the application server instance is required for the full range of metrics in WebSphere Applications agent and for integration with any other products. However, configuring the data collector requires restarting the application server. Also, the data collector might affect server performance. Valid values are `True` and `False`. If this parameter is set to `False`, data collector configuration parameters for integrating with products other than WebSphere Applications agent are disregarded. When this parameter is set to `False`, diagnostics and transaction tracking features are not available, and only resource monitoring data is collected.
tema.jmxport	TCP/IP port number for resource monitoring. The port is used for internal communication between components running on the same host. The default port is 63355; if this port is in use, you can set a different number.
Integration of the data collector with the monitoring agent component of WebSphere Applications agent and with Application Performance Diagnostics Lite	
temaconnect	Specifies whether the data collector connects to the monitoring agent component of WebSphere Applications agent. Valid values are `True` and `False`. **Important:** You must use the `True` value to use the WebSphere Applications agent.

Table 228. Available properties for running the configuration utility in silent mode (continued)

Property	Comment
tema.appserver	Specifies whether you want to configure the data collector within the application server instance. The data collector within the application server instance is required for the full range of metrics in the WebSphere Applications agent and for integration with any other products. However, it requires restarting the application server. Also, the data collector might affect server performance. Valid values are `True` and `False`. If this parameter is set to `False`, the configuration parameters for integrating data collector with products other than WebSphere Applications agent are disregarded, as well as the following tema.host and tema.port parameters. When this parameter is set to `False`, diagnostics and transaction tracking features are not available, and only resource monitoring data is collected.
tema.host	Specifies the fully qualified host name or IP address of the monitoring agent component of WebSphere Applications agent. Use the local host address (127.0.0.1).
tema.port	Specifies the port number of the monitoring agent component of WebSphere Applications agent. Do not change the default value of `63335`.
tema.jmxport	TCP/IP port number for resource monitoring. The port is used for internal communication between components running on the same host. The default port is 63355; if this port is in use, you can set a different number.
Integration of the data collector with ITCAM Agent for WebSphere Applications version 6 Use the following properties to configure one data collector to collect data for both WebSphere Applications agent and ITCAM Agent for WebSphere Applications version 6.	
config.tema.v6	Specifies whether to integrate the data collector with the monitoring agent component of ITCAM Agent for WebSphere Applications version 6. Valid values are `True` and `False`. The default is `False`.
tema.host.v6	Specifies whether to integrate the data collector with the monitoring agent component of ITCAM Agent for WebSphere Applications version 6. Valid values are `True` and `False`. The default is `False`.
tema.port.v6	Specifies the port number of the monitoring agent component of ITCAM Agent for WebSphere Applications version 6. Do no change the default value 63336.
WebSphere Application Server backup	
was.backup.configuration	Specifies whether to back up the current configuration of the WebSphere Application Server configuration before applying the new configuration. Valid values are `True` and `False`.
was.backup.configuration.dir	Specifies the location of the backup directory.
Advanced configuration settings	
was.gc.custom.path	Specifies whether to set a custom path for the Garbage Collection log.

Table 228. Available properties for running the configuration utility in silent mode (continued)

Property	**Comment**
was.gc.file	Specifies the path to the custom Garbage Collection log. Set this value to a file name with its full path. The data collector automatically modifies the log file name, adding the server instance information to it. For example, if you specify `gc.log` as the file name, the actual name is set to *`profile_name`*`.`*`cell_name`*`.`*`node_name`*`.`*`server_name`*`.gc.log` for every configured application server instance. **Important:** In the Garbage Collection log path, you can use WebSphere variables, such as `${SERVER_LOG_ROOT}`. However, do not use templates, such as %pid.
WebSphere Administrative Services connection settings	
was.wsadmin.connection.host	Specifies the name of the host to which the wsadmin tool is connecting. In a Network Deployment environment, specify the wsadmin connection to the Deployment Manger. In a stand-alone environment, specify the wsadmin connection to the server. **Remember:** If the WebSphere Administrative console is on the same system, the value of `localhost` will be used for connection. However, in some cases, `localhost` is not allowed for communication due to system network or security settings. In that case, you must specify this parameter in the silent response file.
was.wsadmin.connection.type	Specifies the port that the wsadmin tool must use to connect to the WebSphere Application Server.
was.wsadmin.connection.port	Specifies the port that the wsadmin tool must use to connect to the WebSphere Application Server.
WebSphere Application Server global security settings	
was.wsadmin.username	Specifies the user ID of a user who is authorized to log in to the IBM WebSphere Application Server administrative console. This user must have the agent role on the application server.
was.wsadmin.password	Specifies the password that corresponds to the user specified in the `was.wsadmin.username` property.
was.client.props	Specifies whether to retrieve security settings from a client properties file. Possible values are `True` and `False`.
WebSphere Application Server settings	
was.appserver.profile.name	Specifies the name of the application server profile that you want to configure. Not used for WebSphere Application Server Liberty.
was.appserver.home	Specifies the WebSphere Application Server home directory.
was.appserver.cell.name	Specifies the WebSphere Application Server cell name. Not used for WebSphere Application Server Liberty.
was.appserver.node.name	Specifies the WebSphere Application Server node name. Not used for WebSphere Application Server Liberty.
WebSphere Application Server runtime instance settings	

Table 228. Available properties for running the configuration utility in silent mode (continued)

Property	Comment
was.appserver.server.name	Specifies the application server instance within the application server profile to configure. **Tip:** • The silent response file can have multiple instances of this property • When adding a second server, uncomment the second server (this is, #[SERVER]) and add the server name.
tema.serveralias	Specifies the name of the node in monitoring user interface that contains the monitoring information for this application server instance. The default is the node name combined with the server name. **Important:** The alias can contain only the following characters: `A-Z`, `a-z`, underbar (`_`), dash (`-`), and period (`.`). Do not use other characters in the alias. **Tip:** The silent response file can have multiple instances of this property. **Remember:** Changing the server alias changes the agent instance name that is registered with the Cloud APM console. If this is not the first time that you configure the data collector and you changed the server alias, you must clear some cache files. For detailed instructions, see Clearing the cache files with the old server names.

Manually configure the data collector if the configuration utilities fail
If you cannot use the provided configuration utility to configure the data collector for WebSphere Applications agent, you can manually configure the data collector in the WebSphere Administrative Console.

Before you begin

- Install the WebSphere Applications agent.
- Get to know the data collector home directory, which is required by the data collector configuration. The default is `/opt/ibm/apm/agent/yndchome/7.3.0.14.08` on Linux and UNIX systems or `C:\IBM\APM\dchome\7.3.0.14.08` on Windows systems.
- If you want to configure the data collector for a Liberty server, get to know the Liberty server home directory. For example, `/opt/ibm/was/liberty/usr/servers/defaultServer`.
- Make sure that a file named `itcam_wsBundleMetaData.xml` exists in the *dc_home*`/runtime/wsBundleMetaData` folder and it contains the following content. If the folder or the file does not exist, manually create it.

  Remember: The *plugins_dir_within_dc_home* value must be set to the absolute path of the `plugins` folder within the data collector home directory. The default is `/opt/ibm/apm/agent/yndchome/7.3.0.14.08/plugins` on Linux and UNIX systems or `C:\IBM\APM\dchome\7.3.0.14.08\plugins` on Windows systems.

```
<bundles>
  <directory path="plugins_dir_within_dc_home">
      <bundle>com.ibm.tivoli.itcam.bundlemanager_7.2.0.jar</bundle>
  </directory>
  <directory path="plugins_dir_within_dc_home">
      <bundle>com.ibm.tivoli.itcam.classicsca_7.2.0.jar</bundle>
  </directory>
  <directory path="plugins_dir_within_dc_home">
      <bundle>com.ibm.tivoli.itcam.toolkitsca.classicsca_7.2.0.jar</bundle>
  </directory>
</bundles>
```

About this task

Important:

- You must make manual changes to the WebSphere Application Server configuration for data collectors as the WebSphere administrative user.
- You must be an experienced WebSphere administrator to make manual changes to the WebSphere Application Server for data collection. Any error in the manual configuration change can result in the application server not starting.
- After you manually configure the data collector to monitor application server instances, you cannot use the unconfiguration utility to unconfigure the data collector. You must manually unconfigure the data collector instead.

Procedure

- To manually configure the data collector for the WebSphere application server, see "Manually configuring data collector for WebSphere Application Server traditional" on page 850.
- To manually configure the data collector for the Liberty server, see "Manually configuring the data collector for WebSphere Application Server Liberty" on page 852.

Manually configuring data collector for WebSphere Application Server traditional

Procedure

1. Log in to the WebSphere Administrative Console as the administrator.
2. In the navigation pane, click **Servers**, expand **Server Types** and click **WebSphere application servers**.
3. Under the **Server Infrastructure** section in the Configuration tab, expand **Java and Process Management** and click **Process Definition**.
4. Under the **Additional Properties** section, click **Java Virtual Machine**.
5. In the **Generic JVM arguments** field, add the following entries.

   ```
   -agentlib:am_ibm_16=${WAS_SERVER_NAME} -Xbootclasspath/p:${ITCAMDCHOME}/
   toolkit/lib/bcm-bootstrap.jar -Djava.security.policy=${ITCAMDCHOME}/itcamdc/
   etc/datacollector.policy -verbosegc
   ```

   When you add the entries, take note of the following:

   - All entries must be on a single line.
   - Separate different arguments by spaces before the minus sign (-), and do not use spaces anywhere else.
6. Click **Apply** and then save the changes to the master configuration.
   - If you are not under a Network Deployment environment, click **Save**.
   - If you are under a Network Deployment environment, ensure that **Synchronize changes with Nodes** is selected in the **Console preferences**options and then click **Save**.
7. In the navigation pane, click **Servers**, expand **Server Types**, click **WebSphere application servers** and then click the server name.
8. In the Configuration tab, go to **Server Infrastructure** > **Java and Process Management** > **Process Definition** > **Environment Entries**.
9. Depending on the operating system, the hardware platform, and the application server JVM, set the following environment entry.

Table 229. Environment entry

Platform	Environment entry name	Environment entry value
AIX R6.1 (64-bit JVM)	LIBPATH	`/lib:${ITCAMDCHOME}/toolkit/lib/aix536`
AIX R7.1 (64-bit JVM)	LIBPATH	`/lib:${ITCAMDCHOME}/toolkit/lib/aix536`
Solaris 10 (64-bit JVM)	LIBPATH	`/lib:${ITCAMDCHOME}/toolkit/lib/sol296`
Solaris 11 (64-bit JVM)	LIBPATH	`/lib:${ITCAMDCHOME}/toolkit/lib/sol296`
Linux Intel R2.6 (32-bit JVM)	LD_LIBRARY_PATH	`/lib:${ITCAMDCHOME}/toolkit/lib/li6263`
Linux x86_64 R2.6 (64-bit JVM)	LD_LIBRARY_PATH	`/lib:${ITCAMDCHOME}/toolkit/lib/lx8266`
Linux on Power Little Endian (64-bit JVM)	LD_LIBRARY_PATH	`/lib:${ITCAMDCHOME}/toolkit/lib/lpl266`
Linux on System z (32-bit JVM)	LD_LIBRARY_PATH	`/lib:${ITCAMDCHOME}/toolkit/lib/ls3263`
Linux on System z (64-bit JVM)	LD_LIBRARY_PATH	`/lib:${ITCAMDCHOME}/toolkit/lib/ls3266`
Windows (32-bit JVM)	PATH	`/lib;${ITCAMDCHOME}/toolkit/lib/win32`
Windows (64-bit JVM)	PATH	`/lib;${ITCAMDCHOME}/toolkit/lib/win64`

10. Click **Apply** and then save the changes to the master configuration.

    - If you are not under a Network Deployment environment, click **Save**.
    - If you are under a Network Deployment environment, ensure that **Synchronize changes with Nodes** is selected in the **Console preferences**options and then click **Save**.

11. In the navigation pane, click **Environment** > **WebSphere Variables**.
12. Specify the scope to appropriate server level and add the *ITCAMDCHOME* variable. Set the *ITCAMDCHOME* variable value to the data collector home directory. For example, `/opt/ibm/apm/agent/yndchome/7.3.0.14.08`.
13. Click **Apply** and then save the changes to the master configuration.

    - If you are not under a Network Deployment environment, click **Save**.
    - If you are under a Network Deployment environment, ensure that **Synchronize changes with Nodes** is selected in the **Console preferences**options and then click **Save**.

14. Restart the application server.

Results

Now you can check the WebSphere Applications agent data in the Cloud APM console after you add this application component to your applications. For instructions on how to start the Cloud APM console, see "Starting the Cloud APM console" on page 957. For instructions on how to add or edit an application, see "Managing applications" on page 1073.

What to do next

After you manually configure the data collector, you cannot use the provided `unconfig` utility to unconfigure the data collector. Manually unconfigure the data collector instead. For instructions, see "Manually unconfigure the data collector" on page 155.

Manually configuring the data collector for WebSphere Application Server Liberty

Procedure

1. Navigate to the Liberty server home directory. For example, `/opt/ibm/wlp/usr/servers/defaultServer`.
2. Edit the `jvm.options` file by adding the following parameters, where *dc_home* is the data collector home directory and *server_name* is the Liberty server name.. If the `jvm.options` file does not exist, create it with a text editor.

   ```
   -agentlib:am_ibm_16=server_name
   –Xbootclasspath/p:dc_home/toolkit/lib/bcm-bootstrap.jar
   -Djava.security.policy=dc_home/itcamdc/etc/datacollector.policy
   -verbosegc
   ```

   When you add the entries, take note of the following things:

   - Each entry must be on a single line.
   - Replace *server_name* with the actual Liberty server name. For example, `defaultServer`.
   - Replace *dc_home* with the actual data collector home directory. For example, `/opt/ibm/apm/agent/yndchome/7.3.0.14.08`.
3. Open the `server.env` file in the same directory and add the following path to the environment entry according to the operating system, where *dc_home* is the data collector home directory. If the `server.env` file does not exist, create it with a text editor.

   Table 230. Environment entry

Platform	Environment entry name	Environment entry value
AIX R6.1 (64-bit JVM)	LIBPATH	`/lib:dc_home/toolkit/lib/aix536`
AIX R7.1 (64 bit JVM)	LIBPATH	`/lib:dc_home/toolkit/lib/aix536`
Solaris 10 (64-bit JVM)	LIBPATH	`/lib:dc_home/toolkit/lib/sol296`
Solaris 11 (64-bit JVM)	LIBPATH	`/lib:dc_home/toolkit/lib/sol296`
Linux x86_64 R2.6 (64-bit JVM)	LD_LIBRARY_PATH	`/lib:dc_home/toolkit/lib/lx8266`
Linux Intel R2.6 (32-bit JVM)	LD_LIBRARY_PATH	`/lib:dc_home/toolkit/lib/li6263`
Windows (32-bit JVM)	PATH	`/lib;dc_home/toolkit/lib/win32`
Windows (64-bit JVM)	PATH	`/lib;dc_home/toolkit/lib/win64`

4. Open the `server.xml` file in the same directory and add the following lines to enable the monitoring feature:

   ```
   <featureManager>
                 <feature>webProfile-7.0</feature>
                 <feature>monitor-1.0</feature>
   ```

```
                <feature>usr:itcam-730.140</feature>
    </featureManager>
```

5. Restart the Liberty server.

Results

Now you can check the WebSphere Applications agent data in the Cloud APM console after you add this application component to your applications. For instructions on how to start the Cloud APM console, see "Starting the Cloud APM console" on page 957. For instructions on how to add or edit an application, see "Managing applications" on page 1073.

What to do next

After you manually configure the data collector, you cannot use the provided `unconfig` utility to unconfigure the data collector. Manually unconfigure the data collector instead. For instructions, see "Manually unconfigure the data collector" on page 155.

(Agent coexistence) Configuring the WebSphere Applications agent and the data collector

In the agent coexistence environment where both WebSphere Applications agent and ITCAM Agent for WebSphere Applications are installed, you must do some additional configuration for the agent and follow a different procedure to configure the data collector.

About this task

In the agent coexistence environment, you configure only one data collector to send data to both WebSphere Applications agent and ITCAM Agent for WebSphere Applications. Both agents must use different ports to listen to the requests from the data collector.

Procedure

1. If the data collector of ITCAM Agent for WebSphere Applications, which is installed in the old IBM® Tivoli® Monitoring infrastructure, exists in your environment, uninstall it.
2. Install the WebSphere Applications agent provided in IBM Cloud Application Performance Management 8.1.3 or later. It guarantees the data collector version 7.3.0.11.0 or later, which is supported for agent coexistence, is installed.
3. Make sure that the user ID that is to configure the data collector and the user ID that installed the application server have the appropriate user privileges required by the agent. For instructions, see "Checking user access requirements" on page 831.
4. Make sure that the WebSphere Applications agent and ITCAM Agent for WebSphere Applications are using different port numbers to listen to the requests from the data collector. The port numbers must be unique, which cannot be used by any other component in your environment. Configure the agent again to change the port, if necessary.
   - For information about how to configure the WebSphere Applications agent, see "Configuring WebSphere Applications agent" on page 854.
   - For information about how to configure the ITCAM Agent for WebSphere Applications, see ITCAM for Application Diagnostics or ITCAM for Applications documentation.
5. Use the provided configuration utility to configure the data collector. For instructions, see "Configuring the data collector for agent coexistence environment" on page 855.

   Tip: If you are familiar with the data collector configuration, you can also configure the data collector in silent mode. For instructions, see "Configuring the data collector in silent mode" on page 842.

Configuring WebSphere Applications agent

In the agent coexistence environment, the data collector is shared by both WebSphere Applications agent and ITCAM Agent for WebSphere Applications. Both agents must use different ports to listen to the requests from the data collector. You must configure the agent to change the port, if necessary.

About this task

- On Linux or AIX systems, you can configure the agent interactively by running the configuration script and then responding to prompts, or silently by creating a silent response file and running the configuration script without interaction.
- On Windows systems, you can configure the agent by creating a silent response file and running the configuration script, or with the provided Manage Monitoring Services utility. For information about how to start Manage Monitoring Services on Windows systems, see "Using the IBM Cloud Application Performance Management window on Windows systems" on page 182.

Procedure

- To configure the agent by editing the silent response file and running the script with no interaction, complete the following steps:

  a) Create a `.txt` file as the silent response file.

  b) Specify the following parameters in the silent response file. The syntax is *parameter_name=parameter_value*.

  configure_type
  Specifies the configuration type. This parameter is required.

  Valid value is `tema_configure` for agent configuration.

  KYN_ALT_NODEID
  Specifies the alternative node ID for identifying the agent. This parameter is required.

  Valid value is an alphanumeric string of up to 24 characters.

  KYN_PORT
  Specifies the listening port that is used by the agent. This is the TCP socket that the agent uses for listening to connection requests from the data collector. This parameter is required.

  The default value is 63335.

  Remember: In agent coexistence environment, make sure that the port number that you specified here is not being used by the ITCAM Agent for WebSphere Applications.

  For example, add the following lines in the `.txt` file that you created.

  ```
  configure_type=tema_configure
  KYN_ALT_NODEID=WASAgent
  KYN_PORT=63335
  ```

  c) Save and close the file, and then enter the following command to run the configuration script:

  – Linux AIX *install_dir*/bin/was-agent.sh config *path_to_responsefile*

  – Windows *install_dir*\bin\was-agent.bat config *path_to_responsefile*

  where *install_dir* is the agent installation directory. The default is `C:\IBM\APM` on Windows systems, `/opt/ibm/apm/agent` on Linux systems and AIX systems.

  d) After configuration completes, restart the WebSphere Applications agent if it is not running with the following command:

  – Linux AIX *install_dir*/bin/was-agent.sh start

  – Windows *install_dir*\bin\was-agent.bat start

- To configure the agent by running the script and responding to prompts, complete the following steps:

a) From the command line, go to the *`install_dir`*`/bin` directory, where *install_dir* is the agent installation directory.

The default is `/opt/ibm/apm/agent` on Linux systems and AIX systems.

b) Run the configuration script from the directory:

```
./was-agent.sh config
```

c) When prompted, enter 1 and press Enter to edit the settings for the monitoring agent of WebSphere Applications agent.

d) Press Enter until you are prompted for an alternative node ID to identify the monitoring agent.

e) Provide the node ID and press Enter. The valid format of the node ID is an alphanumeric string of up to 24 characters.

f) When prompted for the port number, provide the port that is used by the agent to listen to connection requests from the data collector and press Enter.

Remember: For agent coexistence environment, make sure that the specified port is not being used by the ITCAM Agent for WebSphere Applications.

g) After configuration completes, restart the WebSphere Applications agent.

Results

You have configured the WebSphere Applications agent.

What to do next

Next, you must configure the data collector. When you configure the data collector, you will be asked to provide the port number that you configured for the WebSphere Applications agent and ITCAM Agent for WebSphere Applications. For instructions, see "Configuring the data collector for agent coexistence environment" on page 855.

Configuring the data collector for agent coexistence environment

If you have both the WebSphere Applications agent and ITCAM Agent for WebSphere Applications in your environment, you can configure only one data collector for both agents.

Before you begin

Make sure that you have complete other steps that are documented in "(Agent coexistence) Configuring the WebSphere Applications agent and the data collector" on page 853.

About this task

Use the provided interactive configuration utility to configure the data collector for an environment where both WebSphere Applications agent and ITCAM Agent for WebSphere Applications exist and share the data collector.

Limitation: Integration the data collector with the following components or products are not supported for the ITCAM Agent for WebSphere Applications:

- ITCAM for Application Diagnostics Managing Server
- ITCAM for Transactions
- Tivoli Performance Viewer

Remember: Monitoring WebSphere Application Server Liberty is not supported by ITCAM Agent for WebSphere Applications. To monitor WebSphere Application Server Liberty, use the WebSphere Applications agent only. For information about data collector configuration for Liberty monitoring, see "Configuring the data collector interactively" on page 835 or "Configuring the data collector in silent mode" on page 842.

Procedure

1. Log in to the system with the user ID that is used to install the application server.
2. From the command line, go to the `bin` directory within the *dc_home* directory. The *dc_home* directory is as follows:

   - Windows *install_dir*`\dchome\7.3.0.14.08`
   - Linux AIX *install_dir*`/yndchome/7.3.0.14.08`
3. Run the following command to start the configuration utility:

   - Windows `config.bat`
   - Linux AIX `./config.sh`

   The configuration utility starts and displays the IP addresses of all network cards that are found on the local computer system.
4. Enter the number that corresponds to the IP address to use and press Enter.

   The configuration utility displays WebSphere Application Server home directories that are discovered on the system.
5. When prompted for the home directory of the application server, enter the number that corresponds to a WebSphere Application Server home directory or a full path to an application server home directory and press Enter.

   The configuration utility displays all application server profiles that are discovered under the specified home directory.
6. When prompted for the application server profile to be configured, enter the number that corresponds to the WebSphere Application Server profile and press Enter.

   The configuration utility indicates whether WebSphere Global Security is enabled for the WebSphere Application Server profile that you specified. If global security is not enabled, skip to Step "8" on page 856.
7. Specify whether to retrieve security settings from a client properties file. Enter 1 to allow the configuration utility to retrieve the user name and password from the appropriate client properties file. Otherwise, enter 2 to enter the WebSphere administrator user name and password.

   The data collector communicates with the WebSphere Administrative Services by using the Remote Method Invocation (RMI) or the SOAP protocol. If global security is enabled for a profile, you must specify the user ID and password of a user who is authorized to log in to the WebSphere Application Server administrative console for the application server profile. Alternatively, you can encrypt the user name and password and store them in client properties files before you configure the data collector. You must use the `sas.client.props` file for an RMI connection, or the `soap.client.props` file for an SOAP connection.
8. When you are prompted for the host name of WebSphere administrative console, press Enter to accept the default or specify the host name or IP address of the WebSphere administrative console. The default value is `localhost`.
9. When the configuration utility lists all the server instances that are not configured yet for data collection, select one or more application server instances from the list. Enter the number that corresponds to the application server instance to configure for data collection or enter an asterisk (*) to configure all application server instances for data collection, and then press Enter. To specify a subset of servers, enter the numbers that represent the servers, separated by commas.
   For example, `1,2,3`.

   Remember:

   - For a stand-alone environment, application server instances must be running during the configuration.
   - For a Network Deployment environment, the Deployment Manager must be running.
   - Ensure that the application server instances that you select are the actual servers that host the applications or services that you want to monitor.

The configuration utility provides an option for integrating the data collector for WebSphere Applications agent.

10. In the **Integration with Agent for WebSphere Applications** section, specify that you want to integrate the data collector with the monitoring agent. You must enter 1 to select this integration option, and then press Enter.

    The selected server will be registered for PMI resource monitoring.
11. Specify whether you want to configure the data collector within the application server instance. You must enter 1 for yes, and then press Enter.
12. Specify whether to enable the data collector for diagnostics data. Enter 1 for yes, or enter 2 for no.
13. When prompted for the host name of the V8 monitoring agent component, enter the host name or IP address of the WebSphere Applications agent or accept the default.
14. When prompted for the port number of the V8 monitoring agent, enter the port number that is used by the WebSphere Applications agent.

    Remember: The default value might not be appropriate to use if it is in use by another component. You must make sure that the specified port is not being used by any other component in your environment.
15. Specify that you want to configure the V6 monitoring agent. Enter 1 to configure the ITCAM Agent for WebSphere Applications and press Enter.
16. When prompted for the host name or IP address of the V6 monitoring agent, specify the host name or IP address of the ITCAM Agent for WebSphere Applications.
17. When prompted for the port number of the V6 monitoring agent, enter the port number that is used by the monitoring agent component of the ITCAM Agent for WebSphere Applications.

    Remember: The default value might not be appropriate to use if it is in use by another component. You must make sure that the specified port is not being used by any other component in your environment.
18. When prompted for the server alias, do not use the default value, and specify a unique server alias that you want to use. If you are configuring several application server instances, the configuration utility prompts you for an alias for each instance.

    Important: The alias can contain only the following characters: `A-Z`, `a-z`, underbar (_), dash (-), and period (`.`). Do not use other characters in the alias.

    The server alias is the first qualifier of the agent instance name (also known as MSN) that is displayed on the Cloud APM console. The default is the node name combined with the server name. For example, the **node1server1** alias indicates the server named **server1** in the node named **node1**.
19. When prompted for the TCP/IP port number for PMI resource monitoring, press Enter to accept the default or enter a new number. The default port is 63355.

    The port is used for internal communication between components that are running on the same host. If the default port is in use, set a different number.
20. In the **Support for transaction tracking** section, specify whether to enable transaction tracking. Enter 1 for yes, or enter 2 for no and skip to 22.

    Remember: To view transaction tracking information, you need to enable transaction tracking on the Agent Configuration page of the Cloud APM console.
21. Accept the default host name or IP address of the Transaction Framework Extension, which is an internal component of the WebSphere Applications agent that gathers metrics from the data collector.
22. Accept the default port number that the data collector uses to connect to the Transaction Framework Extension. The default is 5457.
23. Specify whether to integrate the data collector with Application Performance Diagnostics Lite. Press Enter to accept the default for no.
24. In the **Advanced settings** section, specify whether you want to change the garbage collection log path. Enter 1 to select a garbage collection log path. Otherwise, enter 2 and skip to step “26” on page 858.

25. Specify the garbage collection log path. Enter a file name with its full path.

    For example, if you specify `gc.log` as the file name, the actual name is set to *profile_name.cell_name.node_name.server_name*.gc.log for every configured application server instance.

    Important: In the garbage collection log path, you can use WebSphere variables such as ${SERVER_LOG_ROOT}. However, do not use templates, such as %pid.
26. In the **Data collector configuration summary** section, review the summary of the data collector configuration that is to be applied to the specified application server instances. If necessary, modify the configuration settings.
27. Enter a to accept the changes.
28. When prompted, specify whether you want to create a backup of your current configuration. Enter 1 to create a backup of the current configuration. Otherwise, enter 2.
29. Restart the application server instances as indicated by the configuration utility.
    a) Go to the `bin` directory under the home directory for the application server profile. For example, opt/IBM/WebSphere/AppServer/profiles/*profile_name*/bin.
    b) Stop the application server by entering the **stopServer** command in the command console.
    - Linux AIX ./stopServer.sh *server_name*
    - Windows stopServer.bat *server_name*

    c) When prompted, enter the user ID and password of WebSphere administrative console administrator.
    d) Start the application server again by entering the **startServer** command in the command console.
    - Linux AIX ./startServer.sh *server_name*
    - Windows startServer.bat *server_name*

    The data collector configuration takes effect after the application server restart.
30. Log in to the Cloud APM console to view data in the dashboards.
    a) Access the console by using the link that is provided in the email alerting you that your service is ready. Alternatively, access the console from the IBM Marketplace website. For detailed instructions, see "Starting the Cloud APM console" on page 957.
    b) Use the Applications editor to add the monitored application server to the Application Performance Dashboard. You can add it as a new component to your existing application or create an application to contain this component.

    For more information about the Applications editor, see "Managing applications" on page 1073.

What to do next

- If the current user ID that is used to configure the data collector is not the same ID of the user running the application server, verify that the user ID for configuring the data collector has read and write permissions to the `runtime` and `logs` directories within the data collector home directory. These two subdirectories are created by the ID of the user running the application server when the server is restarted.
- For WebSphere Applications agent, log in to the Cloud APM console to view the monitoring data in the dashboards. For ITCAM Agent for WebSphere Applications, log in to Tivoli Enterprise Portal to view data. If monitoring data is not available immediately, restart the monitoring agent by running the following commands:
  - Windows

```
cd install_dir\bin
was-agent.bat stop
was-agent.bat start
```

- Linux AIX

  ```
  cd install_dir/bin
  ./was-agent.sh stop
  ./was-agent.sh start
  ```

- Changing the server alias changes the agent instance name that is registered with the Cloud APM console. If this is not the first time that you configure the data collector and you changed the server alias, you must clear some cache files by completing the following steps:

  1. Stop the monitoring agent if it is running.
  2. Open the *hostname*`_yn.xml` file in the following directory with a text editor, where *hostname* is the name of the host where the WebSphere Applications agent or ITCAM Agent for WebSphere Applications is installed.
     - Windows *install_dir*`\TMAITM6_x64` (Default is `C:\IBM\APM\TMAITM6_x64` for WebSphere Applications agent or `C:\IBM\ITM\TMAITM6_x64` for ITCAM Agent for WebSphere Applications)
     - Linux AIX *install_dir*`/config` (Default is `/opt/ibm/apm/agent/config` for WebSphere Applications agent or `/opt/ibm/itm/agent/config` for ITCAM Agent for WebSphere Applications)
  3. Locate the line that starts with the following string and contains the previous server name. For example, `was85.win4net01Cell02.win4net01Node02.AppSrv01.server1`, where `server1` is previous name of the application server.

     ```
     <!ENTITY was_product_code.cellname.nodename.profilename.servername
     ```

     where *was_product_code* is the product code of WebSphere Application Server; *cellname* is the name of the cell; *nodename* is the node name; *profilename* is the name of the application server profile; *servername* is the previous name of the application server.
  4. Locate the .XML file that is indicated in the line within the current directory and delete the file.
  5. Remove the line that you located in Step 3 from the *hostname*`_yn.xml` file.
  6. At the end of the *hostname*`_yn.xml` file, remove the line that contains the previous server names.
  7. Save and close the file.
  8. Restart the monitoring agent.

Reconfiguring the data collector if you change the offering type on Cloud APM server
If you changed the offering type that you installed on the Cloud APM server from Cloud APM, Base to Cloud APM, Advanced and the WebSphere Applications agentwas installed and configured with the Cloud APM, Base offering, to use the agent advanced capabilities provided in the Cloud APM, Advanced offering, you must uninstall the previous WebSphere Applications agent and install the agent again with the Cloud APM, Advanced offering. Alternatively, you can reconfigure the data collector for the capabilities to be available in the new offering.

About this task

The WebSphere Applications agent is configured differently depending on which agent package is used to install the agent. After you change the offering type on the Cloud APM server, you have two choices to make the agent capabilities in the new offering available:

- Remove the agent that you installed with the previous offering and then install the agent in the new offering.
- Reconfigure the data collector again to use the capabilities in the new offering.

Procedure

- Remove the agent that you installed with the previous offering and then install the agent in the new offering.

   a) Unconfigure the data collector. For instructions, see “WebSphere Applications agent: Unconfiguring the data collector” on page 149.
   b) Uninstall the WebSphere Applications agent that you installed with the agent package of previous offering. For instructions, see “Uninstalling your agents” on page 147.
   c) Install the WebSphere Applications agent with the agent package in the new offering and configure the data collector again. For instructions, see “Configuring the data collector with the simple configuration utility” on page 832.

- Reconfigure the data collector again to use the capabilities in the new offering.
   a) Edit the `offering.id` file in the data collector home directory by changing the **`IOFFERING`** value to one of the following values, depending on your new offering type:

      BASE
         If your new offering type is Cloud APM, Base Private.

      ADVANCED
         If your new offering type is Cloud APM, Advanced Private.

      The data collector home directory is as follows:

      – Windows *`install_dir`*`\dchome\7.3.0.14.08`

      – Linux AIX *`install_dir`*`/yndchome/7.3.0.14.08`

   b) Reconfigure the data collector to enable diagnostics, transaction tracking, or both in the data collector based on what is supported in the new offering type. For instructions about how to configure the data collector, see “Configuring the data collector with the simple configuration utility” on page 832.
   c) Restart the WebSphere Application Server.
   d) From any page of the Cloud APM console, click **System Configuration** > **Agent Configuration** to open the **Agent Configuration** page. Ensure that the transaction tracking setting matches the capabilities available in your new offering type. If not, update the setting.

      The transaction tracking setting should be enabled for Cloud APM, Advanced, but be disabled for Cloud APM, Base.

Monitoring WebSphere Application Server Liberty inside a Docker container

To monitor a Liberty profile inside a Docker container, you must use the **`docker run`** command with a few options to configure the data collector before the WebSphere Application Server Liberty can be started.

Before you begin

You must install the WebSphere Applications agent on the Docker host.

About this task

Each Liberty profile running inside a Docker container requires a data collector to collect resource metrics, transaction metrics, and diagnostics data, and then transmit the data to the monitoring agent that is running on the Docker host. All data collectors that are configured on the same Docker host share the same monitoring agent on the host.

Procedure

To configure the data collector for a Liberty profile container, complete the following steps:

1. Create a `.txt` silent response file, specify the following configuration options in the file and save it.

   ```
   tema.host=agent_host
   was.appserver.server.name=liberty_profile_name
   ```

   where **`tema.host`** is used to specify the IP address of the monitoring agent host; **`was.appserver.server.name`** is used to specify the name of the Liberty profile.

Tip: A sample silent response file (`sample_silent_liberty_config.txt`) is provided in the *`<agent_install_dir>`*`/agent/yndchome/7.3.0.14.08/bin` directory. You can create your own response file based on this sample file.

2. Run the following command to launch the new Docker container. Note that you must accept the license to complete the configuration by setting the **`LICENSE`** parameter to `accept`.

   ```
   $docker run -d -e LICENSE=accept \
   -e JAVA_HOME=<java_home_dir> \
   -p <port_number>:<port_number> \
   -v <web_app_dir>:<liberty_install_dir>/usr/servers/<liberty_profile_name> \
   -v <agent_install_dir>/agent/yndchome:<agent_install_dir>/agent/yndchome
   websphere-liberty /bin/bash \
   -c "<agent_install_dir>/agent/yndchome/<dcversion>/bin/config.sh -silent
   <absolute_path_to_silent_response_file> && <liberty_install_dir>/bin/server
   run <liberty_profile_name>"
   ```

   where:

   - *<java_home_dir>* is the directory of the JRE that is used by the Liberty profile. For example, `/opt/ibm/java/jre`.
   - *<port_number>* is the port number that is used for communication between the container and the host.
   - *<web_app_dir>* is the directory where the web application locates.
   - *<liberty_install_dir>* is the installation directory of the WebSphere Application Server Liberty. Default is `/opt/ibm/wlp`.
   - *<liberty_profile_name>* is the name of the Liberty profile.
   - *<agent_install_dir>* is the installation directory of the WebSphere Applications agent. Default is `/opt/ibm/apm`.
   - *<dcversion>* is the version number of the data collector for WebSphere Applications agent. For example, 7.3.0.14.08.
   - *<absolute_path_to_silent_response_file>* is the absolute path to the silent response file that you created.

   For example, the following command configures the data collector for the Liberty profile named newitcam. Both the WebSphere Applications agent and the Liberty profile are installed in the default directories. The version of the monitoring agent and data collector is 7.3.0.14.08.

   ```
   $docker run -d -e LICENSE=accept \
   -e JAVA_HOME=/opt/ibm/java/jre \
   -p 9082:9082 \
   -v /home/kub/liberty-docker/newitcam:/opt/ibm/wlp/usr/servers/newitcam \
   -v /opt/ibm/apm/agent/yndchome:/opt/ibm/apm/agent/yndchome websphere-liberty
   /bin/bash \
   -c "/opt/ibm/apm/agent/yndchome/7.3.0.14.08/bin/config.sh -silent
   /opt/ibm/wlp/usr/servers/newitcam/silent_config.txt && /opt/ibm/wlp/bin/server
   run newitcam"
   ```

Results

Now you can verify that the WebSphere Applications agent data is displayed in the Cloud APM console. The **Cell name** column on the **WAS Information** widget shows the ID of the Docker container where the Liberty profile is running.

What to do next

To unconfigure the data collector interactively, use the following command to start the unconfiguration utility:

```
docker exec -i container_id "<agent_install_dir>/yndchome/7.3.0.14.08/bin
/unconfig.sh"
```

Manually configuring the data collector to monitor dynamic cluster servers

You can configure the data collector to monitor application server instances in a dynamic cluster by adding some data collector configuration parameters to the server template that was used to create the dynamic cluster server instances. This is an alternative method to configure dynamic cluster server instances to creating the server templates specific for the WebSphere Applications agent.

About this task

To configure the data collector for dynamic cluster monitoring, you must create two settings files, and then manually add settings in the WebSphere administrative console to modify the dynamic server template. The `runtime` directory is created automatically when the data collector is started for the application server instance. Note that any upgrade to the server template will erase these changes that you made in this way.

Important:

- The cluster name cannot contain a space.
- You must make manual changes to the WebSphere Application Server configuration for data collectors as the WebSphere administrative user.
- You must be an experienced WebSphere administrator to make manual changes to the WebSphere Application Server for data collection. Any error in the manual configuration change can result in the application server not starting.
- If you manually configure the data collector to monitor application server instances, you cannot use the unconfiguration utility to unconfigure the data collector. To unconfigure the data collector, you must manually change the settings back.

Procedure

1. Create the `dcManualInput.txt` file in the data collector `runtime` directory.

   Follow the instructions in "Creating the dcManualInput.txt file" on page 862.
2. Create the `itcam_wsBundleMetaData.xml` file in the data collector `wsBundleMetaData` directory. Follow the instructions in "Creating the itcam_wsBundleMetaData.xml file" on page 865.
3. Use WebSphere administrative console to modify the dynamic server template by adding the data collector configuration parameter. Follow the instructions in "Adding settings with the WebSphere administrative console" on page 865.

   Tip: The dynamic cluster member name is used as the middle qualifier of the WebSphere Applications agent instance name that is displayed on the Cloud APM console. Sometimes, the cluster member name might be truncated due to the length limit on the agent instance name. In this case, you can modify the dynamic server template by adding a variable named *${MEP_NAME}* and setting the value to the JVM name for each server instance. Then, you can distinguish each cluster member by the actual JVM name on the Cloud APM console. For instructions, see "Optional: Showing actual JVM name to distinguish cluster members" on page 868.

Creating the `dcManualInput.txt` file

About this task

The `dcManualInput.txt` file contains some values that are required for initial configuration of the data collector.

Procedure

To create the `dcManualInput.txt` file, complete the following steps:

1. Check whether a file named *`platform`*`_Template.DCManualInput.txt` exists in the following directory. If it does not exist, create it.

   - Linux AIX *`install_dir`*`/yndchome/7.3.0.14.08/runtime`

- Windows *install_dir*`\dchome\7.3.0.14.08\runtime`

The *platform* variable in the file name indicates the operating system architecture, for example, aix32, xLinux64.

You can name the file anything that you want. However, *platform*`_Template.DCManualInput.txt` follows the standard naming convention when the file is created by running the `configtemplate.sh` script. You will need to specify this file for the server template with WebSphere administrative console in the subsequent step.

2. Copy the contents of the following file to the `.txt` file that you found or created in the previous step.

   - Linux AIX *dc_home*`/itcamdc/etc/was/dcInput_manual.properties`
   - Windows *dc_home*`\itcamdc\etc\was\dcInput_manual.properties`

3. Edit the contents of the `.txt` file. You must set the parameters in section 1 of the file according to the descriptions provided in Table 231 on page 863.

   Remember:

   - Do not change the parameters in section 2.
   - Some of the configuration parameters that are used by the data collector to create runtime directories are always set to none. It is because in dynamic cluster monitoring, the data collector uses the WebSphere server instance configuration to create the directories when JVM starts.

Table 231. Configuration Parameters for Section 1

Parameter	Value
local.hostname	The IP address or fully qualified domain name of the local system.
was.profile.home	The profile home directory. Always set it to none for dynamic cluster server
was.version	A short version number. Always set it to none for dynamic cluster server
itcam.home	The data collector home directory. Example: `/opt/ibm/apm/agent/yndchome/7.3.0.14.08`
was.nodename	Node name. Always set it to none for dynamic cluster server
was.servername	Server name. Always set it to none for dynamic cluster server
was.profilename	WebSphere profile name. Always set it to none for dynamic cluster server

Table 231. Configuration Parameters for Section 1 (continued)

Parameter	Value
am.camtoolkit.gpe.dc.operation.mode	Operation mode of the data collector. Valid values are any combination of `WR`, `TT`, and `DE`, where: **WR** Integrates the data collector with the WebSphere Applications agent. **TT** Integrates the data collector with ITCAM for Transactions. **DE** Integrates the data collector with ITCAM Diagnostics Tool. The tool is previewed in the ITCAM for Application Diagnostics beta. You must specify only the operation modes required. For example, if you are connecting the data collector to the WebSphere Applications agent only, specify `WR`. Separate multiple operation modes with a comma. Example: `am.camtoolkit.gpe.dc.operation.mode=WR, DE`
interp	Platform code. Example: `interp=win64` or `interp=lx6266`
kwj.serveralias	WebSphere Application Server alias name. Always set it to none for dynamic cluster server
temagclog.path	(Optional) Garbage Collection log file path name. Enter a unique file name with full path. The path name must not include spaces.
tema.host	Host name or IP address of the WebSphere Applications agent. Mandatory if the operation mode includes WebSphere Applications agent (`WR`). Usually, the monitoring agent is installed on each system where the data collector is running and the loopback address can be specified. Example: `tema.host=127.0.0.1`
tema.port	Port to use for communicating with the WebSphere Applications agent. Mandatory if the operation mode includes WebSphere Applications agent (`WR`). The default value is 63335. Example: `tema.port=63335`
tt.connection.string	Host name or IP address and the port number of the transaction collector component of ITCAM for Transactions in the format of `tcp:`*`host_name`*`(IP):`*`port`*. Mandatory if the operation mode includes ITCAM for Transactions (TT). Example: `tt.connection.string=192.38.234.77:5455`

4. Add the following lines to the section 1 of the `.txt` file.

```
bcm.helper=com.ibm.tivoli.itcam.was.bcm.websphere.DefaultWASBCMHelper
BCM_HELPER=@{bcm.helper}
```

5. Save and close the file.

Creating the `itcam_wsBundleMetaData.xml` file

About this task

The `itcam_wsBundleMetaData.xml` file contains some of the values that are required initial configuration of the data collector.

Procedure

To create this file, complete the following steps:

1. Create a directory that is named `wsBundleMetaData` under the following directory:
   - Linux AIX *install_dir*/yndchome/7.3.0.14.08/runtime
   - Windows *install_dir*\dchome\7.3.0.14.08\runtime
2. Create a file that is named `itcam_wsBundleMetaData.xml` and copy the contents of the following file into it:
   - Linux AIX *install_dir*/yndchome/7.3.0.14.08/itcamdc/etc/was/itcam_wsBundleMetaData_template.xml
   - Windows *install_dir*\dchome\7.3.0.14.08\itcamdc\etc\was\itcam_wsBundleMetaData_template.xml
3. In the `itcam_wsBundleMetaData.xml` file, replace the *@{CONFIGHOME}* variable with the full path to your data collector home directory.

   The data collector home directory on each operating system is as follows:
   - Linux AIX *install_dir*/yndchome/7.3.0.14.08
   - Windows *install_dir*\dchome\7.3.0.14.08
4. Place the `itcam_wsBundleMetaData.xml` file to the `wsBundleMetaData` directory that you created in Step 1.

Adding settings with the WebSphere administrative console

Procedure

Complete the following steps to modify the dynamic server template with the WebSphere administrative console.

1. Log in to the WebSphere Administrative Console.
2. Click **Servers**.
3. Expand **Clusters** and select **Dynamic Clusters**.
4. Click the name of the dynamic server cluster that you want to configure with the data collector.
5. Under the **Additional Properties** section, click **Server template**.
6. Under the **Server Infrastructure** section, expand **Java and Process Management** and click **Process Definition**.
7. Under the **Additional Properties** section, click **Java Virtual Machine**.
8. In the **Generic JVM arguments** field, add the following entries.

```
-agentlib:am_$jvm-vendor_$jvm-version=${WAS_SERVER_NAME}
-Xbootclasspath/p:${ITCAMDCHOME}/toolkit/lib/bcm-bootstrap.jar
-Djava.security.policy=${ITCAMDCHOME}/itcamdc/etc/datacollector.policy
```

```
-verbosegc -Dcom.ibm.tivoli.itcam.ai.runtimebuilder.inputs=${ITCAMDCHOME}/runtime/
$platform_Template_DCManualInput.txt
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
-Dsun.rmi.transport.connectionTimeout=300000
-Dws.bundle.metadata=${ITCAMDCHOME}/runtime/wsBundleMetaData
–Ditcamdc.dyncluster=true
```

When you are adding the entries, take note of the following things:

- All entries must be on a single line.
- Separate different arguments by spaces before the `-` sign, do not use spaces anywhere else.
- Replace the following variables with the actual names:
  - *$jvm-vendor*: The JVM vendor that is used.
  - *$jvm-version*: The JVM version information, such as 15 based on Java 5, 16 based on Java 6, or 17 based on 7.
  - *$platform_Template_DCManualInput.txt*: The `.txt` file that you created in the previous step.

  Example:

  ```
  -agentlib:am_ibm_16=${WAS_SERVER_NAME}
  -Xbootclasspath/p:${ITCAMDCHOME}/toolkit/lib/bcm-bootstrap.jar
  -Djava.security.policy=${ITCAMDCHOME}/itcamdc/etc/datacollector.policy
  -verbosegc -Dcom.ibm.tivoli.itcam.ai.runtimebuilder.inputs=${ITCAMDCHOME}/runtime/
  aix64_Template_DCManualInput.txt
  -Dsun.rmi.dgc.client.gcInterval=3600000
  -Dsun.rmi.dgc.server.gcInterval=3600000
  -Dsun.rmi.transport.connectionTimeout=300000
  -Dws.bundle.metadata=${ITCAMDCHOME}/runtime/wsBundleMetaData
  –Ditcamdc.dyncluster=true
  ```

9. Click **Apply**.
10. In the Messages dialog box, click **Save**.
11. In the Save to Master Configuration dialog box, complete the following steps:
    - If you are under a Network Deployment environment, ensure that **Synchronize changes with Nodes** is selected and then click **Save**.
    - If you are not under a Network Deployment environment, click **Save**.
12. Go back to expand **Clusters**, click **Dynamic clusters**, and click the same server name.
13. In the **Configuration** tab, go to **Server Infrastructure** > **Java and Process Management** > **Process Definition** > **Environment Entries**.
14. Depending on the operating system, the hardware platform, and the application server JVM, set the following environment entry:

Table 232. Environment entry

Platform	Environment Entry name	Environment Entry value
AIX R6.1 (32-bit JVM)	`LIBPATH`	`/lib:${ITCAMDCHOME}/toolkit/lib/aix533:${ITCAMDCHOME}/toolkit/lib/aix533/ttapi`
AIX R6.1 (64-bit JVM)	`LIBPATH`	`/lib:${ITCAMDCHOME}/toolkit/lib/aix536:${ITCAMDCHOME}/toolkit/lib/aix536/ttapi`
AIX R7.1 (32-bit JVM)	`LIBPATH`	`/lib:${ITCAMDCHOME}/toolkit/lib/aix533:${ITCAMDCHOME}/toolkit/lib/aix533/ttapi`

Table 232. Environment entry (continued)

Platform	**Environment Entry name**	**Environment Entry value**
AIX R7.1 (64-bit JVM)	`LIBPATH`	`/lib:${ITCAMDCHOME}/toolkit/lib/aix536:${ITCAMDCHOME}/toolkit/lib/aix536/ttapi`
Linux x86_64 R2.6 (64-bit JVM)	`LD_LIBRARY_PATH`	`/lib:${ITCAMDCHOME}/toolkit/lib/lx8266:${ITCAMDCHOME}/toolkit/lib/lx8266/ttapi`
Linux Intel R2.6 (32-bit JVM)	`LD_LIBRARY_PATH`	`/lib:${ITCAMDCHOME}/toolkit/lib/lx6263:${ITCAMDCHOME}/toolkit/lib/lx6263/ttapi`
Linux ppc R2.6 (32-bit JVM)	`LD_LIBRARY_PATH`	`/lib:${ITCAMDCHOME}/toolkit/lib/lpp263:${ITCAMDCHOME}/toolkit/lib/lpp263/ttapi`
Linux ppc R2.6 (64-bit JVM)	`LD_LIBRARY_PATH`	`/lib:${ITCAMDCHOME}/toolkit/lib/lpp266:${ITCAMDCHOME}/toolkit/lib/lpp266/ttapi`
Windows (32-bit JVM)	`PATH`	`/lib;${ITCAMDCHOME}/toolkit/lib/win32;${ITCAMDCHOME}/toolkit/lib/win32/ttapi`
Windows (64-bit JVM)	`PATH`	`/lib;${ITCAMDCHOME}/toolkit/lib/win64;${ITCAMDCHOME}/toolkit/lib/win64/ttapi`

15. Set the NLSPATH environment entry name to the following value:

    ```
    ${ITCAMDCHOME}/toolkit/msg/%L/%N.cat
    ```

16. Click **Apply** and click **Save**.
17. In the Save to Master Configuration dialog box, complete the following steps:
    - If you are under a Network Deployment environment, ensure that **Synchronize changes with Nodes** is selected and then click **Save**.
    - If you are not under a Network Deployment environment, click **Save**.
18. Go back to expand **Clusters**, click **Dynamic clusters**, and then click the same server name.
19. In the **Configuration** tab, go to **Server Infrastructure** > **Java and Process Management** > **Process Definition** > **Java Virtual Machine** > **Additional Properties: Custom Properties**.
20. Click **New** to add the following name and value pairs, and then click **Apply**.
    - Create an `am.home` property and set its value to the following directory:
      - Linux AIX `*install_dir*/yndchome/7.3.0.14.08/itcamdc`
      - Windows `*install_dir*\dchome\7.3.0.14.08\itcamdc`
    - Create an `am.orig.wascell` property and set its value to the cell directory. For example, `am.orig.wascell =cellname1`.

- Create a `com.ibm.tivoli.itcam.toolkit.ai.runtimebuilder.enable.rebuild` property and set its value to `true`.
- Create an `ITCAM_DC_ENABLED` property and set its value to `true`.
- Create a `TEMAGCCollector.gclog.path` property. If the generic JVM `verlogsegclog` argument is set, set the value of the `TEMAGCCollector.gclog.path` property to the same value. Otherwise, set the `TEMAGCCollector.gclog.path` property to None.

  Tip: To identify the value of the `verlogsegclog` property, in the **Configuration** tab, click **Server Infrastructure** > **Java and Process Management** > **Process Definition** > **Java Virtual Machine**. The `verlogsegclog` value is in the **Generic JVM arguments** field.

21. In the Messages dialog box, click **Save**.
22. In the Save to Master Configuration dialog box, complete the following steps:
    - If you are under a Network Deployment environment, ensure that **Synchronize changes with Nodes** is selected. Click **Save**.
    - If you are not under a Network Deployment environment, click **Save**.
23. In the Navigation Pane, click **Environment** > **WebSphere Variables**.
24. Set the following variables. For each variable, you must choose the appropriate scope level, depending on the data collector installation directory on various systems. If different systems have different installation directories for the data collector, these variables must be set correctly for each node-level scope. If they all have the same installation directory, the scope can be higher, such as at the cluster level.
    - Set *ITCAMDCHOME* to following directory:
      - Linux AIX *install_dir*/yndchome/7.3.0.14.08/itcamdc
      - Windows *install_dir*\dchome\7.3.0.14.08\itcamdc
    - Set *ITCAMDCVERSION* to the version of the data collector, for example, 7.3.0.14.08.
25. Click **Apply** and click **Save**.
26. In the Save to Master Configuration dialog box, complete the following steps:
    - If you are under a Network Deployment environment, ensure that **Synchronize changes with Nodes** is selected and then click **Save**.
    - If you are not under a Network Deployment environment, click **Save**.

    After the template is modified, the values are synchronized with all the server instances in the dynamic cluster. Any new server that is dynamically created will also have the same data collector configuration parameters.
27. Restart the application server instance for the data collector to be activated. The data collector reads the settings files and creates the runtime directory.

Optional: Showing actual JVM name to distinguish cluster members

About this task

On the Cloud APM console, the WebSphere Applications agent instance name takes the form of *host_name*::*was_server_name*:KYNS and has a maximum length of 32 characters. In the dynamic cluster environment, dynamic cluster member names are used for the middle qualifier *was_server_name*.

Sometimes, the cluster member names are truncated due to the character length limit. In this case, you can specify the actual JVM name to be used for the middle qualifier in the agent instance name.

Procedure

Perform the following steps to show actual JVM name in the agent instance name:

1. Log in to the WebSphere Administrative Console to update the generic JVM arguments by adding a new environment variable *${MEP_NAME}* as follows:

```
-agentlib:am_$jvm-vendor_$jvm-version=${MEP_NAME}${WAS_SERVER_NAME}
-Xbootclasspath/p:${ITCAMDCHOME}/toolkit/lib/bcm-bootstrap.jar
-Djava.security.policy=${ITCAMDCHOME}/itcamdc/etc/datacollector.policy
-verbosegc -Dcom.ibm.tivoli.itcam.ai.runtimebuilder.inputs=${ITCAMDCHOME}/runtime/
$platform_Template_DCManualInput.txt
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
-Dsun.rmi.transport.connectionTimeout=300000
-Dws.bundle.metadata=${ITCAMDCHOME}/runtime/wsBundleMetaData
–Ditcamdc.dyncluster=true
```

Example:

```
-agentlib:am_ibm_16=${MEP_NAME}${WAS_SERVER_NAME}
-Xbootclasspath/p:${ITCAMDCHOME}/toolkit/lib/bcm-bootstrap.jar
-Djava.security.policy=${ITCAMDCHOME}/itcamdc/etc/datacollector.policy
-verbosegc
-Dcom.ibm.tivoli.itcam.ai.runtimebuilder.inputs=${ITCAMDCHOME}/runtime/
aix64_Template_DCManualInput.txt
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
-Dsun.rmi.transport.connectionTimeout=300000
-Dws.bundle.metadata=${ITCAMDCHOME}/runtime/wsBundleMetaData
–Ditcamdc.dyncluster=true
```

2. Save and apply the changes.
3. In the Navigation Pane, click **Environment** > **WebSphere Variables** to define the *${MEP_NAME}* variable for each dynamic cluster member. Set the value to the actual JVM name of the cluster member.
4. Save and apply the changes.
5. Restart the application server instance.

   On the Cloud APM console, a new WebSphere Applications agent instance whose name contains the *${MEP_NAME}* value that you just specified is displayed.

Dynamically configuring data collection on Agent Configuration page

After you enable the support for transaction tracking or diagnostic data collection in the data collector, use the **Agent Configuration** page to dynamically enable or disable the data collection.

Before you begin

- You must install and configure the Monitoring Agent for WebSphere Applications.
- To enable or disable transaction tracking for the monitored application servers, you must install Transaction Tracking. You must also enable support for transaction tracking in the agent as described in "Configuring the data collector interactively" on page 835. If you follow the simple configuration procedure, the data collector is automatically configured with support for transaction tracking.
- To enable or disable the collection of diagnostic data including method trace, you must have Cloud APM, Advanced. You must also enable support for the collection of diagnostic and method trace information in the data collector as described in "Configuring the data collector interactively" on page 835. (Not available for Cloud APM, Base).

Tip: The **Agent Configuration** page displays all the servers that are monitored by the agent. If any server is missing, it might not be correctly monitored. Check the agent log files on the monitored system for error messages, for example, connection errors.

Remember: The WebSphere Applications agent supports only Db2 and Oracle as the data source. For other types of data sources, some KPI values might appear to be null on the Transaction Tracking dashboards and group widgets.

Procedure

Complete the following steps to configure data collection for each server:

1. From the navigation bar, click **System Configuration** > **Agent Configuration**.

   The **Agent Configuration** page is displayed.

2. Click the **WebSphere Applications** tab.
3. Select the check boxes of the servers on which you want to configure data collection and take one of the following actions from the **Actions** list:

   - To enable transaction tracking, click **Enable Transaction Tracking**. The status in the **Current Transaction Tracking** column is updated to Yes for each selected server.
   - To enable only diagnostic data collection, click **Enable Diagnostic Mode**. The status in the **Current Diagnostic Mode** column is updated to Yes for each selected server.
   - To collect both the diagnostic data and method trace information, click **Enable Diagnostic Mode and Method Trace**. The status in the **Current Diagnostic Mode** and **Current Method Trace** columns is updated to Yes for each selected server.
   - To disable transaction tracking for the selected server, click **Disable Transaction Tracking**. The status in the **Current Transaction Tracking** column is updated to No for each selected server.
   - If only the diagnostic data collection is enabled for the selected server, to disable data collection, click **Disable Diagnostic Mode**. The status in the **Current Diagnostic Mode** column is updated to No for each selected server.
   - If both the diagnostic data and method trace data are enabled for the selected server, to disable data collection, click **Disable Diagnostic Mode and Method Trace**. The status in the **Current Diagnostic Mode** and **Current Method Trace** columns is updated to No for each selected server.

   Remember:

   - Unless support for transaction tracking or diagnostic data collection is configured in the data collector, operations on the **Agent Configuration** page does not enable data collection and the column value is set to No.
   - If the application server profile was configured to use `127.0.0.1` as the host name, the **IP Address** column on the **Agent Configuration** page displays the IP address of the host where the WebSphere Applications agent is installed and running.

Results
You have configured data collection for each selected server. Transaction tracking data and diagnostic data can be displayed in the topology dashboards after you enable the data collection.

Important: If an application server restarts, you might need to enable transaction tracking or diagnostic data collection for the server again.

Enabling memory leak monitoring

For the Memory Analysis dashboard to contain data, you must enable memory leak monitoring for the data collector. If the JRE that is used by the application server is supported, the memory leak monitoring function is enabled by default after you enable diagnostic data collection.

Before you begin

- Make sure that `-Xtrace:`none is not defined in the JVM arguments for the application server.
- When memory leak monitoring is enabled, the following settings are defined in the JVM arguments for the application server. If you have defined these settings in your current JVM arguments, make sure that it is OK for the data collector configuration to change them.

```
-Xhealthcenter:level=inprocess
-Xgc:allocationSamplingGranularity=10000
```

- Make sure the JRE that is used by the application server is one of the following versions:
  - IBM JRE 1.6.0 SR16 FP3 or later
  - IBM JRE 1.6.1 SR8 FP3 or later
  - IBM JRE 1.7.0 SR8 FP10 or later
  - IBM JRE 1.7.1 SR2 FP10 or later

- – IBM JRE 1.8 or later
- – Other IBM JRE later than 1.6.0 SR7 with iFix for APAR IV67574

About this task

The memory leak monitoring function requires the IBM Health Center component of IBM JRE. You must make sure the JRE that is used by the application server is supported by this function.

- On AIX or Linux systems, when you configure the data collector to enable diagnostic data collection, if the current JRE is supported, the configuration utility automatically checks whether the Health Center component is eligible and upgrade the Health Center if it is not.
- On Windows systems, you must manually upgrade the Health Center component if the current version is not supported because the configuration utility cannot replace files for a running JRE.

Remember: The following procedure is required on Windows systems only. For AIX or Linux systems, to enable memory leak monitoring, you only need to make sure that the JRE version is supported and the diagnostic data collection is enabled. For Solaris systems, Health Center of IBM JRE is not supported, so memory leak monitoring cannot be enabled on Solaris systems.

Procedure

1. Check the IBM Health Center version that is included in the JRE used by the application server.
   a) At the command prompt, change to the `bin` directory within the JRE home directory.
   b) Type `java -Xhealthcenter -version` and press Enter.

   The command returns the JRE version and the IBM Health Center version. The memory leak monitoring function requires IBM Health Center 3.0.11 or later.
2. If the IBM Health Center version is not eligible, upgrade the JRE to a version that contains IBM Health Center 3.0.11 or later.
3. Run the data collector configuration or reconfiguration utility to enable the diagnostic data collection.
   - If you have not configured the data collector, use the `simpleconfig` or `config`.
   - If you have configured the data collector, use the `reconfig` utility.

   Remember: If you have enabled diagnostic data collection before you upgrade the JRE, you still need to run the data collection configuration utility again.

Configuring the PMI

To view performance data in the operational monitoring dashboards, the Performance Monitoring Infrastructure (PMI) on the WebSphere Application Server must be configured to gather performance data.

About this task

Use the WebSphere Administrative Console to enable the PMI and set the PMI level on the WebSphere Application Server.

The PMI provides four predefined levels:

1. None
2. Basic
3. Extended
4. All

You can use a custom option to selectively enable or disable individual statistics. Each level includes the statistics from the level below it.

To display data in the operational monitoring dashboards, the attributes that are used in the dashboard calculations must be included in selected level.

By default, the WebSphere Applications agent sets the PMI level high enough to collect the required attributes.

Restriction: To see data in some of the Process Server and Transaction Manager group widgets, you must manually set the PMI level. For more information, see the fly-over help on the group widgets.

If you modify the PMI level with the WebSphere Administrative Console, you must verify that the level is high enough to collect the required data.

Procedure

- To enable the PMI on the application server, complete these steps:
  a) In the WebSphere Administrative Console, expand **Monitoring and Tuning**, and then select **Performance Monitoring Infrastructure (PMI)**.
  b) From the list of servers, click the name of your server.
  c) Click the Configuration tab, and then select the **Enable Performance Monitoring Infrastructure (PMI)** check box.
  d) Click **Apply** or **OK**.
  e) Click **Save** to enable the PMI.
- To set the PMI level on the application server, complete these steps:
  a) In the WebSphere administrative console, expand **Monitoring and Tuning**, and then select **Performance Monitoring Infrastructure (PMI)**.
  b) From the list of servers, click the name of your server.
  c) Click the Configuration tab, and then select the statistics set to use; Basic, Extended, All, or Custom.
  d) Click **Apply** or **OK**.
  e) Click **Save** to set the PMI level.

  For information about the PMI level that is required for each attribute, see the "Dashboard attributes" section in the WebSphere Applications agent Reference. The monitoring overhead that is incurred when you turn on the collection of each attribute is displayed.

Restoring the application server configuration from a backup

If you configured a stand-alone application server instance for data collection either manually or with the configuration or migration utility and the application server fails to start, you must restore the application server configuration from a backup. If you did not create a backup, contact IBM Support.

About this task

In a Network Deployment environment, if you configured an application server instance for data collection manually or with the configuration or migration utility and the application server fails to start, you have the following options:

- You can restore the application server configuration from a backup configuration. If you did not create a backup, contact IBM Support.
- You can manually unconfigure the data collector. The Deployment Manager and the Node Agent on the application server must be running. For more information, see “Manually removing data collector configuration from an application server instance” on page 152.

This section applies only to the Windows, UNIX, and Linux operating systems.

Procedure

To apply the backup configuration by using the **`restoreConfig`** command, use one of the following procedures:

- In a non-Network Deployment environment, complete the following steps:

   a) Locate your backup configuration file.

      The default directory is *dc_home*/data. If several backup files are present, check the modification date and time of the file. It must be the date and time of the failed configuration. If you did not complete any other data collector configurations on the same host after the failed one, use the most recent file in the directory.

   b) Stop all instances of the application server.

   c) Run the **restoreConfig** command from the *appserver_home*/profiles/*profile_name*/bin directory.

      The command syntax is as follows:

      – Windows restoreConfig.bat *full_path_to_backup_file*

      – Linux AIX ./restoreConfig.sh *full_path_to_backup_file*

      For more information about the arguments of the **restoreConfig** command, see the topic in WebSphere Application Server Knowledge Center.

   d) Start the instances of the application server again.

- In a Network Deployment environment, complete the following steps:

   a) Locate your backup configuration file.

      The default directory is *dc_home*/data. If several backup files are present, check the modification date and time of the file; it must be the date and time of the failed configuration. If you did not complete any other data collector configurations on the same host after the failed one, use the most recent file in the directory.

   b) Stop all instances of the application server.

   c) Create a temporary directory in any convenient path (*temp_directory*). On a UNIX or Linux system, create it under the /tmp directory.

   d) Run the restoreConfig command from the *appserver_home*/profiles/*profile_name*/bin directory.

      The command syntax is as follows:

      – Windows restoreConfig.bat *full_path_to_backup_file*

      – Linux AIX ./restoreConfig.sh *full_path_to_backup_file*

      The **restoreConfig** command restores the original application server configuration to the temporary directory.

   e) Copy the server.xml, variables.xml, and pmi-config.xml files from temporary directory to the Deployment Manager system.

      – Source directory: *temp_directory*/*restored_configuration_home*/cells/*cell_name*/nodes/*node_name*/servers/*server_name*

      – Target directory: *appserver_home*/profiles/*profile_name*/config/cells/*cell_name*/nodes/*node_name*/servers/*server_name*

   f) Complete a node sync from the Deployment Manager administrative console for the node.

   g) In the Deployment Manager administrative console, save changes to the master configuration.

   h) Start the instances of the application server.

Advanced data collector configuration

You can modify data collector configuration files to change additional monitoring settings.

Properties files for Liberty data collector

Various configuration files are provided for you to further control the data collector configuration and behavior.

After you extract the data collector package to a local directory, the data collector files are located in the *local_dir*/liberty_dc/.gdc/7.3.0.14.08 directory. For example, /opt/ibm/apm/.gdc/7.3.0.14.08. This folder becomes the home directory of the data collector, which is referred to as *dc_home* in the following statements for simplification.

Data collector properties files

Each application server instance that is monitored by the data collector has its own properties file. The data collector automatically creates the properties file. The name of the file is *dc_home*/runtime/*appserver_version.node_name.profile_name.server_name*/datacollector.properties.

To facilitate future upgrades, do not change this file.

Instead, add the settings that you want to modify to the data collector custom properties file. This file is named *dc_home*/runtime/*app_server_version.node_name.profile_name.server_name*/custom/datacollector_custom.properties. Settings in the data collector custom properties file override the values that are in the data collector properties file.

Important: If the *dc_home*/runtime/*app_server_version.node_name.profile_name.server_name*/custom/datacollector_custom.properties file does not exist, create it when you want to make changes. You might also have to create the custom directory.

Toolkit properties file

The toolkit properties file is automatically created by the data collector at startup, using various input files. It is unique for every application server instance monitored by the data collector. Its name is *dc_home*/runtime/*appserver_version.node_name.profile_name.server_name*/toolkit.properties.

Because this file is re-created at each data collector startup, **do not make any changes** to this file; if you do, they will be overwritten.

Instead, add the settings that you want to modify to the toolkit custom properties file. This file is named *dc_home*/runtime/*app_server_version.node_name.profile_name.server_name*/custom/toolkit_custom.properties. Settings in the toolkit custom properties file override the values in the toolkit properties file.

You can also set toolkit properties for all the application server instances that are monitored by this installation of the data collector. To do this, add the settings to the global toolkit custom properties file: *dc_home*/runtime/custom/toolkit_global_custom.properties. However, if a property is set in the instance-specific toolkit_custom.properties file, it overrides the value in the global file for this instance.

Important: If the *dc_home*/runtime/*app_server_version.node_name.profile_name.server_name*/custom/toolkit_custom.properties or *dc_home*/runtime/custom/toolkit_custom.properties file does not exist, create it when you want to make changes. You might also have to create the custom directory.

Other properties files

Besides the data collector properties file and toolkit properties file, there are other properties files that are unique for every application server instance monitored by the data collector.

***dc_home*/runtime/*appserver_version.node_name.profile_name.server_name*/custom/gdc/gdc_custom.properties**

Defines the details for collecting diagnostic and method trace data. For information about changing this file, see "Configuring collection of detailed diagnostic information" on page 877.

***dc_home*/runtime/*appserver_version.node_name.server_name*/hc.properties**

Defines the details for heap snapshot collection and memory allocation collection. For information about changing this file, see "Configuring collection of detailed diagnostic information" on page 877.

***dc_home*/runtime/*app_server_version.node_name.profile_name.server_name*/cynlogging.properties**

Defines the log file names and logging details for the Java portion of the data collector.

***dc_home*/runtime/*app_server_version.node_name.profile_name.server_name*/cyn-cclog.properties**

Defines the log file names and logging details for the C++ portion of the data collector.

Data collector trace files

The data collector trace files are stored by default in the following locations:

- Windows *dc_home*\logs\CYN\logs.
- Linux AIX *dc_home*/logs/CYN/logs.

Enabling or disabling transaction tracking and diagnostic data collection

By default, transaction tracking and method tracing are enabled for the data collector. Heap snapshot collection and memory allocation collection are disabled. You can customize the data collection or the intervals at which the diagnostic data is collected by editing the `.properties` files of the data collector.

About this task

The properties files of the data collector are in the *dc_home* directory, for example, `/opt/ibm/apm/.gdc/7.3.0.14.08`. Use different properties to customize the data collector for the following purposes:

- Enable or disable transaction tracking.
- Enable or disable heap snapshot collection.
- Specify the interval at which the data collector takes snapshot of heap dump.
- Enable or disable memory allocation monitoring.
- Specify the interval at which the data collector collects memory allocation information.
- Enable or disable method tracing.

Remember: Depending on whether you have restarted the Liberty server after data collector configuration, different `.properties` files are applicable. If you have restarted the Liberty server after data collector configuration, a `runtime` directory is created within the *dc_home* directory. After that, you can only use the `.properties` files within the *dc_home*/runtime/*appserver_version.node_name.profile_name.server_name* directory to customize data collector for each application server.

Procedure

- To enable or disable transaction tracking, set the **com.ibm.tivoli.itcam.dc.bluemix.transaction.enabled** property in the following file to `true` or `false`:

  dc_home/runtime/*appserver_version.node_name.server_name*/ldc.properties (if the `runtime` directory does not exist, use *dc_home*/ldc/etc/ldc.properties)

After transaction tracking is enabled, you can monitor IBM Java application stack in the topologies.

- To enable or disable heap snapshot collection, set the **com.ibm.tivoli.itcam.hc.send.heap.enable** and **com.ibm.tivoli.itcam.hc.snapshot.automatic.enable** properties in the following file to true or false.

  dc_home/runtime/*appserver_version.node_name.server_name*/hc.properties (if the runtime directory does not exist, use *dc_home*/healthcenter/etc/hc.properties)

  After heap snapshot collection is enabled, the data collector can take heap snapshot at specified intervals. Heap dump information can be displayed in the Heap Dump dashboard.
- To change the interval at which heap snapshot is taken by the data collector, set the **com.ibm.tivoli.itcam.hc.snapshot.automatic.interval** property in the same file to a positive integer. The unit of the interval is minute and the default is 360.

  dc_home/runtime/*appserver_version.node_name.server_name*/hc.properties (if the runtime directory does not exist, use *dc_home*/healthcenter/etc/hc.properties)
- To enable or disable memory allocation collection, set the **com.ibm.tivoli.itcam.hc.events.collection.automatic.enable** property in the following file to true or false.

  dc_home/runtime/*appserver_version.node_name.server_name*/hc.properties (if the runtime directory does not exist, use *dc_home*/healthcenter/etc/hc.properties)

  Remember: To enable memory allocation collection, you also need to ensure the following two lines are added to the jvm.options file of the Liberty server.

  ```
  -Xhealthcenter:level=inprocess
  -Xgc:allocationSamplingGranularity=10000
  ```

  After memory allocation collection is enabled, data is available in the Memory Analysis dashboard.
- To specify the interval at which memory allocation information is collected, set the **com.ibm.tivoli.itcam.hc.events.collection.automatic.interval** property in the same file to a positive integer. The unit of the interval is minute and the default is 15.

  dc_home/runtime/*appserver_version.node_name.server_name*/hc.properties (if the runtime directory does not exist, use *dc_home*/healthcenter/etc/hc.properties)
- To enable or disable method tracing, set the **dfe.enable.methoddata** property in the following file to true or false:

  dc_home/runtime/*appserver_version.node_name.profile_name.server_name*/custom/gdc/gdc_custom.properties (if the runtime directory does not exist, use *dc_home*/gdc/etc/gdc_dfe.properties)

Results

After you modify the .properties files, restart the Liberty server for the change to take effect.

For more information about the data collector .properties files for each application server, see “Properties files for Liberty data collector” on page 874.

What to do next

- After method tracing is enabled, you can set thresholds for different types of requests, so that different levels of monitoring data can be collected for different requests. For instructions, see “Customizing the request thresholds” on page 878.
- If you disabled memory allocation collection, remember to remove the following lines from the jvm.options file of the Liberty server:

  ```
  -Xhealthcenter:level=inprocess
  -Xgc:allocationSamplingGranularity=10000
  ```

Configuring collection of detailed diagnostic information

If you have IBM Cloud Application Performance Management, Advanced, you can use the data collector to collect detailed diagnostic information on the monitored application server instance. To configure the behavior of the diagnostic data collection, including the amount of diagnostic information that the data collector stores, customize the `gdc_custom.properties` configuration file.

About this task

You can find the `gdc_custom.properties` file in the *dc_home*`/runtime/`*appserver_version.node_name.profile_name.server_name*`/custom/`gdc directory.

The following examples describe how to use the properties in the `gdc_custom.properties` configuration file to do the following things:

- Setting limits for the size and number of detailed information files
- Setting full or partial collection of request and method diagnostic data

You can also set other properties in the `gdc_custom.properties` file to customize collection of diagnostic data. Refer to the comments in the file that describe the properties.

Remember: After you complete editing the `gdc_custom.properties` file, you must restart the monitored application server instance for the changes to take effect.

Setting limits for the size and number of detailed information files

About this task

The data collector stores diagnostic information in a number of files. By default, it stores 100 files; if 100 files are already stored and a new file is created, the oldest file is deleted. The data collector creates a new file every 15 minutes, or when the size of the current file exceeds 200 megabytes. When the total size of the directory that contains the files exceeds 2 gigabytes, the data collector deletes the oldest file.

Procedure

You can change the following settings in the *dc_home*`/runtime/`*appserver_version.node_name.profile_name.server_name*`/custom/gdc/gdc_custom.properties` file:

- To set the maximum number of files with diagnostic information, set the `com.ibm.itcam.gdc.dfe.filelimit` property.
  For example:

  ```
  com.ibm.itcam.gdc.dfe.filelimit=100
  ```

- To set the time, in minutes, after which the data collector creates a new diagnostic data file, set the `com.ibm.itcam.gdc.dfe.frequency` property.
  For example:

  ```
  com.ibm.itcam.gdc.dfe.frequency=15
  ```

- To set the maximum diagnostic data file size, in megabytes, set the `dfe.file.maxlimit` property.
  For example:

  ```
  dfe.file.maxlimit=200
  ```

  If the current diagnostic data file reaches this size, the data collector creates a new diagnostic data file.

- To set the maximum total size of all data files, in bytes, set the `trace.dir.size.limit` property.
  For example:

  ```
  trace.dir.size.limit=2147483648
  ```

If the sum of the sizes of all the diagnostic data files exceeds this value, the data collector deletes the oldest data file. The minimum total size is 25 megabytes.

Setting full or partial collection of request and method diagnostic data

About this task

The data collector has the following default settings:

- The data collector collects diagnostic data only for the selected requests. The selection (sampling) of the requests aims to include all errors and some good requests.
- Method data collection is disabled at server startup.
- When method data collection is enabled, the data collector gathers method data only for some requests (of those for which diagnostic data is collected). This further selection (sampling) again aims to include all errors and some good requests.

Important: Changing these settings affects performance of the application server. On production servers, the performance degradation might be critical.

Procedure

You can change these settings by using properties in the `*dc_home*/runtime/*appserver_version.node_name.profile_name.server_name*/custom/gdc/gdc_custom.properties` file.

- To enable method data collection, set the property as follows:

  ```
  dfe.enable.methoddata=true
  ```

  Tip: You can also use the **Agent Configuration** page to dynamically enable or disable the method trace data collection.
- To collect diagnostic data for every request, disable the sampling by setting the property as follows:

  ```
  dc.sampling.enable=false
  ```
- To enable method data collection for every request for which diagnostic data is collected, set the property as follows:

  ```
  dc.sampling.enable=false
  dc.sampling.methsampler.enabled=false
  ```

  Remember: The `dc.sampling.methsampler.enabled` property takes effect only when method data collection is enabled on the Agent Configuration page or by the `dfe.enable.methoddata` property.

Customizing the request thresholds

Some of the requests might not have enough information if the default thresholds are high. You can customize the request thresholds so that more requests or request context data can be captured by the data collector.

About this task

Each request type has two threshold types, which are named **`perfThreshold`** and **`secondaryPerfThreshold`**. A request is captured by the data collector only when it takes more time than what is specified for the **`perfThreshold`** threshold. Context data, such as stack trace and SQL statement, is captured only when the request takes more time than what is specified for the **`secondaryPerfThreshold`** threshold. You can adjust these threshold values to suit your needs.

Procedure

1. Go to the *dc_home*`\gdc\etc` directory, where *dc_home* is the home directory of the data collector.

2. In a text editor, open the XML file for the request type that you want to customize. You can tell which file is for which request type from the XML file name.
   For example, the `ejb.xml` file is for EJB requests, the `custom.xml` file is for custom requests, and the `appMethods.xml` file is for class and methods when method trace is enabled.
3. Set the `<collectContextData>`, `<collectStackTrace>`, and `<createDataRow>` tags to `ifThresholdExceeded`.

   ```
   <collectContextData>ifThresholdExceeded</collectContextData>
   <collectStackTrace>ifThresholdExceeded</collectStackTrace>
   <createDataRow>ifThresholdExceeded</createDataRow>
   ```

4. Set the `<perfThreshold>` and `<secondaryPerfThreshold>` tags to the threshold values that you desire. The unit of the threshold is millisecond.
   For example, the `ejb.xml` file has the following settings for EJB requests. As a result, only the EJB requests that take more than 1 second (1000 milliseconds) are captured by the data collector. Further, the EJB request-related context data, such as stack trace and EJB home, is captured only when the EJB request takes more than 1.5 seconds (1500 milliseconds).

   ```
   <requestProbePoint id="EJB">
   <interface>com.ibm.tivoli.itcam.toolkit.ai.boot.aspectmanager.ITurboEJBEventListener</
   interface>
   <family>EJB</family>
   <collectContextData>ifThresholdExceeded</collectContextData>
   <collectStackTrace>ifThresholdExceeded</collectStackTrace>
   <perfThreshold>1000</perfThreshold>
   <secondaryPerfThreshold>1500</secondaryPerfThreshold>
   <dataToCollect>instanceAndSummary</dataToCollect>
   <createDataRow>ifThresholdExceeded</createDataRow>
   <requestType>EJB Method</requestType>
   </requestProbePoint>
   ```

5. Save your changes and restart the application server.

Disabling various types of Byte Code Instrumentation for Java EE APIs
In Byte Code Instrumentation (BCI), the data collector intercepts method entry and exit calls for various types of Java Platform Enterprise Edition (Java EE) APIs to create an execution flow of each application request. Some resources are used for the monitoring. You can tune the data collector so that some of the APIs are not monitored, reducing resource use.

To disable BCI monitoring for Java EE APIs, add the following properties to the toolkit custom properties file. For more information about this file, see "Toolkit properties file" on page 874.

Table 233. Adding lines to the toolkit custom properties file

Type of Java EE API	Line to add to `toolkit_custom.properties` file
Enterprise JavaBeans (EJB)	`com.ibm.tivoli.itcam.toolkit.ai.enableejb=false`
Java Connector Architecture (JCA)	`com.ibm.tivoli.itcam.toolkit.ai.enablejca=false`
Java Database Connectivity (JDBC)	`com.ibm.tivoli.itcam.toolkit.ai.enablejdbc=false`
Java Naming and Directory Interface (JNDI)	`com.ibm.tivoli.itcam.toolkit.ai.enablejndi=false`
Java Message Service (JMS)	`com.ibm.tivoli.itcam.toolkit.ai.enablejms=false`
Web containers for Servlets/JavaServer Pages (JSP)	`com.ibm.tivoli.itcam.dc.was.webcontainer=false`

Table 233. Adding lines to the toolkit custom properties file (continued)

Type of Java EE API	Line to add to `toolkit_custom.properties` file
HTTP session count tracking	`com.ibm.tivoli.itcam.toolkit.ai.enablesessioncount=false`
CICS® Transaction Gateway (CTG)	`com.ibm.tivoli.itcam.dc.ctg.enablectg=false`
IMS	`com.ibm.tivoli.itcam.dc.mqi.enableims=false`
Java Data Objects (JDO)	`com.ibm.tivoli.itcam.dc.mqi.enablejdo=false`
Message Queue Interface (MQI)	`com.ibm.tivoli.itcam.dc.mqi.enablemqi=false`
Axis web service	`com.ibm.tivoli.itcam.toolkit.ai.axis.enablewebservice=false`
Remote Method Invocation (RMI)	`am.ejb.rmilistener.enable=false`
WebSphere Application Server EJB container	`com.ibm.tivoli.itcam.dc.was.enableEJBContainer=false`

Disabling transaction tracking for a certain type of transactions

When transaction tracking is enabled for the data collector, all types of transactions are by default monitored. You can use the toolkit properties file to disable transaction tracking for specific types of transactions.

About this task

Edit the `toolkit_custom.properties` file to customize transaction tracking for each application server or edit the `toolkit_global_custom.properties` file for all the application server instances.

The `toolkit_custom.properties` file is used in the following procedure for a single application server. The properties are also supported in the `toolkit_global_custom.properties` file. For more information about the toolkit properties files, see Property files for Liberty data collector.

Procedure

1. Open the `toolkit_custom.properties` file of the application server with a text editor. This file can be found in the following directory:

   dc_home/runtime/*app_server_version*.*node_name*.*profile_name*.*server_name*/custom

2. According to your needs, specify one or more of the following properties and set the property value to `false` to disable transaction tracking for a certain type of transactions.

   For CICS requests
   `com.ibm.tivoli.itcam.dc.ttapi.cics.enabled=false`

   For custom requests
   `com.ibm.tivoli.itcam.dc.ttapi.ims.enabled=false`

   For EJB requests
   `com.ibm.tivoli.itcam.dc.ttapi.ejb.enabled=false`

   For HTTP Client calls
   `com.ibm.tivoli.itcam.dc.ttapi.httpclient.enabled=false`

   Exception: To disable transaction tracking for Apache HTTP Client calls, specify `com.ibm.tivoli.itcam.toolkit.dc.enable.apache.httpclient=false`.

For IMS requests
`com.ibm.tivoli.itcam.dc.ttapi.ims.enabled=false`

For JDBC requests
`com.ibm.tivoli.itcam.dc.ttapi.jdbc.enabled=false`

For JMS requests
`com.ibm.tivoli.itcam.dc.ttapi.jms.enabled=false`

For JNDI requests
`com.ibm.tivoli.itcam.dc.ttapi.jndi.enabled=false`

For MQI requests
`com.ibm.tivoli.itcam.dc.ttapi.mqi.enabled=false`

For Portal requests
`com.ibm.tivoli.itcam.dc.ttapi.portal=false`

For RMI-IIOP requests
`com.ibm.tivoli.itcam.dc.ttapi.rmiiiop.enabled=false`

For Servlet requests
`com.ibm.tivoli.itcam.dc.ttapi.arm.servlet.enabled=false`

For Web service requests
`com.ibm.tivoli.itcam.dc.ttapi.webservice.enabled=false`

Tip: For more information about these properties, refer to the *dc_home*/ttdc/etc/ttdc.properties file.

3. Save and close the `toolkit_custom.properties` file.
4. Restart the application server for the changes to take effect.

Excluding classes from being monitored

You can customize data collection by excluding certain classes from being monitored. Use the toolkit properties file for this customization.

About this task

Edit the `toolkit_custom.properties` file to customize transaction tracking for each application server or edit the `toolkit_global_custom.properties` file for all the application server instances.

The `toolkit_custom.properties` file is used in the following procedure for a single application server. The properties are also supported in the `toolkit_global_custom.properties` file. For more information about the toolkit properties files, see Property files for Liberty data collector.

Procedure

1. Open the `toolkit_custom.properties` file of the application server with a text editor. This file can be found in the following directory:

   dc_home/runtime/*app_server_version*.*node_name*.*profile_name*.*server_name*/custom
2. Edit the file to add the following property and save your changes.

```
am.camtoolkit.gpe.customxml.exclude=excludes.xml
```

3. In the same `custom` directory, create the `excludes.xml` file with the following content and specify the class name to be excluded within the <exclude> tag. You can add as many classes as needed and the asterisk wildcard (*) is supported.

```
<gpe>
  <bci>
    <classExcludes>
      <exclude>class_name_to_be_exclued</exclude>
      <exclude>class_name_to_be_exclued</exclude>
    </classExcludes>
  </bci>
</gpe>
```

Example:

```
<gpe>
  <bci>
    <classExcludes>
      <exclude>org.apache.struts.action.ActionServlet</exclude>
      <exclude>com.company.package.*</exclude>
    </classExcludes>
  </bci>
</gpe>
```

4. Restart the application server.

What to do next

To verify that the class are excluded, look in the `toolkit.xml` file and the class name should be listed in the `<classExcludes>` section.

Remember: The `toolkit.xml` file contains runtime settings and it is refreshed every time the application server is restarted.

Customizing request information mapping

In some cases, you might have to change the information that identifies the requests monitored by the agent. This information includes the request name, and any data that can be displayed for the request (for example, the query text for an SQL request). To change the information, set up a custom request mapper configuration.

About this task

To customize request information mapping, you must define a custom request mapper configuration in an XML file.

In this file, some built-in *symbols* represent values from the runtime context of the request. You can create extra symbols, which calculate new values. The calculation can include original request values, expressions, calls to Java methods (including methods in the monitored application), conditionals, and iteration over a set of values.

Then, you can *map* the contents of the symbols into the new request data that is provided to Cloud APM server. If a particular variable in the request data is not mapped, the original value is retained.

Because different data is collected for request types, a custom request mapper configuration must be specific for a request type. You can configure different request mappers for different request types on the same data collector.

Procedure

To set a custom request mapper configuration for a request type, complete the following steps:

1. Define a custom request mapper configuration in an XML file.

   For information about the XML syntax, see "XML file syntax" on page 883.

2. Place the XML file in the *dc_home*`/runtime/custom` directory to use it for all application server instances, or in the *dc_home*`/runtime/`*appserver_version.node_name.profile_name.server_name*`/custom` directory to use it for one application server instance.

3. Enable custom request mapping for this type in the toolkit custom configuration file, `toolkit_custom.properties`, or `toolkit_global_custom.properties`.

   For instructions, see "Enabling a request mapper" on page 892.

4. Reference the XML file that you defined from the same toolkit custom configuration file.

   For instructions, see "Enabling a request mapper" on page 892.

XML file syntax

The XML file that you create for request mapper configuration must be valid XML and must remain available when the configuration is in use. Place the XML file in the `*dc_home*/runtime/custom` directory to use it for all application server instances, or in the `*dc_home*/runtime/*appserver_version.node_name.profile_name.server_name*/custom` directory to use it for one application server instance.

Top level

The top-level tag is <gpe>. Within this tag, use the <runtimeConfiguration> tag. These tags have no attributes.

Within the <runtimeConfiguration> tag, create a <requestMapperDefinition> tag. This tag must have a `type` attribute. Set it to the request mapper type name for the required request type. For more information, see Table 235 on page 894.

Within the <requestMapperDefinition> tag, the following two tags must be present:

<symbolDefinitions>
Contains all definitions of symbols. Symbols represent values that the agent calculates every time a request of this type is detected.

<selection>
Contains the mapping of context keys to values. The keys represent the custom data that is passed to the agent. They are predefined for each request type. The mapping can be conditional.

For more information about request mapper enabling properties and type names, see Table 235 on page 894.

Also, within the <runtimeConfiguration> tag, you can create a <requestMapperClassPath> tag. Within this tag, you can define JAR files. You can reference Java classes in these JAR files within Request Mapper definitions.

Defining an expression

To define symbols, you must use expressions. The agent evaluates the expressions to assign values to symbols.

Using data in an expression

An expression can use the following data:

- The input data symbols for the request type
- Other symbols described in the same request mapper definition
- Numeric constants
- String constants (delimited with `"`, for example, `"string"`)
- Boolean constants (`true`, TRUE, `false`, FALSE)
- The `null` constant

For more information about input data symbols, see Table 236 on page 895.

If the value of a symbol is an instantiation of a Java class, expressions can contain references to fields and methods that are defined within the class. To refer to a field, use `*symbol.fieldname*`. To refer to a method, use `*symbol.methodname*(*parameters*)`. The method call must return a value. For example, you can use the Java String methods with a symbol that has a String value.

To refer to a static field or method of a class, you can also use `*classname.fieldname*` and `*classname.methodname*(*parameters*)`.

If a symbol refers to an array object, the expression can select an element (`*symbol*[*selector*]`) and determine the length of the array (`*symbol*.length`)

Operators

You can use the following operators in an expression:

- Boolean operators: AND, &, OR, |, NOT, !
- Comparison: ==, !=, GT, >, LT, <, GE, >=, LE, <=
- Numeric operators: +, -, *, /
- Parentheses to force order of evaluation: (,)

Important: You must escape the symbols <, >, and & in XML. Alternatively, you can use the GT (greater than), GE (greater than or equal), LT (less than), LE (less than or equal), and AND operators.

The expression can evaluate whether a value is an instance of a class, by using the `instanceof` operator:

```
expression instanceof java.class.name
```

This operator, similar to the Java `instanceof` operator, produces a Boolean value. In this example, the value is true if the class to which the *expression* value belongs meets any of the following conditions:

- Is named *`java.class.name`*
- Is a direct or indirect subclass of the class identified by *`java.class.name`*.
- Implements, directly or indirectly, the interface identified by *`java.class.name`*.

The expression can also instantiate a new object of a Java class, by using the `new` operator. This operator is similar to the Java new operator:

```
new java.class.name(expression1, expression2, ... expressionN)
```

Operator precedence

Operators are evaluated in order of precedence. Operators of the same order of precedence are evaluated from left to right. You can change the order of evaluation by using parentheses (and).

The order of precedence is as follows:

1. . operator (method call or field reference)
2. [ ] (array element selector)
3. new
4. !, NOT
5. *, /
6. +, -
7. GT, >, LT, <, GE, >=, LE, <=, instanceof
8. ==, !=
9. AND, &
10. OR, |

Example

```
$s1 >= ( 2 * ($s2.sampMethod($s3, true) + 1))
```

The agent evaluates this expression in the following way:

1. The $s1 symbol is evaluated. It must yield a numeric value.
2. The $s2 symbol is evaluated. It must yield a Java object.
3. The $s3 symbol is evaluated.

4. The sampMethod method for the object that results from the evaluation of $s2 is called. The result of the evaluation of $s3 is passed as the first parameter, and the Boolean value true is passed as the second parameter. The call to sampMethod must return a numeric value.
5. 1 is added to the result of step "4" on page 885.
6. The result of step "5" on page 885 is multiplied by 2.
7. The result of step "1" on page 884 is compared with the result of step "6" on page 885. If the result of step "1" on page 884 is greater than or equal to the result of step "6" on page 885, true is returned. Otherwise, false is returned.

Defining basic symbols
To define symbols, you must use expressions. The agent evaluates the expressions to assign values to symbols.

Within the <symbol> tag, use the following tags:

<name>
The name of the symbol. It is a string and must start with the $ character.

<eval>
The expression that the agent must evaluate to produce the value for this symbol. For more information about defining expressions, see "Defining an expression" on page 883.

<type>
The type of the value that the symbol returns. Specify this value as a fully qualified Java class name, or a Java primitive. Specifying the type for the symbol is optional. If it is not defined, the request mapper attempts to establish the field type based on the expression. If the request mapper is unable to determine the symbol type before it evaluates the expression, performance is affected. Therefore, for best performance, it is better to specify the type.

<args>
The arguments for the symbol. This tag is optional; if it is specified, arguments must be supplied for evaluating the symbol. For more information, see "Defining symbol arguments" on page 885.

Example

```
<symbol>
   <name>$doubles1</name>
   <eval>$s1*2</eval>
   <type>int</type>
</symbol>
```

This symbol returns double the value of another symbol, $s1.

Defining symbol arguments

Within the <args> tag of a symbol definition, you can define argument types for the symbol.

In this tag, use the <type> tag to specify the types of arguments. Specify this value as a fully qualified Java class name, or a Java primitive. You can specify any number of <type> tags; each of these tags defines an argument.

In this case, the symbol must be referenced with arguments in parentheses:

```
$symbol(argument1,argument2...)
```

The number of arguments must be the same as the number of argument type definitions.

Within the symbol definition, refer to the first argument as $p0, the second argument as $p1, and so on.

A symbol with arguments works like a Java method. It takes input arguments, and returns a value that depends on the values of the arguments.

Example

```
<symbol>
   <name>$double</name>
   <eval>$p0*2</eval>
   <type>int</type>
   <args>
       <type>int</type>
   </args>
</symbol>
```

This symbol returns double the value of the argument. To evaluate it, supply a numeric argument: `$double(2)`, `$double($s1)`.

Defining iteration symbols
Within the <symbolDefinitions> tag, you can define an iteration symbol by using the <iterationSymbol> tag. An iteration symbol represents a value that is acquired by iterating through a set of objects in a Java array, enumeration, or collection. For each of the members, the request mapper evaluates one or more condition expressions. If an expression returns `true`, request mapper uses the member to calculate the return value. When a member meets the condition expression, the request mapper does not evaluate the rest of the members.

Within the <iterationSymbol> tag, use the following tags.

<name>
The name of the symbol. It is a string and must start with the $ character.

<type>
The type of the value that the symbol returns. Specify this value as a fully qualified Java class name or a Java primitive. Specifying the type for the symbol is optional. If it is not defined, the request mapper attempts to establish the field type based on the expression. If the request mapper is unable to determine the symbol type before it evaluates the expression, performance is affected. Therefore, for best performance, it is better to specify the type.

<args>
The arguments for the symbol. This tag is optional; if it is specified, arguments must be supplied for evaluating the symbol. For more information, see "Defining symbol arguments" on page 885.

<iterate over="*expression*">
Defines the object (array, Enumeration, or Collection) that contains the members to iterate through. The expression must return such an object. request mapper iterates over its members until either one of them causes a condition expression to return `true`, or no more members remain. Define the set of iteration expressions in tags within this tag:

<test>
Define the condition and return expression within this tag. An <iterate> tag can contain several <test> tags. In this case, request mapper evaluates all of them. If any condition expression is true, the symbol returns a value using the result expression in the same <test> tag, and no further evaluation is performed.

<castTo>
Optional: If this tag is present, specify the name of a Java type within it, as a fully qualified Java class name or a Java primitive. The request mapper casts the iterated element to this type before it evaluates the condition and return expressions. If this tag is not present, request mapper casts a member of an array to the array base type, and a member of an Enumeration or Collection to `java.lang.Object`. For an array member, the array base type is usually the correct choice; therefore, use this tag for request mapper to iterate over an enumeration or collection.

<condition>
An expression that must yield a Boolean value. Use `$iterElement` to refer to the element that is being iterated.

<return>

If the expression in the <condition> tag returns `true`, request mapper evaluates the expression in the <return> tag. The iteration symbol returns the value that this expression produces. Use `$iterElement` to refer to the element that is being iterated.

<defaultValue>

Optional. If the request mapper has iterated over all members of the object, but no condition expression has returned `true`, the request mapper evaluates the expression in the <defaultValue> tag. The iteration symbol returns the value that the expression produces. If this tag is not present, the default value is `null`.

Examples

```
<iterationSymbol>
  <name>$userNameCookieValue</name>
  <iterate over="$httpServletRequest.getCookies()">
     <test>
        <condition>$iterElement.getName().equals("userName")</condition>
        <return>$iterElement.getValue()</return>
     </test>
   </iterate>
</iterationSymbol>
```

This symbol finds the cookie named "username", and returns its value. `$httpServletRequest.getCookies()` returns an array, so there is no need for the <castTo> element.

```
<iterationSymbol>
  <name>$headerNameStartingWithA</name>
  <iterate over="$httpServletRequest.getHeaderNames()">
    <test>
       <castTo>java.lang.String</castTo>
       <condition>$iterElement.startsWith("A")</condition>
       <return>$iterElement</return>
    </test>
  </iterate>
</iterationSymbol>
```

This symbol finds the header with a name starting with "A", and returns its name. `$httpServletRequest.getHeaderNames()` returns an Enumeration, so the <castTo> element is required.

```

<iterationSymbol>
  <name>$determined_gender</name>
  <iterate over="$children">
    <test>
       <castTo>java.lang.String</castTo>
       <condition>$iterElement.equals("male")</condition>
       <return>"It's a boy"</return>
    </test>
    <test>
       <castTo>java.lang.String</castTo>
       <condition>$iterElement.equals("female")</condition>
       <return>"It's a girl"</return>
    </test>
  </iterate>
  <defaultValue>"unknown"</defaultValue>
</iterationSymbol>
```

This symbol iterates over `$children`, which must be an array, Enumeration, or Collection of strings. If any of the strings equals "male", it returns "it’s a boy". If any of the strings equals "female", it returns "it’s a girl". Finally, if no string in the `$children` object equals either "male" or "female", the symbol returns "unknown".

Defining conditional symbols

Within the <symbolDefinitions> tag, you can define a conditional symbol by using the <conditionalSymbol> tag. A conditional symbol represents a value that is acquired by evaluation a series of condition expressions. If any expression returns `true`, request mapper uses the member to calculate the return value. When a member meets the condition expression, request mapper evaluates a corresponding return expression, and return the result. After the request mapper finds a result to return, it does not evaluate any further expressions.

Within the <conditionalSymbol> tag, use the following tags.

<name>

The name of the symbol. It is a string and must start with the $ character.

<type>

The type of the value that the symbol returns. Specify this value as a fully qualified Java class name, or a Java primitive. Specifying the type for the symbol is optional. If it is not defined, the request mapper attempts to establish the field type based on the expression. If the request mapper is unable to determine the symbol type before it evaluates the expression, performance is affected. Therefore, for best performance, it is better to specify the type.

<args>

The arguments for the symbol. This tag is optional; if it is specified, arguments must be supplied for evaluating the symbol. For more information, see "Defining symbol arguments" on page 885.

<if condition="*expression*">

The `condition` attribute defines a condition expression to evaluate. The expression must yield a Boolean value. If the value is `true`, request mapper uses the contents of the <if> tag to try to determine the return value. The <if> tag must contain either, but not both, of the following contents:

- A <return> tag. This tag contains an expression. If the condition expression is true, request mapper evaluates the expression and returns the result.
- Any number of <if> tags, nested within this <if> tag. If the condition expression is true, request mapper processes the nested <if> tags in the same way as a top-level <if> tag. That is, it evaluates the expression in the `condition` attribute, and if the expression is true, uses the contents of the tag to try and determine the return value.

Important: If a return value is determined, request mapper does not evaluate any further expressions. However, if a condition expression in an <if> tag is true, but it contains nested <if> tags and none of their condition expressions are true, no value is determined. In this case, request mapper continues to evaluate subsequent expressions.

<defaultValue>

Optional. If request mapper has evaluated all condition expressions, but none of the condition expression has returned `true`, request mapper evaluates the expression in the <defaultValue> tag. The conditional symbol returns the value that the expression produces. If this tag is not present, the default value is `null`.

Example

```
<symbol>
  <name>$GET</name>
  <eval>"GET"</eval>
</symbol>
<symbol>
  <name>$PUT</name>
  <eval>"PUT"</eval>
</symbol>
<conditionalSymbol>
  <name>$sessionAttribute</name>
  <if condition="$httpServletRequest.getSession(false) != null>
     <if condition="$httpServletRequest.getSession(false).getAttribute($GET)
!= null">
       <return>$httpServletRequest.getSession(false).getAttribute($GET)</return>
     </if>
     <if condition="true">
       <return>$httpServletRequest.getSession(false).getAttribute($PUT)</return>
      </if>
  </if>
</conditionalSymbol>
```

This symbol is assumed to be a part of the servlet request mapper. First, it checks whether an HTTP session exists for the servlet; if not, the symbol returns `null`. If a session is present, the symbol checks whether the servlet has a GET attribute, it returns the value of that attribute. Otherwise, it returns the value of the PUT attribute. The second condition expression is true; this value is used as an else clause. If the first condition is true, the request mapper does not evaluate any further expressions; otherwise, it continues to the second expression.

Defining external class symbols
Within the <symbolDefinitions> tag, you can define an external class by using the <externalClassSymbol> tag. An external class symbol represents an external Java class. External class symbol definition is optional; you can use external Java classes in expressions directly. However, it might enhance the readability of the request mapper configuration.

Within the <externalClassSymbol> tag, use the following tags.

<name>
The name of the symbol. It is a string and must start with the $ character.

<className>
The name of the customer defined class.

Important: To refer to any Java class in request mapper configuration, whether in an external class symbol definition or in any expression, you must add the full path and name of the JAR file that contains the class to the <requestMapperClassPath> tag within the <runtimeConfiguration> tag.

After defining an external symbol, you can refer to the class by the name of the symbol. You can also refer to static methods and fields of the class by using the symbol.

Example

```
<externalClassSymbol>
  <name>$rand</name>
  <className>user.class.Random</className>
</externalClassSymbol>
```

This symbol refers to a user-written class, generating a random number. The full path and name of the JAR file that contains this class must be present in the <requestMapperClassPath> tag within the <runtimeConfiguration> tag.

To refer to the static method `user.class.Random.generate()` in an expression, you can use the external symbol:

```
$rand.generate()
```

Mapping values to context keys
Within the <requestMapperDefinition> tag, map values to context keys by using the <selection> tag. This mapping provides the changes in the monitoring information.

You can map values to the output keys defined for the request type. For more information, see Table 235 on page 894.

If no value is mapped to a key after the evaluation of the request mapper configuration, ITCAM uses the original value extracted from the request.

Within the <selection> tag, use the following tags.

<matchCriteria>
An expression that must return a Boolean value. The mapping that is defined within this tag is only used if this expression returns `true`.

<mapTo>
Defines a key and the value to map to it. Within this tag, a <key> tag contains the key, and a <value> tag contains the value.

<selection>
  You can nest <selection> tags, placing one within another.

If <selection> tags are nested, then the nested mapping is only used if both the outer and the nested <matchCriteria> expressions return `true`.

You can use multiple <selection> tags within a <requestMapperDefinition> tag or within another <selection> tag. If the same key is mapped several times in several <selection> tags on the same nesting level (that is, within the same parent tag), then the first mapping for which the <matchCriteria> expression returned `true` is used.

Do not map the same key both in the outer and nested <selection> tags.

Typically, use the <matchCriteria> value of `true` as an "else" value for the last selection tag on a nesting level. If you want to map different values in different cases, use several <selection> tags within this outer tag; each of them can contain the criteria and values for a particular case. The last tag, with a value of `true`, covers the case when the available data meets none of the criteria.

Examples

```
<selection>
   <matchCriteria>true</matchCriteria>
   <mapTo>
      <key>Result</key>
      <value>$s1</value>
   </mapTo>
</selection>
```

In this mapping configuration, `Result` is set to the value of the symbol `$s1`.

```
<matchCriteria>true</matchCriteria>
   <selection>
      <matchCriteria>$b1</matchCriteria>
      <mapTo>
         <key>Result</key>
         <value>1</value>
      </mapTo>
   </selection>
   <selection>
      <matchCriteria>true</matchCriteria>
      <mapTo>
         <key>Result</key>
         <value>2</value>
      </mapTo>
</selection>
```

In this mapping configuration, the symbol `$b1` must return a Boolean value. `Result` is set to 1 if `$b1` returns `true`, and to 2 if `$b1` returns `false`. If `$b1` returns `true`, request mapper uses the mapping for `Result` in the first <selection> tag; the mapping for the same key in the second tag is not used.

Defining custom requests

By default, only certain types of Java classes and methods are monitored as requests by the data collector. Servlets, JSPs, EJB business methods, and certain standard Java EE APIs are recognized as requests. You can designate extra classes and methods as *custom requests*.

About this task

To enable monitoring of custom requests by the data collector, define the custom requests in an XML file and set the `am.camtoolkit.gpe.customxml.custom` property in the toolkit custom properties file.

For example, the data collector does not recognize Struts Action classes as requests by default. However, you can set up custom request definitions and cause the actions to be recognized as Nested Requests.

Procedure

Complete the following procedure to enable monitoring of custom requests and designate one or more methods as custom requests:

1. Make a copy of the *dc_home*/itcamdc/etc/custom_requests.xml file in a temporary location. Then, open the copy in a text editor.
2. Modify the parameters in the file.

   The following table describes the parameters that you can modify:

Table 234. Parameters for the custom requests configuration file

Tag name	Description
edgeRequest	Identifies one or more application methods that are to be Byte-Code-Instrumented for custom request processing. By modifying the requestName, Matches, type, and methodName tags within the edgeRequest tag, you can customize the selection. Each edgeRequest tag must contain exactly one methodName tag, and one or more Matches tags. Multiple edgeRequest tags can be specified.
requestName	Defines a unique name for this request. The request name is displayed to the user when the method entry and exit are traced.
Matches	Identifies a class or classes that contain the methods that are to be Byte-Code-Instrumented for custom request processing. Multiple Matches tags can be present within a single edgeRequest tag.
type	Indicates whether a class must be a system or application class to match the edgeRequest tag.
methodName	Identifies the names of the methods within one of the classes identified by the Matches tag that are to be Byte-Code-Instrumented for custom request processing. Exactly one methodName tag can be specified in each edgeRequest tag.
requestMapper	Optional. If this tag is specified, the data collector uses a request mapper to determine information that identifies the request. You can define nonstandard ways of extracting this information. For more information about enabling and defining request mappers, see "Customizing request information mapping" on page 882.

Remember: The Matches and methodName tags can include wildcard characters. How the wildcard characters works is described as follows:

- Asterisk (*) stands for zero or more occurrences of any character when used by itself. When embedded within a sequence of characters (for example, java.*.String), it matches zero or more occurrences of any character except the package separator (.).
- Two periods (..) can be used to specify all sub-packages. It matches any sequence of characters that starts and ends with the package separator (.). For example, java..String matches java.lang.String and com.ibm..* matches any declaration beginning with com.ibm.

For example, an application with a package name of com.mycompany.myapp has the following requirements:

- Within the Customer class, the creditCheck() method must be treated as a custom request called CreditCheck.
- Within the Supplier class, the inventoryCheck() method must be treated as a custom request called SupplyCheck.

The contents of the customized custom_requests.xml file that accomplishes the requirements are as follows:

```
<customEdgeRequests>
      <edgeRequest>
              <requestName>CreditCheck</requestName>
              <Matches>com.mycompany.myapp.Customer</Matches>
              <type>application</type>
              <methodName>creditCheck</methodName>
      </edgeRequest>
      <edgeRequest>
```

```
              <requestName>SupplyCheck</requestName>
              <Matches>com.mycompany.myapp.Supplier</Matches>
              <type>application</type>
              <methodName>inventoryCheck</methodName>
      </edgeRequest>
</customEdgeRequests>
```

3. Complete one of the following steps:

   - Save the file in the *dc_home*/runtime/*app_server_version.node_name.profile_name.server_name*/custom directory. Then, in the toolkit custom properties file, set the property am.camtoolkit.gpe.customxml.custom to the name (without path) of the file that you modified in Step "2" on page 891.
   - Save the file in any directory on your computer. Then, in the toolkit custom properties file, set the property am.camtoolkit.gpe.customxml.custom to the path and name for the file that you modified in Step "2" on page 891.

   For more information about the toolkit custom properties file, see"Toolkit properties file" on page 874.

Enabling a request mapper

To enable a request mapper for a request type, edit the toolkit custom configuration file or the toolkit global custom configuration file. Procedures are different for common request types and for custom requests.

Before you begin

Define the request mapper configuration in an XML file. Then, place the XML file that contains request mapper configuration in the same directory as the toolkit properties file.

- To enable the request mapper for all application server instances, place it in the *dc_home*/runtime/custom directory.
- To enable the request mapper for one application server instance, place it in the *dc_home*/runtime/*appserver_version.node_name.profile_name.server_name*/custom/ directory.

For information about the XML file syntax, see "XML file syntax" on page 883.

About this task

Edit the toolkit_custom.properties file or the toolkit_global_custom.properties file to enable the request mapper for one or all application server instances.

Procedure

- To enable a request mapper for common requests, complete the following steps:

  a) In a text editor, open one of the following toolkit custom configuration files:

     – To enable the request mapper for all application server instances, open the *dc_home*/runtime/custom/toolkit_global_custom.properties file.

     – To enable the request mapper for one application server instance, open the *dc_home*/runtime/*appserver_version.node_name.profile_name.server_name*/custom/toolkit_custom.properties file.

  b) Edit the toolkit properties file as follows:

     – Add a line setting the enabling property for this request type to true. For more information, see Table 235 on page 894.

     – Add a line setting the am.camtoolkit.gpe.customxml.* property to the name of the mapper XML file. Use any unique value instead of the * symbol. For more information, see "XML file syntax" on page 883.

  c) Save and close the properties file.

  Example:

To enable a request mapper that is defined in renameDataSource.xml for the SQL request type, add the following lines to the toolkit custom configuration file or the toolkit global custom configuration file:

```
com.ibm.tivoli.itcam.toolkit.ai.enable.sqlrequestmapper=true
am.camtoolkit.gpe.customxml.renameDataSource=renameDataSource.xml
```

- To enable a request mapper for custom requests, complete the following steps:

  a) Under the <edgerequest> tag in the custom request definition XML file, create a <requestMapper> tag. Place a unique request map type name in this tag. For information about defining custom requests, see "Defining custom requests" on page 890.

  b) In the request mapper XML file, use the unique request map type name in the type attribute of the <requestMapperDefinition> tag.

  c) In a text editor, open one of the following toolkit custom configuration files:

    – To enable the request mapper for all application server instances, open the *dc_home*/runtime/custom/toolkit_global_custom.properties file.

    – To enable the request mapper for one application server instance, open the *dc_home*/runtime/*appserver_version.node_name.profile_name.server_name*/custom/toolkit_custom.properties file.

  d) Edit the toolkit properties file to add a line setting the am.camtoolkit.gpe.customxml.* property to the name of the mapper XML file. Use any unique value instead of the * symbol. For more information, see "XML file syntax" on page 883.

  e) Save and close the properties file.

  Example:

  To enable a request mapper that is defined in customMapper.xml for the SupplyCheck custom request type that is defined in the custom_requests.xml file, complete the following steps:

  1. Use the following definition in the custom_requests.xml file:

```
<customEdgeRequests>
  <edgeRequest>
    <requestName>SupplyCheck</requestName>
    <Matches>com.mycompany.myapp.Supplier</Matches>
    <type>application</type
    <methodName>inventoryCheck</methodName>
    <requestMapper>customMapper</requestMapper>
  </edgeRequest>
</customEdgeRequests>
```

  2. In the customMapper.xml file, make sure that the type name is set:

```
<requestMapperDefinition type="customMapper">
```

  3. Add the following line to the toolkit custom configuration file or the toolkit global custom configuration file:

```
am.camtoolkit.gpe.customxml.customMapper=customMapper.xml
```

Request mapper type names, input, and output data

The following tables list the information necessary to configure and enable request mappers for different request types.

The meaning of each table header is explained as follows:

Request type
The request type.

Enabling property
To enable the request mapper, set this property to true in the toolkit_custom.properties or toolkit_global_custom.properties file.

Important: If you copy this value from the table, remove any spaces and line breaks.

Request mapper type name
Assign this value to the `type` attribute of the <requestMapperDefinition> tag in the request mapper definition XML file.

Input data symbol names
The symbols that represent the request information. You can use these symbols in expressions within the request mapper definitions. For more information, see "Defining an expression" on page 883.

Output data context keys
To provide changes in the monitoring information, assign values to these keys in the request mapper definition. For more information, see "Mapping values to context keys" on page 889.

Table 235. Request mapper enabling properties and type names

Request type	Enabling property	Request mapper type name
Servlet	`com.ibm.tivoli.itcam.toolkit.ai.enable.servletrequestmapper`	`servlet`
JNDI	`com.ibm.tivoli.itcam.toolkit.ai.enable.jndirequestmapper`	`jndiLookup`
Custom Request		Defined by user in the edgeRequest definition
EJB	`com.ibm.tivoli.itcam.toolkit.ai.enable.ejbrequestmapper`	`ejb`
JCA	`com.ibm.tivoli.itcam.toolkit.ai.enable.jcarequestmapper`	`jca`
JDBC data source	`com.ibm.tivoli.itcam.toolkit.ai.enable.datasourcerequestmapper`	`dataSource`
JDBC SQL statement	`com.ibm.tivoli.itcam.toolkit.ai.enable.sqlrequestmapper`	`sqlStatement`
JMS	`com.ibm.tivoli.itcam.toolkit.ai.enable.jmsrequestmapper`	`jms`
JAX-RPC web service	`com.ibm.tivoli.itcam.toolkit.ai.enable.webservicerequestmapper`	`webServices`
Axis web service	`com.ibm.tivoli.itcam.toolkit.ai.enable.webservicerequestmapper`	`webServices`
MQI	`com.ibm.tivoli.itcam.toolkit.ai.enable.mqrequestmapper`	`mqi`
EJB	`com.ibm.tivoli.itcam.toolkit.ai.enable.ejbrequestmapper`	`ejb`
JDBC connection factory	`com.ibm.tivoli.itcam.toolkit.ai.enable.sqlconnectfactoryrequestmapper`	`connectionFactory`
SCA	`com.ibm.tivoli.itcam.toolkit.ai.enable.scarequestmapper`	`sca`
JAX-WS web service	`com.ibm.tivoli.itcam.toolkit.ai.enable.webservicerequestmapper`	`webServices`

Table 235. Request mapper enabling properties and type names (continued)		
Request type	**Enabling property**	**Request mapper type name**
WebSphere Portal Server Portal (extending the `org.apache.jetspeed.portlet.Portlet` class)	`com.ibm.tivoli.itcam.toolkit.ai.enable.portalrequestmapper`	`portalPortal`
WebSphere Portal Server version 6.1, 7, and 8 Portal (implementing the `javax.portlet.Portlet` interface)	`com.ibm.tivoli.itcam.toolkit.ai.enable.portal6requestmapper`	`Portal6Portal`

Important: There is no meaningful way to configure the custom request mapper for the request types that are not listed in Table 235 on page 894.

Table 236. Request mapper input and output data		
Request type	**Input data symbol names**	**Output data context keys**
Servlet	For more information, see Table 237 on page 899.	**remappedURI** defines a renamed URI **remappedURL** defines a renamed URL **appName** defines a renamed application name **userid** defines the user ID for the request
JNDI	• **$jndiContext** the Context object • **$lookup** the lookup string • **$context** "JNDIlookup"	**renamedLookup** defines a renamed lookup string
Custom Request	• **$this** the ‘this' object for the custom request method • **$0** the arguments passed to the custom request method, specified as an array of Objects • **$className** the custom request class name • **$methodName** the custom request method name • **$context** the original request name from the edgeRequest definition	**customRequestName** defines the renamed custom request name

Table 236. Request mapper input and output data (continued)

Request type	Input data symbol names	Output data context keys
EJB	- **$ejb** the EJB implementation object - **$appName** the name of the application - **$ejbType** the type of the EJB - **$className** the Class name of the EJB implementation object - **$methodName** the EJB business method name - **$context** "EJBBusinessMethod"	**appName** defines the renamed application name **ejbType** defines the renamed EJB type **className** defines the renamed class name **methodName** defines the renamed method name
JCA	- **$interaction** the Interaction object - **$interactionSpec** the InteractionSpec object - **$record** the Record object - **$context** "J2Cexecute"	**lookupName** is the renamed lookupName **productName** is the renamed product name **productVersion** is the renamed product version
JDBC data source	- **$this** either the data source or the driver object - **$dataSource** the $this object, cast as a data source - **$driver** the $this object, cast as a Driver - **$dataSourceName** is the name of the data source, as java.lang.String - **$context** indicates the type of request, either "JDBCgetConnection" or "JDBCgetConnection FromDriver"	**dataSourceName** is the renamed data source name, if the $this object is a data source **url** is the renamed Driver URL, if the $this object is a Driver
JDBC SQL Statement	- **$this** either the SQL statement, or the SQL Connection - **$sqlText** contains the SQL text as java.lang.String, if the $this object is an SQL statement - **$sqlStatement** the $this object, cast as an SQL Statement - **$sqlConnection** the $this object, cast as an SQL Connection - **$dataSourceName** the data source name - **$context** indicates the type of request: "JDBCexecute", JDBCexecuteQuery", "JDBCexecuteUpdate", "JDBCcreateStatement", "JDBCprepareStatement"	**dataSourceName** is the renamed data source name **sqlText** is the renamed SQL text

Table 236. Request mapper input and output data (continued)

Request type	Input data symbol names	Output data context keys
JMS	• **$this** the ‘this' object for the instrumented method. Can be a QueueBrowser, MessageConsumer, MessageProducer, or MessageListener • **$0** the Queue object, for a send request, or a topic object, for a Publish request • **$queueBrowser** the $this object, cast as a QueueBrowser • **$messageConsumer** the $this object, cast as a MessageConsumer • **$messageProducer** the $this object, cast as a MessageProducer • **$messageListener** the $this object, cast as a MessageListener • **$queue** the $0 object, cast as a queue • **$topic** the $0 object, cast as a topic • **$context** indicates the type of request: "JMSreceive", "JMSsend", "JMSbrowse", "JMSpublish", "JMSonmessage"	**queueName** the renamed queue name **providerURL** the renamed provider URL **topicName** the renamed topic name
JAX-RPC web service	• **$messageContext** the IMessageContextWrapper • **$appName** the application name • **$requestName** the default request name • **$url** the URL • **$context** indicates the type of request: "WebServicesJaxRpc ProviderRequest", "WebServicesJaxRpc ClientRequest"	**appName** the renamed application name **requestName** the renamed request name **url** the renamed URL
Axis web service	• **$messageContext** the IMessageContextWrapper • **$appName** the application name • **$requestName** the default request name • **$url** the URL • **$context** indicates the type of request: "WebServicesAxisClient Request", "WebServicesAxis ProviderRequest"	**appName** the renamed application name **requestName** the renamed request name **url** the renamed URL

Table 236. Request mapper input and output data (continued)

Request type	Input data symbol names	Output data context keys
MQI	• **$queue** the MQQueue object, if it is known • **$qmgr** the MQQueueManager object, if it is known • **$message** the MQMessage or MQMsg2 object, if it is known • **$session** the MQSESSION object, if it is known • **$getMsgOptions** the MQGetMessageOptions object, if it is known • **$qmgrName** the name of the queue manager • **$queueName** the name of the queue • **$context** the type of MQ request: "MQCONN", "MQCONNX", "MQDISC", "MQBACK", "MQBEGIN", "MQCLOSE", "MQCMIT", "MQINQ", "MQOPEN", "MQSET", "MQGET", "MQPUT", "MQPUT1", "MQGETBROWSE"	**qmgrName** the rename queue manager name **qname** the renamed queue name
EJB	• **$appName** the name of the application • **$ejbType** the type of the EJB • **$className** the Class name of the EJB implementation object • **$methodName** the EJB business method name • **$context** "EJBBusinessMethod"	**appName** defines the renamed application name **ejbType** defines the renamed EJB type **className** defines the renamed class name **methodName** defines the renamed method name
JDBC Connection Factory	• **$connectionFactory** the ConnectionFactory • **$dataSourceName** the data source name • **$context** "JDBCgetConnection"	**dataSourceName** is the renamed data source name
SCA	• **$uri** the URI • **$operationName** the operation name • **$context** indicates the type of request: "SCA_Generic", "SCA_Ref", "SCA_Target"	**uri** is the renamed URI **operationName** is the renamed operation name

Table 236. Request mapper input and output data (continued)

Request type	Input data symbol names	Output data context keys
JAX-WS web service	• **$messageContext** the IMessageContextWrapper • **$appName** the application name • **$requestName** the default request name • **$url** the URL • **$context** indicates the type of request: "WebServicesJAXWS ClientRequest", "WebServicesJAXWS ProviderRequest", "WebServicesJAXWS AsyncRequest"	**appName** the renamed application name **requestName** the renamed request name **url** the renamed URL
WebSphere Portal Server Portal (extending the `org.apache.jetspeed.portlet.Portlet` class)	• **$portletAdapter** PortletAdapter • **$portletRequest** PortletRequest • **$portletResponse** PortletResponse • **$portletName** Portlet name • **$pageTitle** Page title • **$url** URL of the request • **$userid** Request user ID • **$context** "Portal.Portlet"	**portletName** the renamed portlet name **title** the renamed page title **url** the renamed URL **userid** the renamed user ID
WebSphere Portal Server version 6.1, 7, and 8 Portal (implementing the `javax.portlet.Portlet` interface)	• **$portlet** Portlet • **$renderRequest** RenderRequest • **$renderResponse** RenderResponse • **$portletName** Portlet name • **$pageTitle** Page title • **$url** URL of the request • **$userid** Request user ID • **$context** "Portal.Portlet"	**portletName** The renamed portlet name **title** The renamed page title **url** The renamed URL **userid** The renamed user ID

For servlet requests, a larger number of input data symbols is provided.

Table 237. Input data symbol names for servlet requests

Symbol Name	Value Type	Symbol Contents
$context	String	"ServletMethod"
$servlet	javax.servlet.http.HttpServlet	The HttpServlet object associated with the servlet request
$httpServletRequest	javax.servlet.http.HttpServletRequest	The HttpServletRequest object associated with the servlet request
$httpServletResponse	javax.servlet.http.HttpServletResponse	The HttpServletResponse object associated with the servlet request
$appName	java.lang.String	The application name associated with the servlet

Table 237. Input data symbol names for servlet requests (continued)

Symbol Name	Value Type	Symbol Contents
$URL	java.lang.StringBuffer	The URL that the client used to make the request
$RemoteUser	java.lang.String	The login name of the user making this request, if authenticated
$URI	java.lang.String	The part of the request URL from the protocol name up to the query string
$ServletPath	java.lang.String	The part of the request URL that calls the servlet.
$SessionID	javax.servlet.http.HttpSession	The current session associated with this request
$QueryString	java.lang.String	The query string that is contained in the request URL after the path.
$SessionAttribute	java.lang.String	This parameterized symbol returns a session attribute value. It has one parameter, the attribute name (must be a string). For example, `$SessionAttribute("attr1")` returns the value of the attribute named attr1.
$cookie	javax.servlet.http.Cookie	This parameterized symbol returns a named cookie. It has one parameter, the cookie name (must be a string). For example, `$cookie("cookie1")` returns the value of the attribute named cookie1.

Example request mapper definitions

The following examples illustrate usage of the request mapper functionality.

Changing the servlet application name

In this example, the application name in a servlet request is replaced by the URI and the query string.

The *dc_home*`/runtime/changeAppname.xml` file contains the following request mapper definition:

```
<gpe>
   <runtimeConfiguration>
         <requestMapperDefinition type="servlet">
               <selection>
                      <matchCriteria>true</matchCriteria>
                      <mapTo>
                             <key>appName</key>
                             <value>$URI + “.” + $QueryString</value>
                      </mapTo>
               </selection>
         </requestMapperDefinition>
   </runtimeConfiguration>
</gpe>
```

Renaming a data source

In this example, the data source name in an SQL request is changed to a version that a user can understand more easily.

The *dc_home*/`runtime/renameDataSource.xml` file contains the following request mapper definition:

```
<gpe>
 <runtimeConfiguration>
   <requestMapperDefinition type=”sqlStatement”>
      <selection>
         <matchCriteria>$dataSourceName != null</matchCriteria>
         <selection>
           <matchCriteria>$dataSourceName.equals("jdbc/TradeDataSource")
</matchCriteria>
           <mapTo>
             <key>dataSourceName</key>
             <value>"Daytrader Data Source"</value>
           </mapTo>
         </selection>
         <selection>
           <matchCriteria>$dataSourceName.equals("jdbc/LongDataSource")
</matchCriteria>
           <mapTo>
             <key>dataSourceName</key>
             <value>"Long term trader Data Source"</value>
           </mapTo>
         </selection>
      </selection>
   </requestMapperDefinition>
 </runtimeConfiguration>
<gpe>
```

The first <selection> tag ensures that `$dataSourceName` is not null. Then, the second <selection> tag can safely evaluate `$dataSourceName.equals()`.

If the first <selection> tag was not present, and a null `$dataSourceName` was passed, the request mapper would generate an exception. Such an exception might result in missing monitoring information.

To enable this request mapper, the file *dc_home*/`runtime/toolkit_global_custom.properties` contains the following lines:

```
com.ibm.tivoli.itcam.toolkit.ai.enable.sqlrequestmapper=true
am.camtoolkit.gpe.customxml.renameDataSource=renameDataSource.xml
```

Removing sensitive information from an SQL request

In this example, an application includes social security numbers in SQL requests. The request mapper removes the numbers from the version of the request that the user can see.

In the SQL requests, the social security number is listed with the SS column name, `SS = `*`number`*. The request mapper looks for the `SS =` string and removes the nine symbols after it.

The *dc_home*/`runtime/removeSSN.xml` file contains the following request mapper definition:

```
<gpe>
    <runtimeConfiguration>
         <requestMapperDefinition type="sqlStatement">
             <symbolDefinitions>
                 <symbol>
                     <name>$offsetOfSS</name>
                     <eval>$sqlText.indexOf("SS = ")</eval>
                 </symbol>
                 <symbol>
                     <name>$sqlTextContainsSS</name>
<eval>$sqlText != null AND $offsetOfSS > 0 AND $sqlText.length() GE
$offsetOfSS+16</eval>
                 </symbol>
                 <conditionalSymbol>
                     <name>$sqlTextPriorToSSKeyword</name>
                     <type>java.lang.String</type>
                     <defaultValue>""</defaultValue>
                     <if condition="$sqlTextContainsSS">
                       <return>$sqlText.substring(0, $offsetOfSS+5)</return>
                     </if>
                 </conditionalSymbol>
                 <conditionalSymbol>
                     <name>$sqlTextAfterSS</name>
                     <type>java.lang.String</type>
                     <defaultValue>""</defaultValue>
                     <if condition="$sqlTextContainsSS">
                       <return>$sqlText.substring($offsetOfSS+16)</return>
                     </if>
                 </conditionalSymbol>
             </symbolDefinitions>
             <selection>
                  <matchCriteria>$sqlText != null AND $sqlText.length() >
0</matchCriteria>
                  <selection>
                       <matchCriteria>$sqlTextContainsSS</matchCriteria>
                       <mapTo>
                            <key>sqlText</key>
                            <value>$sqlTextPriorToSSKeyword + "?" +
$sqlTextAfterSS</value>
                       </mapTo>
                  </selection>
             </selection>
         </requestMapperDefinition>
    </runtimeConfiguration>
</gpe>
```

To enable this request mapper, the file *dc_home*/runtime/toolkit_global_custom.properties contains the following lines:

```
com.ibm.tivoli.itcam.toolkit.ai.enable.sqlrequestmapper=true
am.camtoolkit.gpe.customxml.renameDataSource=removeSSN.xml
```

Configuring the WebSphere Applications agent to monitor WebSphere Extreme Scale

After you install WebSphere Applications agent, you can make extra configuration to monitor WebSphere Extreme Scale (WXS) in a stand-alone or WebSphere Application Server environment.

About this task

Configuration steps are different depending on the installation mode of WebSphere Extreme Scale and whether security is enabled. Do the following steps before you run the configuration process.

Procedure

1. Confirm the installation mode of WebSphere Extreme Scale.

   Stand-alone mode

   WebSphere Extreme Scale is installed in an environment that does not have WebSphere Application Server.

   Embedded WAS mode

   WebSphere Extreme Scale is installed in a WebSphere Application Server environment.
2. Confirm whether the security is enabled for WebSphere Extreme Scale. If a secure Java™ client is used in embedded WebSphere Application Server mode, you must complete security connection steps.
3. Click the links for instructions.
   - To configure WebSphere Extreme Scale in stand-alone environment, click "Configuring WebSphere Extreme Scale monitoring in stand-alone environment" on page 903.
   - To configure WebSphere Extreme Scale in embedded environment without security, click "Configuring WebSphere Extreme Scale monitoring in WebSphere environment without security enabled" on page 904.
   - To configure WebSphere Extreme Scale in embedded environment with security enabled, click "Configuring WebSphere Extreme Scale monitoring in security-enabled WebSphere environment" on page 905.

Configuring WebSphere Extreme Scale monitoring in stand-alone environment

Learn how to configure the WebSphere Applications agent when your WebSphere Extreme Scale is installed in an environment that does not have WebSphere Application Server.

Procedure

1. Stop the WebSphere Applications agent.
   a) Go to the directory *install_dir*where you install the WebSphere Applications agent.
   b) Run command `bin/was-agent.sh stop`.
2. Run the configuration script.

   ```
   install_dir/platform_code/yn/bin/wxs-agent-config.sh config
   ```

   Where

   - *install_dir* is the installation directory of the WebSphere Applications agent.
   - *platform_code* is the platform code where you install the agent, for example, lx8266 represents Linux x86_64 R2.6 (64 bit), aix536 represents AIX R5.3 (64 bit).

   Example command:

   ```
   /opt/ibm/apm/agent/lx8266/yn/bin/wxs-agent-config.sh config
   ```

   ```
   /opt/ibm/apm/agent/aix536/yn/bin/wxs-agent-config.sh config
   ```

3. When prompted for the agent installation path, specify the home directory of WebSphere Applications agent.

   Note: The script looks for the configuration file name based on the installation path you specify. The default is *install_dir*`/config/${hostname}_yn.xml`. If prompted that the file does not exist, it might be because you did not start WebSphere Applications agent before you do this configuration. Start the WebSphere Applications agent and then stop it at least once.
4. When prompted for `WebSphere Extreme Scale Catalog Server connector type`, enter 1 to continue.
5. When prompted to `Input a node name to identify this agent node on UI`, enter the node name.

   The node name is used to identify the monitored WebSphere Extreme Scale zone and it is displayed in the instance name that you can see in the Application Performance Dashboard UI.
6. When prompted for `WebSphere Extreme Scale Catalog Server security enabled?`, enter 1 if there is enabled security. Then, enter the user name and password. If there is no enabled security, enter 2.
7. Specify the host name and port number of the catalog server. If there are multiple catalog servers, you can add them one by one. You can also add multiple zones one after another.
   - The host name is the name of the system where the catalog server is located. Make sure the host name can be accessed. If not, use the IP address as the host name.
   - The port number is the **`JMXServicePort`** number of the WebSphere Extreme Scale catalog server. The default value is 1099. More details about the port number can be found in WebSphere Extreme Scale Knowledge Center.
8. To start the agent, run the following command.

   ```
   install_dir/bin/was-agent.sh start
   ```

   Note:

   - Agent configuration is stored in *install_dir*`/config/${hostname}_yn.xml`. If you want to change any configuration, run this script again or modify the `.xml` file directly.

- Previous configuration is backed up as `install_dir/config/${hostname}_yn.xml.bak`. You can restore previous configuration if necessary.
- You can press Ctrl-C to exit the script when you run `install_dir/platform_code/yn/bin/wxs-agent-config.sh config`. Your existing configuration is not changed.

Configuring WebSphere Extreme Scale monitoring in embedded WebSphere environment

Learn how to configure the WebSphere Applications agent when your WebSphere Extreme Scale is installed in a WebSphere Application Server environment.

About this task

If security is not enabled for the WebSphere Extreme Scale server, you can directly run the configuration process. Otherwise, you must first complete "Configuring WebSphere Extreme Scale monitoring in security-enabled WebSphere environment" on page 905.

Configuring WebSphere Extreme Scale monitoring in WebSphere environment without security enabled

If you install WebSphere Extreme Scale in a WebSphere Application Server environment without security enabled, you can directly configure the WebSphere Applications agent.

Procedure

1. Stop the WebSphere Applications agent.
   a) Go to the directory *install_dir*where you install the WebSphere Applications agent.
   b) Run command `bin/was-agent.sh stop`.
2. Run the configuration script.

   ```
   install_dir/platform_code/yn/bin/wxs-agent-config.sh config
   ```

   Where

   - *install_dir* is the installation directory of the WebSphere Applications agent.
   - *platform_code* is the platform code where you install the agent, for example, lx8266 represents Linux x86_64 R2.6 (64 bit), aix536 represents AIX R5.3 (64 bit).

   Example command:

   ```
   /opt/ibm/apm/agent/lx8266/yn/bin/wxs-agent-config.sh
   ```

   ```
   /opt/ibm/apm/agent/aix536/yn/bin/wxs-agent-config.sh
   ```

3. When prompted for the agent installation path, specify the home directory of WebSphere Applications agent.

   Note: The script looks for the configuration file name based on the installation path you specify. The default is `install_dir/config/${hostname}_yn.xml`. If prompted that the file does not exist, it might be because you did not start WebSphere Applications agent before you do this configuration. Start the WebSphere Applications agent and then stop it at least once.
4. When prompted for `WebSphere Extreme Scale Catalog Server connector type`, enter 2 to continue.
5. When prompted to `Input a node name to identify this agent node on UI`, enter the node name.

   The node name is used to identify the monitored WebSphere Extreme Scale zone and it is displayed in the instance name that you can see in the Application Performance Dashboard UI.
6. When prompted for `WebSphere Extreme Scale Catalog Server security enabled?`, enter 2 to continue.
7. Specify the host name and port number of the catalog server. If there are multiple catalog servers, you can add them one by one. You can also add multiple zones one after another.

- The host name is the name of the system where the catalog server is located. Make sure the host name can be accessed. If not, use the IP address as the host name.
- The port number indicates the **JMXServicePort** number of the WebSphere Extreme Scale catalog server. It is inherited from the **BOOTSTRAP_ADDRESS** value for each WebSphere Application Server. More details about the port number can be found in WebSphere Extreme Scale Knowledge Center.

8. To start the agent, run the following command.

   ```
   install_dir/bin/was-agent.sh start
   ```

   Note:

   - Agent configuration is stored in `install_dir/config/${hostname}_yn.xml`. If you want to change any configuration, run this script again or modify the `.xml` file directly.
   - Previous configuration is backed up as `install_dir/config/${hostname}_yn.xml.bak`. You can restore previous configuration if necessary.
   - You can press Ctrl-C to exit the script when you run `install_dir/platform_code/yn/bin/wxs-agent-config.sh config`. Your existing configuration is not changed.

Configuring WebSphere Extreme Scale monitoring in security-enabled WebSphere environment

If you install WebSphere Extreme Scale in a WebSphere Application Server environment with security enabled, you must complete initial setup steps before you configure the WebSphere Applications agent.

About this task

If you want to monitor WebSphere Extreme Scale servers in WebSphere Application Server security enabled environments, it is necessary to configure security settings manually.

The procedure applies to the following case:

- WebSphere Extreme Scale servers must be deployed inside the WebSphere Application Server application servers(or node agent or DMGR processes).
- WebSphere Applications agent must be deployed on a node where a WebSphere Extreme Scale zone catalog service is running. Configure the Agent for WebSphere Extreme Scale monitoring under this node, and set it to connect to this catalog service instance.
- One Agent instance must be used to monitor only one WebSphere Extreme Scale zone.

Procedure

1. If the JDK version of your WebSphere Application Server is prior to 1.7, you must reconfigure the WebSphere Applications agent to use the same JRE as WebSphere Application Server.

   a) Open the `install_dir/config/.yn.environment` file.

   b) Add the following value to the first line.

   ```
   #JAVAHOME=/opt/IBM/WebSphere/AppServer/java/8.0/jre
   ```

2. Configure the WebSphere Applications agent security properties file.

   For instructions, see "Configuring the agent to work with JAR files and security properties of WebSphere Application Server" on page 906.

3. Optional: If a secure Java™ client is used, you must ensure that the authentication is properly configured. You must edit the client properties file and SSL properties file. For instructions, see "Setting up connection credentials" on page 907.

   Note: If the key is not secured by SSL settings, you need to enter a password and user name only, then you can skip this step.

4. Run the configuration script to launch the configuration console. See "Running the configuration" on page 909.

Configuring the agent to work with JAR files and security properties of WebSphere Application Server
Configure the WebSphere Applications agent to work with WebSphere Application Server JAR files and security properties.

About this task

To do this configuration, edit the `kynwb.properties` file.

Procedure

1. Open the file *`install_dir`*`/`*`platform_code`*`/yn/config/kynwb.properties`. If this file does not exist, create one.
   - *install_dir* is the installation directory of the WebSphere Applications agent.
   - *platform_code* is the platform code where you install the agent, for example, lx8266 represents Linux x86_64 R2.6 (64 bit), aix536 represents AIX R5.3 (64 bit).
2. In the beginning of the file, the class path is listed. Add the following lines before the existing lines.
   - For WebSphere Application Server 9.0:

     ```
     appserver_home/plugins/com.ibm.ws.runtime.jar:\
     appserver_home/lib/bootstrap.jar:\
     appserver_home/runtimes/com.ibm.ws.admin.client_9.0.jar:\
     appserver_home/lib/wsogclient.jar:\
     ```

   - For WebSphere Application Server 8.5:

     ```
     appserver_home/plugins/com.ibm.ws.runtime.jar:\
     appserver_home/lib/bootstrap.jar:\
     appserver_home/runtimes/com.ibm.ws.admin.client_8.5.0.jar:\
     appserver_home/lib/wsogclient.jar:\
     ```

   Example of a modified class path:

   ```
   /opt/IBM/WebSphere/plugins/com.ibm.ws.runtime.jar:\
   /opt/IBM/WebSphere/lib/bootstrap.jar:\
   /opt/IBM/WebSphere/runtimes/com.ibm.ws.admin.client_8.5.0.jar:\
   /opt/IBM/WebSphere/lib/wsogclient.jar:\
   lib/kynwb.jar:\
   lib/kynwxssec_api.jar:\
   lib/itcam.cg.mbean.jar:\
   wasdc/7.3/installer/lib/itcamfwas.jar:\
   ```

3. At the end of the *`install_dir`*`/`*`platform_code`*`/yn/config/kynwb.properties` file, add the lines that indicate the security property files for use by the agent. Typically, these files are the ones that are used by the wsadmin utility:

   ```
   -Dcom.ibm.CORBA.ConfigURL=file:/appserver_profile/properties/sas.client.props
   -Dcom.ibm.SSL.ConfigURL=file:/appserver_profile/properties/ssl.client.props
   ```

   If the agent required security settings are different from those settings that are used by the wsadmin utility, create separate copies of the files and provide paths to them instead, for example:

   ```
   -Dcom.ibm.CORBA.ConfigURL=file:/opt/IBM/ITM/config/sas.client.props
   -Dcom.ibm.SSL.ConfigURL=file:/opt/IBM/ITM/config/ssl.client.props
   ```

   Note: When you install a fix pack or interim fix for WebSphere Applications agent, the changes that are made in the `yn.ini` and `kynwb.properties` files are overwritten. Therefore, after you install a fix pack or interim fix, you must complete the changes in these two files again.

Setting up connection credentials

When a secure Java client is used, it needs to read a properties file that contains a list of CSIv2 settings. These settings determine how the client authenticates to a server. You must ensure that the authentication is properly configured.

About this task

Typically, the file with these settings is specified in the `com.ibm.CORBA.ConfigURL` JVM property. More SSL settings can be found in file specified in the `com.ibm.SSL.ConfigURL` JVM property.

When WebSphere Applications agent is configured to monitor eXtreme Scale servers embedded in WebSphere Application Server, it acts as secure Java client. For that reason, `-Dcom.ibm.CORBA.ConfigURL` and `-Dcom.ibm.SSL.ConfigURL` must be specified in the `kynwb.properties` file.

In most cases, these properties point to the `sas.client.props` and `ssl.client.props` files in the *`appserver_profile`*`/properties` directory. These files are used by tools like `wsadmin` or `xscmd`. Therefore, if you can use one of these tools to connect to an Extreme Scale catalog server without the need to enter any credentials, you do not need to customize the settings.

If the connection fails or requires entering a username or password, you must complete extra configuration.

Modifying client properties file

Edit the `sas.client.props` file to be used by the WebSphere Applications agent.

About this task

The full path to the `sas.client.props` file is specified in `kynwb.properties`, in the `-Dcom.ibm.CORBA.ConfigURL` property. Provide the connection and security information for the WebSphere Application Server instance running the Catalog Services instance for which the agent is configured.

Procedure

1. Open the *`appserver_profile`*`/properties/sas.client.props` file.
2. Change the value of the `com.ibm.CORBA.loginSource` property to `properties`:

   ```
   com.ibm.CORBA.loginSource=properties
   ```

3. Set the property `com.ibm.CORBA.securityServerHost` to the host name of an application server within the WebSphere Extreme Scale zone. The server can be the local server or a different one. The server must be always available when the agent starts up. For example:

   ```
   com.ibm.CORBA.securityServerHost=server.company.com
   ```

4. Set the `com.ibm.CORBA.securityServerPort` property to the RMI port for the application server profile, for example:

   ```
   com.ibm.CORBA.securityServerPort=2819
   ```

5. Set the property `com.ibm.CORBA.loginUserid` to the login name for communicating with the application server, and the property `com.ibm.CORBA.loginPassword` to the password, for example:

   ```
   com.ibm.CORBA.loginUserid=admin
   com.ibm.CORBA.loginPassword=password
   ```

6. Set the following properties to `true` or `false`, corresponding to the **CSIv2 inbound communications** settings in the WebSphere administrative console:

   ```
   com.ibm.CSI.performTLClientAuthenticationRequired
   com.ibm.CSI.performTLClientAuthenticationSupported
   ```

```
com.ibm.CSI.performTransportAssocSSLTLSRequired
com.ibm.CSI.performTransportAssocSSLTLSSupported
```

The `com.ibm.CSI.performTLClientAuthentication*` properties are related to **Client certificate authentication** settings. The `com.ibm.CSI.performTransportAssocSSLTLS*` are related to **Transport** settings.

7. Optional: If the default SSL alias (`DefaultSSLSettings`) is not used, set the SSL configuration alias name in the `com.ibm.ssl.alias` property.
8. Save the file, and then encrypt the password within the `sas.client.props` file. To encrypt the password, run the following command:

   - On Linux and UNIX systems, run *appserver_profile*`/bin/PropFilePasswordEncoder.sh sas.client.props com.ibm.CORBA.loginPassword`

   Important: When client certificate authentication is required and basic authentication is enabled, you might also need to set the property `com.ibm.CORBA.validateBasicAuth=false`.

Modifying client SSL properties file
Modify the SSL properties file that the WebSphere Applications agent uses to access server certificates.

About this task

Edit the `ssl.client.props` file to be used by the agent. The full path to the file is specified in the `kynwb.properties` file, in the `-Dcom.ibm.SSL.ConfigURL` property. Provide the SSL truststore and keystore information for the WebSphere Application Server instance running the Catalog Services instance for which the agent is configured.

You can create and manage certificates by using the WebSphere administrative console (**Security** > **SSL certificate and key management** > **Key stores and certificates**) or by using the iKeyman tool.

Procedure

1. Open the *appserver_profile*`/properties/ssl.client.props` file.
2. Change the value of the `com.ibm.ssl.alias` property to match the value of the same property in the `sas.client.props` file.

   Tip: The `ssl.client.props` file can contain several SSL configurations. Each configuration starts with the `com.ibm.ssl.alias` property.
3. Set the `com.ibm.ssl.enableSignerExchangePrompt` property to `false`.
4. Set the following keystore properties to enable the client application to access the encryption key:

   com.ibm.ssl.keyStoreName
   The name that identifies this keystore

   com.ibm.ssl.keyStore
   The full path and name of the keystore file

   com.ibm.ssl.keyStorePassword
   The password for the keystore

   com.ibm.ssl.keyStoreType
   The keystore type. Use the default PKCS12 type because of its interoperability with other applications.

   Important: If client certificate authentication is not required, the keystore can contain any self-signed key. Otherwise, the keystore must contain a key that is signed by a certificate that is in the server truststore.
5. Set the following truststore properties to enable the client application to access signer certificates:

   com.ibm.ssl.trustStoreName
   The name that identifies this truststore

   com.ibm.ssl.trustStore
   The full path and name of the truststore file

com.ibm.ssl.trustStorePassword
The password for the truststore

com.ibm.ssl.trustStoreType
The truststore type. Use the default PKCS12 type because of its interoperability with other applications.

Important: If the client is to use an SSL connection, the server signer certificate must be in its truststore.

Running the configuration
After you check the environment and security, you can run the configuration process.

Procedure

1. Stop the WebSphere Applications agent.

   a) Go to the directory *install_dir*where you install the WebSphere Applications agent.

   b) Run command `bin/was-agent.sh stop`.
2. Run the configuration script.

   ```
   install_dir/platform_code/yn/bin/wxs-agent-config.sh config
   ```

   Where

   - *install_dir* is the installation directory of the WebSphere Applications agent.
   - *platform_code* is the platform code where you install the agent, for example, lx8266 represents Linux x86_64 R2.6 (64 bit), aix536 represents AIX R5.3 (64 bit).

   Example command:

   ```
   /opt/ibm/apm/agent/lx8266/yn/bin/wxs-agent-config.sh config
   ```

   ```
   /opt/ibm/apm/agent/aix536/yn/bin/wxs-agent-config.sh config
   ```
3. When prompted for the agent installation path, specify the home directory of WebSphere Applications agent.

   Note: The script looks for the configuration file name based on the installation path you specify. The default is `install_dir/config/${hostname}_yn.xml`. If prompted that the file does not exist, it might be because you did not start WebSphere Applications agent before you do this configuration. Start the WebSphere Applications agent and then stop it at least once.
4. When prompted for `WebSphere Extreme Scale Catalog Server connector type`, enter 2 to continue.
5. When prompted to `Input a node name to identify this agent node on UI`, enter the node name.

   The node name is used to identify the monitored WebSphere Extreme Scale zone and it is displayed in the instance name that you can see in the Application Performance Dashboard UI.
6. When prompted for `WebSphere Extreme Scale Catalog Server security enabled?`, enter 1 to continue. Then, enter the user name and password.
7. Specify the host name and port number of the catalog server. If there are multiple catalog servers, you can add them one by one. You can also add multiple zones one after another.

   - The host name is the name of the system where the catalog server is located. Make sure the host name can be accessed. If not, use the IP address as the host name.
   - The port number indicates the **`JMXServicePort`** number of the WebSphere Extreme Scale catalog server. It is inherited from the **`BOOTSTRAP_ADDRESS`** value for each WebSphere Application Server. More details about the port number can be found in WebSphere Extreme Scale Knowledge Center.
8. To start the agent, run the following command.

```
install_dir/bin/was-agent.sh start
```

Note:

- Agent configuration is stored in *install_dir*/config/${hostname}_yn.xml. If you want to change any configuration, run this script again or modify the .xml file directly.
- Previous configuration is backed up as *install_dir*/config/${hostname}_yn.xml.bak. You can restore previous configuration if necessary.
- You can press Ctrl-C to exit the script when you run *install_dir*/bin/wxs-agent-config.sh config. Your existing configuration is not changed.

Unconfiguring the WebSphere Extreme Scale monitoring
When you do not want to monitor WebSphere Extreme Scale, you can unconfigure the WebSphere Applications agent.

Procedure

1. Stop the WebSphere Applications agent.
   a) Go to the directory *install_dir*where you install the WebSphere Applications agent.
   b) Run command bin/was-agent.sh stop.
2. Run the unconfiguration script.

   ```
   install_dir/{pc}/yn/bin/wxs-agent-config.sh unconfig
   ```

   Where

   - *install_dir* is the installation directory of the WebSphere Applications agent.
   - *platform_code* is the platform code where you install the agent, for example, lx8266 represents Linux x86_64 R2.6 (64 bit), aix536 represents AIX R5.3 (64 bit).

   Example command:

   ```
   /opt/ibm/apm/agent/lx8266/yn/bin/wxs-agent-config.sh unconfig
   ```

   ```
   /opt/ibm/apm/agent/aix536/yn/bin/wxs-agent-config.sh unconfig
   ```

Configuring WebSphere Infrastructure Manager monitoring

Configure the WebSphere Infrastructure Manager agent to monitor the performance of WebSphere Deployment Manager and Node Agent.

About this task

The WebSphere Infrastructure Manager agent is a multiple instance agent. You must create the first instance and start the agent manually.

Procedure

1. To configure the agent, run the following command.

   ```
   install_dir/bin/wim-agent.sh config instance_name
   ```

   Where *instance_name* is the name you want to give to the instance, and *install_dir* is the installation directory of WebSphere Infrastructure Manager agent. The default installation directory is /opt/ibm/apm/agent.
2. When prompted to Edit 'Monitoring Agent for WebSphere Infrastructure Manager' settings, enter 1 to continue.
3. When prompted for Java home, specify the directory where Java is installed.

The default value is `/opt/ibm/apm/agent/JRE/lx8266/jre`.

4. When prompted for `DMGR Profile Home`, specify the home directory of the Deployment Manager profile.

   The default directory is `/opt/IBM/WebSphere/AppServer/profiles/Dmgr01`.
5. When prompted for `JMX user ID`, specify the user ID that is used to connect to the MBean server.
6. When prompted to `Enter JMX password`, specify the password for the user.
7. When prompted to `Re-type JMX password`, enter the password again.
8. To start the agent, run the following command.

```
install_dir/bin/wim-agent.sh start instance_name
```

Results

You created a WebSphere Infrastructure Manager agent instance and started the monitoring agent to begin collecting data samples for resource monitoring.

Configuring WebSphere MQ monitoring

Before you can start the agent, you must assign an instance name to the agent and complete the several configuration tasks for the user ID and managed system names. Optionally you can also enable transaction tracking for the agent.

Before you begin

- The directions here are for the most current release of this agent. For information about how to check the version of an agent in your environment, see Agent version command. For detailed information about the agent version list and what's new for each version, see "Change history" on page 54.
- Make sure that the system requirements for the WebSphere MQ agent are met in your environment. For the up-to-date system requirement information, see the Detailed System Requirements Report for the WebSphere MQ agent.

About this task

The directions are for the most current release of the agent, except as indicated.

To set up the environment for the WebSphere MQ agent, you must first make sure the agent user ID can access IBM MQ (WebSphere MQ) objects, configure IBM MQ (WebSphere MQ) for data enablement, and then configure the WebSphere MQ agent.

The following procedure is a roadmap for configuring WebSphere MQ agent, which includes both required and optional steps. Complete the necessary steps according to your needs.

Procedure

1. Authorize the user ID that is used to configure, start, and stop the agent to access IBM MQ (WebSphere MQ) objects. See "Authorizing the user IDs to run the agent" on page 912.
2. Configure IBM MQ (WebSphere MQ) to enable the data that you want to monitor. See "Configuring IBM MQ (WebSphere MQ) for data enablement" on page 913.
3. Configure the agent by providing an agent instance name, a queue manager name, and optionally an agent name. See "Configuring the WebSphere MQ agent" on page 915.
4. Optional: Depending on your monitoring requirements, you might require a unique managed system name to distinguish different monitoring agents. Use the `Agent Name` option in the `mq-agent.sh config` command to specify the middle qualifier of the managed system name. See "Specifying unique managed system names for multiple queue managers" on page 918.
5. Optional: To configure the agent to collect transaction tracking data of the monitored queue manager, use the **Agent Configuration** page. For instructions, see "Configuring transaction tracking for the WebSphere MQ agent" on page 920.

6. Optional: Enable the agent to collect the long-term history data for queues and channels. For instructions, see "Enabling data collection for queue and channel long-term history" on page 921.
7. Optional: To remotely monitor queue manager on MQ Appliance, additional configuration of both the agent and IBM MQ (WebSphere MQ) is required. For instructions, see "Remotely monitoring queue managers on MQ Appliance" on page 922 or "Remotely monitoring HA queue managers on MQ Appliance" on page 923.

Authorizing the user IDs to run the agent

For a user ID to configure, start, and stop the WebSphere MQ agent, the user ID must belong to the **mqm** group, which has full administrative privileges over IBM MQ (WebSphere MQ). Also, for a non-root user or a non-administrator user, you must grant users the access to the IBM MQ (WebSphere MQ) objects by using the IBM MQ (WebSphere MQ) control command.

About this task

On AIX or Linux system, you must add the user ID to the **mqm** group and then grant the user ID appropriate access to the IBM MQ (WebSphere MQ) objects with the **`setmqaut`** command.

On Windows systems, you must add the user ID to the **mqm** group. If the user ID does not belong to the Administrator user group, you must also use the Registry Editor to grant permissions to the user ID to start or stop the agent.

Procedure

- Linux AIX

  On AIX or Linux system, complete the following steps:

  a) Log on to the AIX or Linux system by using the root ID.

  b) Add the user ID that is used to run the agent to the **mqm** group.

  c) (WebSphere MQ V7.5 or later): If the user ID is a non-root user on the AIX or Linux system, set the appropriate level of authority for the user ID to access the IBM MQ (WebSphere MQ) objects by running the following command:

  ```
  setmqaut -m queue_manager -t qmgr -p user_ID +inq +connect +dsp +setid
  ```

  where *queue_manager* is the name of the queue manager of WebSphere MQ V7.5 or later and *user_ID* is the non-root or non-administrator user ID to run the agent.

- Windows

  On Windows systems, complete the following steps:

  a) Log on to the Windows systems as a system administrator.

  b) Add the user ID that is used to run the agent to the **mqm** group.

  c) If the user ID that you use to start, run, and stop the agent is not a member of the Administrators group, use the Registry Editor to set permissions for a user ID to ensure that the agent can be started and stopped successfully:

     a. Click **Start** > **Run**, and then type `regedit.exe` to open the Registry Editor.
     b. In the Registry Editor, locate the key, `HKEY_LOCAL_MACHINE\SOFTWARE\Candle`.
     c. Right-click the key and click **Permissions**.
     d. If the user ID for the WebSphere MQ agent is not in the Group or user names list, click **Add** to add the user ID to the list.
     e. Click the user ID in the list.
     f. In the Permissions for the *user-ID* list, where *user-ID* is the user ID of WebSphere MQ agent, select **Full Control** in the Allow column and click **OK**.
     g. In the Registry Editor, locate the key, `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib`.

h. Right-click the key and click **Permissions**.
i. If the user ID for the WebSphere MQ agent is not in the Group or user names list, click **Add** to add the user ID to the list.
j. Click the user ID in the Group or user names list.
k. In the Permissions for the *user-ID* list, where *user-ID* is the user ID of WebSphere MQ agent, select **Read** in the Allow column and click **OK**.
l. Close the Registry Editor.
m. Locate the *install_dir* directory, where *install_dir* is the agent installation directory.
n. Right-click the directory and click **Properties**.
o. On the Security tab, if the user ID for WebSphere MQ agent is not in the Group or user names list, click **Edit** and then **Add** to add the user ID to the list.
p. Click the user ID in the Group or user names list.
q. In the Permissions for the *user-ID* list, select **Full Control** in the Allow column, where **user-ID** is the user ID of WebSphere MQ agent.
r. Click **OK**.

What to do next

The next step to configure IBM MQ (WebSphere MQ) for data enablement. See "Configuring IBM MQ (WebSphere MQ) for data enablement" on page 913.

Configuring IBM MQ (WebSphere MQ) for data enablement

Before you configure the WebSphere MQ agent, it is recommended to the configure IBM MQ (WebSphere MQ) first to enable the data that you want to monitor.

About this task

Decide what type of data that you want the WebSphere MQ agent to monitor. Enable the data at the queue manager by using the MQSC commands if the data is not produced by the queue manager by default.

Remember: You must start MQSC for the target queue manager before you issue the MQSC commands. To get a list of the queue manager, issue the **dspmq** command from the `bin` directory within the IBM MQ (WebSphere MQ) installation directory. To start MQSC for a queue manager, issue the following command from the `bin` directory, where *<qmgr_name>* is the name of the queue manager that you want to configure.

```
runmqsc <qmgr_name>
```

Procedure

- To see the age of the oldest message on a queue, complete the steps as documented in "Enabling real-time monitoring for queues" on page 914.
- To monitor certain queue manager events that are not generated by the queue manager by default, complete the steps as documented in "Enabling event monitoring for the queue manager" on page 914.
- To get the transaction tracking data, complete the steps as documented in "Enabling MQI application activity trace" on page 915.
- To monitor a remote queue manager, ensure that the WebSphere MQ agent can collect monitoring data through a channel on the remote system. For more information, see "Security settings for remote monitoring" on page 915.

Enabling real-time monitoring for queues

About this task

To see the age of the oldest message (in seconds) on a queue, you must enable the real-time monitoring for the queue.

Procedure

Use the following commands to enable real-time monitoring for the queues in your environment.

- To enable real-time monitoring for all the queues whose MONQ attribute is set to QMGR, issue the following command:

  ```
  ALTER QMGR MONQ(collection_level)
  ```

  where *collection_level* specifies the collection level of monitoring data for the queues. You can set it to LOW, MEDIUM, or HIGH to suit the requirements of your environment.

- To enable real-time monitoring for individual queue, issue the following command:

  ```
  ALTER QLOCAL(queue_name) MONQ(collection_level)
  ```

  where *queue_name* is the name of the queue; *collection_level* specifies the collection level of monitoring data for the queues. You can set it to LOW, MEDIUM, or HIGH to suit the requirements of your environment.

Results

The data can be displayed in the Oldest Message Age for Queue group widget after the WebSphere MQ agent is started.

Enabling event monitoring for the queue manager

About this task

Event monitoring is one of the monitoring techniques that are available to monitor your IBM MQ network. After you enable the queue manager to emit certain types of events, event messages are put on event queues when the event occurs. So that these event messages can be monitored and displayed by the WebSphere MQ agent.

The following types of events are not monitored and displayed with the default queue manager configuration. Use the **`ALTER QMGR`** command to enable the queue manager to generate these events so that they can be displayed on the Application Performance Dashboard.

- Channel events
- Performance events

Procedure

Use the following commands to enable the queue manager to generate the events that you care:

- To generate channel events, issue `ALTER QMGR CHLEV(ENABLED)`.
- To generate performance events, issue `ALTER QMGR PERFMEV(ENABLED)`.

Results

The monitored events can be displayed in the Queue Manager Events group widget after the WebSphere MQ agent is started.

Enabling MQI application activity trace

About this task

For transaction tracking data to be displayed in the middleware and topology dashboards, the MQI application activity trace must be enabled at the queue manager.

Procedure

- To enable MQI application activity trace information collection, issue the following MQSC command:

  ```
  ALTER QMGR ACTVTRC(ON)
  ```

Security settings for remote monitoring

About this task

To use the WebSphere MQ agent to monitor a remote queue manager, you must make sure that the security settings of IBM MQ (WebSphere MQ) does not prevent the agent from collecting monitoring data through a channel on the remote system.

The following procedure provides an example of a simple security setting for remote monitoring. To exercise more precise control over the access granted to connecting systems at a channel level, you can use channel authentication records. For more information, refer to IBM MQ security mechanisms documentation.

Procedure

1. Disable channel authentication by running the following MQSC command:

   ```
   ALTER QMGR CHLAUTH(DISABLED) CONNAUTH(' ')
   ```

2. Change the channel settings as follows, where *channel_for_remote_monitor* is the name of the channel used for remote monitoring.

   - Linux AIX

     ```
     ALTER CHANNEL(channel_for_remote_monitor) CHLTYPE(SVRCONN) MCAUSER('mqm')
     ```

   - Windows

     ```
     ALTER CHANNEL(channel_for_remote_monitor) CHLTYPE(SVRCONN) MCAUSER(MUSR_MQADMIN)
     ```

3. Refresh the security settings.

   ```
   REFRESH SECURITY
   ```

Configuring the WebSphere MQ agent

You must assign an instance name to the WebSphere MQ agent and configure the agent before it can start monitoring your IBM MQ (WebSphere MQ) environment.

Before you begin

- Make sure that the agent user ID has appropriate permission to access IBM MQ (WebSphere MQ) objects. If you have not done it, follow the instructions in “Authorizing the user IDs to run the agent” on page 912.
- Configure IBM MQ (WebSphere MQ) to enable the required data collection. If you have not done it, see “Configuring IBM MQ (WebSphere MQ) for data enablement” on page 913.
- You must provide the name of queue manager to be monitored by the WebSphere MQ agent. Contact the IBM MQ (WebSphere MQ) administrator if you do not know the appropriate queue manager name. Alternatively, issue the **dspmq** command from the `bin` directory within the IBM MQ (WebSphere MQ)

installation directory to get a list of the queue managers. The returned QMNAME value is what you must provide when you configure the WebSphere MQ agent.

About this task

The WebSphere MQ agent is a multiple instance agent; you must create the first instance and manually start the agent.

On UNIX or Linux systems, you can choose to configure the agent with or without interactions. On Windows systems, you can configure the agent without interactions only.

- To configure the agent with interaction, run the configuration script and respond to prompts. See “Interactive configuration” on page 916.
- To configure the agent without interaction, edit the silent response file and then run the configuration script. See “Silent configuration” on page 917.

Important: If you also installed Monitoring Agent for WebSphere MQ, which is delivered as one component of the ITCAM for Applications product, on the same system as the WebSphere MQ agent, which is delivered in Cloud APM, do not use them to monitor the same queue manager on the system.

Interactive configuration

Procedure

To configure the agent by running the script and responding to prompts, complete the following steps:

1. Enter the following command to create an agent instance:

   ```
   install_dir/bin/mq-agent.sh config instance_name
   ```

   where *instance_name* is the name you want to give to the instance.
2. When prompted for `Queue Manager Name`, specify the name of the queue manager to be monitored.
3. When prompted for `Agent Name`, specify the agent name to be used as the middle qualifier of the managed system name. Do not press Enter to skip specifying this parameter.

   Remember: This agent name is different from the agent instance name. The agent instance name is used in the agent configuration file name to distinguish the configuration files between agents, for example, *`hostname`*`_mq_`*`instancename`*`.cfg`. The agent name is used as a short identifier to create unique managed system names. To understand when a unique managed system name is required, see “Specifying unique managed system names for multiple queue managers” on page 918.
4. If you want to monitor a remote queue manager, specify the following configuration parameters. If you want to monitor a local queue manager, press Enter to proceed.

   - `Connection Name`: The connection name for remote monitoring. The format is *`IP_address`*`(`*`port_number`*`)`, for example, `127.0.0.1(1414)`. If this is the first time you configure the agent instance, you can press Enter to accept the default of null. The appropriate connection name can be automatically discovered.
   - `Channel`: The name of channel used for remote data collection. If this is the first time you configure the agent instance, you can press Enter to accept the default of null. The SYSTEM.DEF.SVRCONN channel will be used.

   Limitation: Error logs of a remote queue manager cannot be monitored. When the agent is monitoring a remote queue manager, the MQ Errors Details dashboard contains no data.
5. When prompted for WebSphere MQ library path, press Enter to accept the default value, which is the 64-bit library path of IBM MQ (WebSphere MQ) automatically discovered by the WebSphere MQ agent. If no default value is displayed, you must provide the 64-bit library path of IBM MQ (WebSphere MQ) to proceed.
   An example of the 64-bit library path is `/opt/mqm8/lib64` for a Linux system.
6. To start the agent, enter the following command:

```
install_dir/bin/mq-agent.sh start instance_name
```

Silent configuration

Procedure

To configure the agent by editing the silent response file and running the script without interaction, complete the following steps:

1. Open the `mq_silent_config.txt` file in a text editor.
   - Linux AIX *install_dir*/samples/mq_silent_config.txt
   - Windows *install_dir*\tmaitm6_x64\samples\mq_silent_config.txt

   where *install_dir* is the installation directory of WebSphere MQ agent.
2. Required: For **QMNAME**, specify the name of the queue manager to be monitored.
3. Required: For **AGTNAME**, specify the agent name to be used as the middle qualifier of the managed system name.

   Remember: This agent name is different from the agent instance name. The agent instance name is used in the agent configuration file name to distinguish the configuration files between agents, for example, *hostname*_mq_*instancename*.cfg. The agent name is used as a short identifier to create unique managed system names. To understand when a unique managed system name is required, see "Specifying unique managed system names for multiple queue managers" on page 918.
4. If you want to monitor a remote queue manager, specify the following configuration parameters:
   - **CONNAME**: The connection name for remote monitoring. The format is *IP_address* (*port_number*), for example, 127.0.0.1(1414).
   - **CHANNEL**: The name of channel used for remote data collection. If not specified, the SYSTEM.DEF.SVRCONN channel will be used.

   Limitation: Error logs of a remote queue manager cannot be monitored. When the agent is monitoring a remote queue manager, the MQ Errors Details dashboard contains no data.
5. Optional: For **WMQLIBPATH**, specify the 64-bit library path of IBM MQ (WebSphere MQ). For example, /opt/mqm8/lib64. If no value is specified, the path can be automatically discovered during agent configuration.
6. Save and close the `mq_silent_config.txt` file, and then run the following command from the command line:
   - Linux AIX *install_dir*/bin/mq-agent.sh config *instance_name* *path_to_responsefile*
   - Windows *install_dir*\BIN\mq-agent.bat config *instance_name* "*path_to_responsefile*"

   where *instance_name* is the name of the instance that you configure, and *path_to_responsefile* is the full path of the silent response file.

   Remember: On Windows systems, do not omit the double quotation marks ("") that enclose the path to the silent response file, especially when the path contains special characters.

   For example, if the response file is in the default directory, run the following command.

   - Linux AIX

     ```
     /opt/ibm/apm/agent/bin/mq-agent.sh config instance_name
     /opt/ibm/apm/agent/samples/mq_silent_config.txt
     ```

   - Windows

```
C:\IBM\APM\BIN\mq-agent.bat config instance_name
"C:\IBM\APM\tmaitm6_x64\samples\mq_silent_config.txt"
```

7. To start the agent, enter the following command:

   - Linux AIX

     ```
     install_dir/bin/mq-agent.sh start instance_name
     ```

   - Windows

     ```
     install_dir\bin\mq-agent.bat start instance_name
     ```

Results

Now, you can log in to the Cloud APM console and use the Applications editor to add the WebSphere MQ agent instance to the Application Performance Dashboard. For instructions on how to start the Cloud APM console, see "Starting the Cloud APM console" on page 957. For information about using the Applications editor, see "Managing applications" on page 1073.

What to do next

- If you have enabled MQI application activity trace information collection at the queue manager, use the **Agent Configuration** page to configure the WebSphere MQ agent to collect transaction tracking data of the monitored queue manager. See "Configuring transaction tracking for the WebSphere MQ agent" on page 920. If the agent does not show up on the **Agent Configuration** page, restart the Cloud APM server.
- Depending on your monitoring requirements, you might require a unique managed system name to distinguish different monitoring agents. Use the `Agent Name` option in the `mq-agent.sh config` command to specify the middle qualifier of the managed system name. See "Specifying unique managed system names for multiple queue managers" on page 918.
- To configure the WebSphere MQ agent for remote monitoring, you must do some manual configuration after you create an agent instance. For instructions, see the following topics:
  - "Remotely monitoring queue managers on MQ Appliance" on page 922
  - "Remotely monitoring HA queue managers on MQ Appliance" on page 923

Specifying unique managed system names for multiple queue managers

Unique managed system names are required sometimes to distinguish different monitoring agents that connect to the same Cloud APM server. Use the **AGTNAME** parameter in the silent response file or the `Agent Name` option in the **`mq-agent.sh config`** command to specify the middle qualifier that is used in the managed system name.

About this task

When the WebSphere MQ agent is started, it registers the following managed system:

```
monitoredqueuemanagername:agentname:MQ
```

where

- *monitoredqueuemanagername* is the name of the queue manager that is monitored by the agent.
- *agentname* is the middle qualifier of the managed system name. If the *agentname* value is not specified, no value is used.

Specifying the agent name value is useful in the following circumstances:

- If your site has multiple queue managers with the same name that are running on different nodes, specify the agent name for each queue manager, so that the WebSphere MQ agent can create unique managed system names.

- If the length of the managed system name exceeds 32 characters, 2 different queue manager names might resolve to the same name because of truncation. To distinguish the managed system names for queue managers, specify the agent name for each queue manager.
- If you want to group and identify queue manager names by something other than the host name and queue manager name, such as a high-availability cluster name.
- If you want to enable multiple agents that are connected to the same Cloud APM server to monitor the queue managers with the same name on different hosts.

Interactive configuration

Procedure

To use the `Agent Name` option in the **`mq-agent.sh config`** command, complete the following steps:

1. In the command line, run the following command to start configuring the WebSphere MQ agent.

   ```
   ./mq-agent.sh config instance_name
   ```

   where *instance_name* is the name of the instance that you started.
2. Follow the options to configure the agent instance.

   The name of the queue manager is required. For other options, if no change is needed, use the default value.
3. When the `Agent Name` option appears, specify the middle qualifier for the managed system name.

   Remember: The complete managed system name is *monitoredqueuemanagername*:*agentname*:MQ. The maximum length for the complete managed system name is 32 characters, so the maximum length for the middle qualifier *agentname* depends on the length of the queue manager name. If the value that is specified for the `Agent Name` option exceeds the maximum length, the value for *agentname* is truncated to no less than 8 characters.

   For example, to monitor a queue manager that is named PERSONNEL on the AIX1 node while another queue manager that is named PERSONNEL is on a node that is named LINUX2, run the following command first for the AIX1 node:

   ```
   ./mq-agent.sh config PERSONNEL
   ```

   Then specify the agent name when the *Agent Name* option appears:

   ```
   Agent Name (default is: ): AIX1
   ```

   To simultaneously monitor the PERSONNEL queue manager on the LINUX2 node, run the following command first:

   ```
   ./mq-agent.sh config PERSONNEL
   ```

   Then specify the agent name:

   ```
   Agent Name (default is: ): LINUX2
   ```

   Remember: The names of the agent nodes are used for the `Agent Name` option in the code samples for explanatory purpose only. You can specify other strings for the `Agent Name` option.

Silent configuration

Procedure

To use the **AGTNAME** parameter in the silent response file, complete the following steps:

1. Open the `mq_silent_config.txt` silent response file in a text editor.
2. Specify an agent name for the **AGTNAME** parameter.

Remember: The complete managed system name is *monitoredqueuemanagername*:*agentname*:MQ. The maximum length for the complete managed system name is 32 characters, so the maximum length for the middle qualifier *agentname* depends on the length of the queue manager name. If the value that is specified for the **AGTNAME** parameter exceeds the maximum length, the value for *agentname* is truncated to no less than 8 characters.

3. Save and close the `mq_silent_config.txt` file, and then run the following command from the command line:

   ```
   install_dir/BIN/mq-agent.sh config instance_name path_to_responsefile
   ```

   where *instance_name* is the name of the instance that you configure, and *path_to_responsefile* is the full path of the silent response file.

What to do next

Log in to the Cloud APM console. If the agent instance with previous MSN is still displayed as offline, edit your application to remove it and then add the new agent instance with the assigned agent name.

Configuring transaction tracking for the WebSphere MQ agent

Transaction tracking data for IBM MQ (WebSphere MQ) can be displayed in the middleware and topology dashboards after you enable the data collection on the **Agent Configuration** page for the WebSphere MQ agent.

Before you begin

- Make sure that MQI application activity trace information collection is enabled at the queue manager. If you have not done it before the WebSphere MQ agent is configured and started, follow the instructions in "Enabling MQI application activity trace" on page 915 and then restart the agent.
- Make sure that the version of IBM MQ (WebSphere MQ) that you are using is supported by the transaction tracking feature. For the up-to-date information about supported IBM MQ (WebSphere MQ), see the prerequisites statement in the Detailed System Requirements Report for the WebSphere MQ agent.
- Make sure that the WebSphere MQ agent is configured to monitor the queue manager. For instructions, see "Configuring the WebSphere MQ agent" on page 915.

Remember: Ensure that you have upgraded the WebSphere MQ agent to the latest version. You must upgrade the agent, and configure and enable transaction tracking to see data in some of the widgets, such as the Message Volume widget.

Procedure

To configure transaction tracking for the WebSphere MQ agent, complete the following steps:

1. From the navigation bar, click **System Configuration** > **Agent Configuration**.

   The **Agent Configuration** page is displayed.
2. Click the **WebSphere MQ** tab.
3. Select the check boxes for the queue managers that you want to monitor and take one of the following actions from the **Actions** list:

   - To enable transaction tracking, click **Set Transaction Tracking** > **Enabled**. The status in the **Transaction Tracking** column is updated to Enabled.

     Tip: Tracking alias queues and remote queues is enabled by default. To reduce the volume of data that is being tracked, you can disable the tracking of alias and remote queues by clicking **Set Alias Queue Tracking** > **Disabled** from the **Actions** list. After tracking alias and remote queues is disabled, alias queues and remote queues are eliminated from the Transaction Topology view.

   - To disable transaction tracking, click **Set Transaction Tracking** > **Disabled**. The status in the **Transaction Tracking** column is updated to Disabled.

Results

You have configured the WebSphere MQ agent for tracking the selected queue managers. Transaction tracking data can be displayed in the middleware and topology dashboards. For more information, see “Adding middleware applications to the Application Performance Dashboard” on page 100.

Enabling data collection for queue and channel long-term history

By default, the queue long-term history and channel long-term history is not collected and is not displayed on any predefined dashboards or group widgets. However, you can enable the agent to collect the long-term history data and then use the **Attribute Details** tab to query the collected data.

Before you begin

Ensure the WebSphere MQ agent is installed and configured. For information, see “Configuring the WebSphere MQ agent” on page 915.

About this task

The channel long-term history or the queue long-term history data can be useful for detecting problems with individual channels or queues.

If you are a user of Tivoli Data Warehouse, the agent can also send the long-term history data to Tivoli Data Warehouse for further processing.

Procedure

Complete the following steps to enable the WebSphere MQ agent to collect queue long-term history and channel long-term history data:

1. Open the following agent environment file with a text editor. If the `mq.environment` file does not exist, create it by yourself.

   - Linux AIX *install_dir*`/config/mq.environment`
   - Windows *install_dir*`\Config\KMQENV_`*instance*

   where:

   - *install_dir* is the agent installation directory. The default is `/opt/ibm/apm` on Linux and AIX systems, and `C:\IBM\APM` on Windows systems.
   - *instance* is the agent instance name.
2. Enable the data collection by setting the **LH_COLLECTION** value to ENABLED.

   ```
   LH_COLLECTION=ENABLED
   ```

3. Optional: If you are a user of Tivoli Data Warehouseand you want the agent to send the collected data to Tivoli Data Warehouse, set the **LH_PVTHISTORY** value to ENABLED.

   ```
   LH_PVTHISTORY=ENABLED
   ```

   Remember: Enable this option only if you need the collected data to be sent to Tivoli Data Warehouse.
4. Save your change and restart the agent.

Results

The WebSphere MQ agent starts to collect the queue long-term history and channel long-term history data. If you specified `LH_PVTHISTORY=ENABLED`, the collected long-term history data can also be sent to Tivoli Data Warehouse.

What to do next

Use the **Attribute Details** tab to view the collected data on the dashboard for the configured agent instance. Select **Channel_Long-Term_History** or **Queue_Long-Term_History** from the **Data Set** list. For more information about the **Attribute Details** tab, see "Creating a custom chart or table page" on page 1068.

Enabling queue statistics monitoring for the queue manager of IBM MQ

By default, the queue statistics are not collected and are not displayed on any predefined dashboards or group widgets. However, you can enable the agent to collect statistics for the queue manager and then view the collected data.

Before you begin

Ensure the WebSphere MQ agent is installed and configured. For information, see "Configuring the WebSphere MQ agent" on page 915.

Procedure

Complete the following steps to enable the WebSphere MQ agent to collect statistics data:

1. Configure the queue manager to collect queue statistics information. Run the following MQSC command:

   ```
   ALTER QMGR STATQ(ON)
   ```

2. Set the interval over which the accounting data is collected. Run the following command:

   ```
   ALTER QMGR STATINT(n)
   ```

   Where **n** is the number of seconds over which the accounting data is collected.
3. Enable statistics information collection for a specific queue. Run the following MQSC command:

   ```
   ALTER QLOCAL(queue_name) STATQ(QMGR)
   ```

   Where **queue_name** is the name of the queue for which you want to collect statistics information.

What to do next

Use one of the following methods to view the MQ Queue statistics monitoring data:

- View the monitoring data from the **Attribute Details** tab of **MQ_Queue_Statistics** Data Set. For more information about the **Attribute Details** tab, see "Creating a custom chart or table page" on page 1068.
- Define the thresholds based on **Expired Message Count** and other metrics of **MQ_Queue_Statistics**. For more information about thresholds, see "Thresholds and resource groups" on page 958.

Remotely monitoring queue managers on MQ Appliance

You can use the WebSphere MQ agent to monitor remote queue manager on MQ Appliance environment.

Before you begin

- Install WebSphere MQ agent on a supported platform.
- Install IBM MQ Client. The version of MQ client must be same as the version of the remote MQ Queue Manager.

Procedure

1. Set up connection to the remote queue manager. On the remote queue manager, define a server connection channel and a listener that is used for communication with the monitoring agent. Run the following command:

   ```
   M2000# mqcli
   M2000(mqcli)#runmqsc qmgr_remote
   > DEFINE LISTENER(listener) TRPTYPE(TCP) PORT(port_NO)
   > DEFINE CHANNEL(chl_name)CHLTYPE(SVRCONN) TRPTYPE(TCP)
   ```

```
CONNAME('host_IP(port_NO)') QMNAME(Qmgr_remote)
> END
```

where:

- *qmgr_remote* is the name of the remote queue manager.
- *listener* is the name of the listener on the remote queue manager.
- *port_NO* is the port number to be used for the listener.
- *chl_name* is the name that you assign to both the server channel and the client channel.
- *host_IP* is the IP address of the remote system.

2. Configure the listener to start automatically and then start the listener on the remote queue manager by running the following commands on the remote system:

```
M2000# mqcli
M2000(mqcli)#runmqsc qmgr_remote
> ALTER LISTENER(listener) TRPTYPE(tcp) CONTROL(qmgr_remote)
> START LISTENER(listener)
> END
```

3. Make sure that channel authentication settings are configured appropriately for the user ID that is used to start the MQ agent instance. For more information, see Setting up a queue manager to accept client connections in IBM MQ Appliance Knowledge Center.
4. Create an instance of the WebSphere MQ agent for remote monitoring by following instructions in "Configuring the WebSphere MQ agent" on page 915 and provide remote queue manager connection information in prompts following **Remote Monitoring Settings**.

```
Remote Monitoring Settings (For a local queue manager, just press Enter in
this section) :
Connection name for remote monitoring, for example: 192.168.1.1(1415)
Connection Name (default is: null):
Channel name for remote monitoring, SYSTEM.DEF.SVRCONN is as default.
Channels (default is: null):
```

5. Start the WebSphere MQ agent instance.

Remotely monitoring HA queue managers on MQ Appliance

To remotely monitor HA queue manager on MQ Appliance, you have two options. One is to use a single agent instance to connect to whichever system that has the active queue manager. The other option is to use separate agent instance for each appliance that the queue manager might be running on.

About this task

Only the second option is explained here. To use different agent instances, you need two installations of the WebSphere MQ agent on Linux or UNIX systems. On Windows systems, you only need one installation of the agent and create separate agent instances.

Procedure

- Linux AIX

  Perform the following steps to use the WebSphere MQ agent installed on Linux or UNIX systems for remote monitoring:

  a) Install the WebSphere MQ agent in different directories on the system.

  b) Create an instance of each installed WebSphere MQ agent. For instructions, see "Configuring the WebSphere MQ agent" on page 915.

  c) Modify the configuration file of each agent instance to enable remote monitoring by replacing the content with the following lines:

```
SET GROUP NAME (GROUP1) -
DEFAULT(YES) -
RETAINHIST(120) -
COMMAND (YES) -
```

```
MSGACCESS(DESC) -
EVENTS(REMOVE) -
ACCOUNTINGINFO(REMOVE) -
STATISTICSINFO(REMOVE)

SET MANAGER NAME(qmgr_name) REMOTE(YES)
SET AGENT NAME(agentID)
SET QUEUE NAME(*) MGRNAME(qmgr_name) QDEFTYPE(PREDEFINED)
SET CHANNEL NAME(*) MGRNAME(qmgr_name)
PERFORM STARTMON SAMPINT(300) HISTORY(NO)
```

where:

- *qmgr_name* is the name of the HA queue manager.
- *agentID* is the ID to identify the queue manager system. It is usually the host name or the IP address of the remote system where the HA queue manager is running.

The configuration file name and path is `install_dir/config/hostname_mq_qmgr_name.cfg`.

d) Create a pair of client and server channels between the primary queue manager and the WebSphere MQ agent, between the secondary queue manager and the WebSphere MQ agent on the remote system where the primary queue manager is installed.

Remember: You must run all the following commands before you proceed to the next step.

a. Run the following commands for the primary queue manager:

```
M2000# mqcli
M2000(mqcli)#runmqsc qmgr_primary
>DEFINE LISTENER(listener_primary) TRPTYPE(TCP) PORT(port_no_primary)
>DEFINE CHANNEL(chl_name_primary) CHLTYPE(SVRCONN) TRPTYPE(TCP)
>DEFINE CHANNEL(chl_name_primary) CHLTYPE(CLNTCONN) TRPTYPE(TCP)
CONNAME('host_IP(port_no_primary)') QMNAME(qmgr_primary)
```

where:

- *qmgr_primary* is the name of the primary queue manager.
- *listener_primary* is the name of the listener for the primary queue manager.
- *port_no_primary* is the port number that is used by the listener.
- *chl_name_primary* is the name that you want to assign to both the server channel and the client channel.
- *host_IP* is the IP address of the system where the primary queue manager is installed.

b. Run the following commands for the secondary queue manager on the primary queue manager. This is to add the connection information for the secondary queue manager to the client channel definition table file of the primary queue manager. The same agent can then connect to the secondary queue manager automatically when the primary queue manager fails over.

```
>DEFINE LISTENER(listener_secondary) TRPTYPE(TCP) PORT(port_no_secondary)
>DEFINE CHANNEL(chl_name_secondary) CHLTYPE(SVRCONN) TRPTYPE(TCP)
>DEFINE CHANNEL(chl_name_secondary) CHLTYPE(CLNTCONN) TRPTYPE(TCP)
CONNAME('host_IP(port_no_secondary)') QMNAME(qmgr_secondary)
```

where:

- *qmgr_secondary* is the name of the secondary queue manager on the remote system. It is the same as the primary queue manager name.
- *listener_secondary* is the name of the listener for the secondary queue manager.
- *port_no_secondary* is the port number that is used by the listener.
- *chl_name_secondary* is the name that you want to assign to both the server channel and the client channel.
- *host_IP* is the IP address of the system where the secondary queue manager is installed.

c. Finally, run the following command:

```
>END
>EXIT
```

e) Create the client channel definition table file (`AMQCLCHL.TAB`) for the WebSphere MQ agent instance on the 1st MQ appliance.

   a. Use the **`runmqsc`** command or the **`runmqsc -n`** command to create the `AMQCLCHL.TAB` file for the queue manager on the 1st MQ appliance:

   ```
   runmqsc -n
   >DEFINE CHANNEL(chl_name_primary) CHLTYPE(CLNTCONN) TRPTYPE(TCP)+
   >CONNAME('host_IP_appliance1(port_no_primary)') QMNAME(qmgr_name)
   ```

   where *host_IP_appliance1* is the IP address of the 1st MQ appliance; *chl_name_primary* and *port_no_primary* are the same as the ones that you defined in Step 4.

   Tip: By default, the `AMQCLCHL.TAB` file is created in the `var/mqm/qmgrs/`*`qmgr_name`*`/@ipcc` directory.

   b. Move the primary `AMQCLCHL.TAB` file to the *`agent_install_dir`*`/`*`arch`*`/mq/bin` directory on the system where the WebSphere MQ agent is installed for the primary queue manager.

f) Create the client channel definition table file (`AMQCLCHL.TAB`) for the WebSphere MQ agent instance on the 2nd MQ appliance.

   a. Use the **`runmqsc`** command or the **`runmqsc -n`** command to create the `AMQCLCHL.TAB` file for the queue manager on the 2nd MQ appliance:

   ```
   runmqsc -n
   >DEFINE CHANNEL(chl_name_secondary) CHLTYPE(CLNTCONN) TRPTYPE(TCP)+
   >CONNAME('host_IP_appliance2(port_no_secondary)') QMNAME(qmgr_name)
   ```

   where *host_IP_appliance2* is the IP address of the 2nd MQ appliance; *chl_name_secondary* and *port_no_secondary* are the same as the ones that you defined in Step 4.

   b. Move the secondary `AMQCLCHL.TAB` file to the *`agent_install_dir`*`/`*`arch`*`/mq/bin` directory on the system where the WebSphere MQ agent is installed for the secondary queue manager.

g) Make sure that channel authentication settings are configured appropriately for the user ID that is used to set up connection between agent instance and queue manager.

h) Start all listeners for both remotely monitored queue manager and start all the WebSphere MQ agent instances.

- Windows

  Perform the following steps to use the WebSphere MQ agent installed on Windows systems for remote monitoring:

  a) Install the WebSphere MQ agent on the Windows system.

  b) Create two instances of the WebSphere MQ agent for each HA queue manager.

  c) Modify the configuration file of each agent instance to enable remote monitoring by replacing the content with the following lines:

  ```
  SET GROUP NAME (GROUP1) -
  DEFAULT(YES) -
  RETAINHIST(120) -
  COMMAND (YES) -
  MSGACCESS(DESC) -
  EVENTS(REMOVE) -
  ACCOUNTINGINFO(REMOVE) -
  STATISTICSINFO(REMOVE)

  SET MANAGER NAME(qmgr_name) REMOTE(YES)
  SET AGENT NAME(agentID)
  SET QUEUE NAME(*) MGRNAME(qmgr_name) QDEFTYPE(PREDEFINED)
  SET CHANNEL NAME(*) MGRNAME(qmgr_name)
  PERFORM STARTMON SAMPINT(300) HISTORY(NO)
  ```

  where:

- *qmgr_name* is the name of the HA queue manager.
- *agentID* is the ID to identify the queue manager system. It is usually the host name or the IP address of the remote system where the HA queue manager is running.

Tip: The configuration file name and path is *install_dir*\TMAITM6_x64\mq_<*instance_name*>.cfg.

d) Create a pair of client and server channels between the primary queue manager and the WebSphere MQ agent, between the secondary queue manager and the WebSphere MQ agent on the remote system where the primary queue manager is installed.

Remember: You must run all the following commands before you proceed to the next step.

a. Run the following commands for the primary queue manager:

```
M2000# mqcli
M2000(mqcli)#runmqsc qmgr_primary
>DEFINE LISTENER(listener_primary) TRPTYPE(TCP) PORT(port_no_primary)
>DEFINE CHANNEL(chl_name_primary) CHLTYPE(SVRCONN) TRPTYPE(TCP)
>DEFINE CHANNEL(chl_name_primary) CHLTYPE(CLNTCONN) TRPTYPE(TCP)
CONNAME('host_IP(port_no_primary)') QMNAME(qmgr_primary)
```

where:

- *qmgr_primary* is the name of the primary queue manager.
- *listener_primary* is the name of the listener for the primary queue manager.
- *port_no_primary* is the port number that is used by the listener.
- *chl_name_primary* is the name that you want to assign to both the server channel and the client channel.
- *host_IP* is the IP address of the system where the primary queue manager is installed.

b. Run the following commands for the secondary queue manager on the primary queue manager. This is to add the connection information for the secondary queue manager to the client channel definition table file of the primary queue manager. The same agent can then connect to the secondary queue manager automatically when the primary queue manager fails over.

```
>DEFINE LISTENER(listener_secondary) TRPTYPE(TCP) PORT(port_no_secondary)
>DEFINE CHANNEL(chl_name_secondary) CHLTYPE(SVRCONN) TRPTYPE(TCP)
>DEFINE CHANNEL(chl_name_secondary) CHLTYPE(CLNTCONN) TRPTYPE(TCP)
CONNAME('host_IP(port_no_secondary)') QMNAME(qmgr_secondary)
```

where:

- *qmgr_secondary* is the name of the secondary queue manager on the remote system. It is the same as the primary queue manager name.
- *listener_secondary* is the name of the listener for the secondary queue manager.
- *port_no_secondary* is the port number that is used by the listener.
- *chl_name_secondary* is the name that you want to assign to both the server channel and the client channel.
- *host_IP* is the IP address of the system where the secondary queue manager is installed.

c. Finally, run the following command:

```
>END
>EXIT
```

e) Create the client channel definition table file (AMQCLCHL.TAB) for each WebSphere MQ agent instance.

a. Use the **runmqsc** command or the **runmqsc -n** command to create the AMQCLCHL.TAB file for the queue manager on the 1st MQ appliance:

```
runmqsc -n
>DEFINE CHANNEL(chl_name_primary) CHLTYPE(CLNTCONN) TRPTYPE(TCP)+
>CONNAME('host_IP_appliance1(port_no_primary)') QMNAME(qmgr_name)
```

where *host_IP_appliance1* is the IP address of the 1st MQ appliance; *chl_name_primary* and *port_no_primary* are the same as the ones that you defined in Step 4.

b. Create the AMQCLCHL.TAB file for the queue manager on the 2nd MQ appliance:

```
runmqsc -n
>DEFINE CHANNEL(chl_name_secondary) CHLTYPE(CLNTCONN) TRPTYPE(TCP)+
>CONNAME('host_IP_appliance2(port_no_secondary)') QMNAME(qmgr_name)
```

where *host_IP_appliance2* is the IP address of the 2nd MQ appliance; *chl_name_secondary* and *port_no_secondary* are the same as the ones that you defined in Step 4.

f) Rename the two AMQCLCHL.TAB file to different names, for example, NODE1.TAB and NODE2.TAB. Transfer them to the directory of *install_dir*\TMAITM6_x64, where *install_dir* is the installation directory of the WebSphere MQ agent.

g) Modify the kmqcma_*instance_name*.ini file to set the **MQCHLTAB** value to the client channel definition table file for each agent instance.
For example, set MQCHLTAB=NODE1.TAB in the kmqcma_instance1.ini file and set MQCHLTAB=NODE2.TAB in the kmqcma_instance2.ini file.

h) Open the Windows Register Editor, locate the following key of **MQCHLTAB** and change it from AMQCLCHL.TAB to the appropriate client channel definition table file name for each agent instance.

– **HKEY_LOCAL_MACHINE\SOFTWARE\Candle\KMQ\Ver730\*instance1*\Environment**

  MQCHLTAB=*NODE1.TAB*

– **HKEY_LOCAL_MACHINE\SOFTWARE\Candle\KMQ\Ver730\*instance2*\Environment**

  MQCHLTAB=*NODE2.TAB*

i) Make sure that channel authentication settings are configured properly for the user ID that is used to set up connection between agent instance and the queue manager.

j) Start all listeners for both remotely monitored queue manager and start all the WebSphere MQ agent instances.

Chapter 8. Integrating with other products and components

You can integrate other products and components with IBM Cloud Application Performance Management to provide you with a robust solution.

Integrating with Cloud Event Management

Cloud Event Management provides real-time incident management across your services, applications, and infrastructure. When you set up the integration between Cloud Event Management and IBM Cloud Application Performance Management, all the events that are generated in Cloud APM are sent to Cloud Event Management.

About this task

Configure a webhook URL in Cloud Event Management. Then, configure Cloud APM to use the webhook URL to send events to Cloud Event Management. For additional information about Cloud Event Management, see the IBM Cloud Event Management Knowledge Center.

Procedure

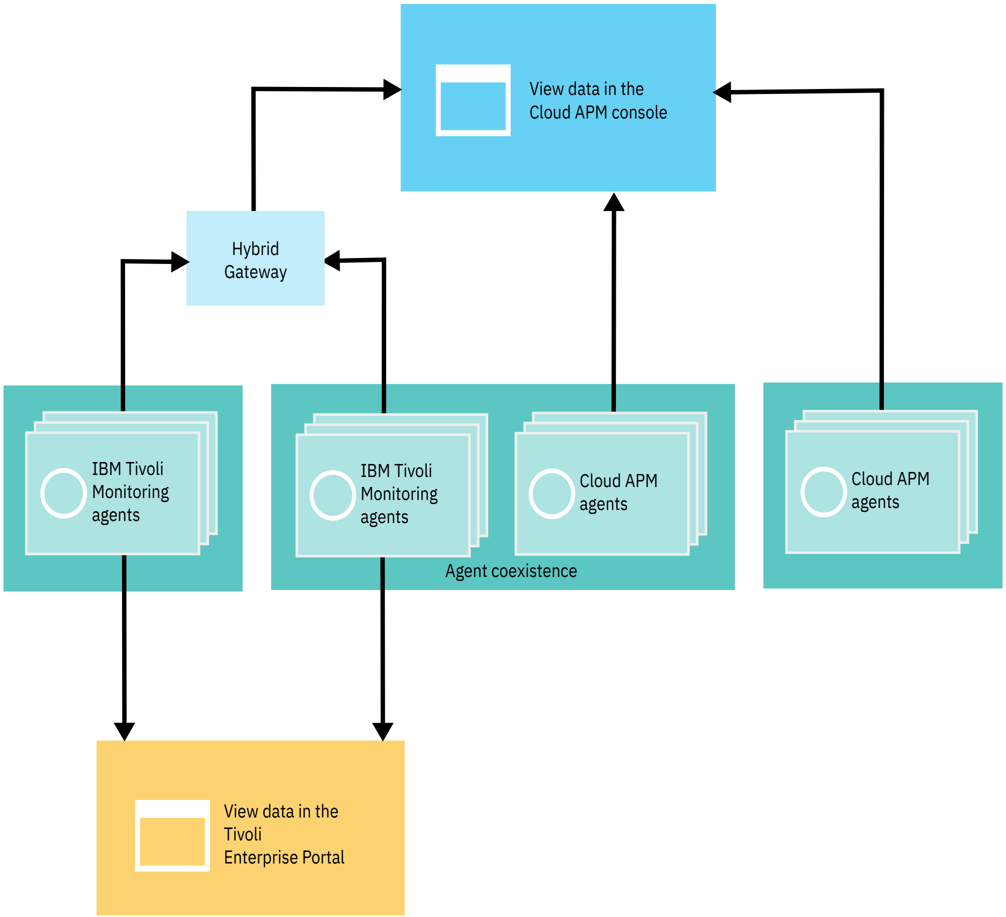
1. Click **Integrations** on the Cloud Event Management **Administration** page.
2. Click **Configure an integration**.
3. Go to the **IBM Cloud Application Performance Management** tile and click **Configure**.
4. Enter a name for the integration and click **Copy** to add the generated webhook URL to the clipboard. Ensure that you save the generated webhook to make it available later in the configuration process. For example, you can save it to a file.
5. To start receiving alert information from Cloud APM, ensure that **Enable event management from this source** is set to On in Cloud Event Management.
6. Click **Save**.
7. Log in to your Cloud APM subscription.
8. Navigate to **System Configuration** > **Advanced Configuration** > **Event Manager**. For more information, see Advanced Configuration.
9. Paste the webhook URL into the **Cloud Event Management Webhook** field.
10. Click **Save**.

Integrating with IBM Tivoli Monitoring V6.3

In an environment that includes both IBM Tivoli Monitoring and IBM Cloud Application Performance Management products, you can use these products together in several ways.

The following options are available to integrate with IBM Tivoli Monitoring:

- You can install the IBM Cloud Application Performance Management Hybrid Gateway to provide a consolidated view of managed systems from one or more Tivoli Monitoring domains and your Cloud APM domain in the Application Performance Dashboard pages. For more information about integrating agents, see "Hybrid Gateway" on page 933.
- You can install Tivoli Monitoring and Cloud APM agents on the same system. When agents coexist on the same computer, but not in the same directory, data from Cloud APM agents is available in the Cloud APM console and data from Tivoli Monitoring agents is available in the Tivoli Enterprise Portal. If coexisting agents are monitoring the same resources, certain limitations apply. For more information about agent coexistence, see "Cloud APM agent and Tivoli Monitoring agent coexistence" on page 930.


View data in the Cloud APM console
Hybrid Gateway
IBM Tivoli Monitoring agents
IBM Tivoli Monitoring agents
Cloud APM agents
Agent coexistence
Cloud APM agents
View data in the Tivoli Enterprise Portal

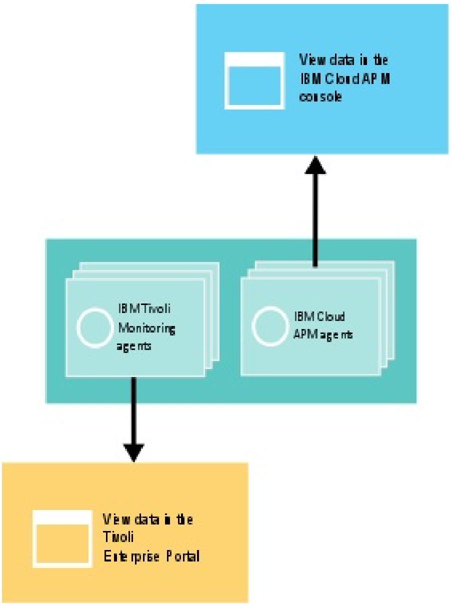
Cloud APM agent and Tivoli Monitoring agent coexistence

Agent coexistence is supported. You can install IBM Cloud Application Performance Management agents on the same computer where IBM Tivoli Monitoring agents are installed. However, both agent types cannot be installed in the same directory.

Cloud APM agents are referred to as version 8 agents. Tivoli Monitoring agents are referred to as version 6 or 7 agents.

When agents coexist on the same computer, data from version 8 agents is available in the Cloud APM console and data from version 6 or 7 agents is available in the Tivoli Enterprise Portal.

When version 6 or 7 agents, which coexist on the same computer as version 8 agents and monitor different resources, are integrated with the IBM Cloud Application Performance Management Hybrid Gateway, data from both agents is available in the Cloud APM console. For more information, see "Hybrid Gateway" on page 933.


View data in the IBM Cloud APM console
IBM Tivoli Monitoring agents
IBM Cloud APM agents
View data in the Tivoli Enterprise Portal

The following table lists the Tivoli Monitoring agents with documentation links:

Table 238. Documentation links for Tivoli Monitoring agents

Tivoli Monitoring agents	IBM Knowledge Center topic collection links
IBM Monitoring Agent for Citrix Virtual Desktop Infrastructure	IBM Tivoli Monitoring for Virtual Environments
IBM Tivoli Monitoring for Virtual Environments Agent for Cisco UCS	IBM Tivoli Monitoring for Virtual Environments
IBM Tivoli Monitoring for Virtual Environments Agent for Linux Kernel-based Virtual Machines	IBM Tivoli Monitoring for Virtual Environments
IBM Tivoli Monitoring for Virtual Environments Agent for VMware VI	IBM Tivoli Monitoring for Virtual Environments
IBM Tivoli Monitoring: HMC Base Agent	IBM Tivoli Monitoring
IBM Tivoli Monitoring: Linux OS Agent	IBM Tivoli Monitoring
IBM Tivoli Monitoring: UNIX OS Agent	IBM Tivoli Monitoring
IBM Tivoli Monitoring: Windows OS Agent	IBM Tivoli Monitoring
ITCAM Agent for DB2	ITCAM for Applications
ITCAM Agent for HTTP Servers	ITCAM for Applications
ITCAM Agent for J2EE	ITCAM for Applications
ITCAM for Microsoft Applications: Microsoft Active Directory Agent	ITCAM for Microsoft Applications
ITCAM for Microsoft Applications: Microsoft Cluster Server Agent	ITCAM for Microsoft Applications
ITCAM for Microsoft Applications: Microsoft Exchange Server Agent	ITCAM for Microsoft Applications
ITCAM for Microsoft Applications: Microsoft Hyper-V Server Agent	ITCAM for Microsoft Applications

Table 238. Documentation links for Tivoli Monitoring agents (continued)

Tivoli Monitoring agents	**IBM Knowledge Center topic collection links**
ITCAM for Microsoft Applications: Microsoft Internet Information Services Agent	ITCAM for Microsoft Applications
ITCAM for Microsoft Applications: Skype for Business Server Agent	ITCAM for Microsoft Applications
ITCAM for Microsoft Applications: Microsoft .NET Framework Agent	ITCAM for Microsoft Applications
ITCAM for Microsoft Applications: Microsoft SharePoint Server Agent	ITCAM for Microsoft Applications
Monitoring Agent for Microsoft SQL Server	ITCAM for Microsoft Applications
ITCAM Agent for SAP Applications	ITCAM for Applications
ITCAM Agent for WebSphere Applications	IBM Tivoli Composite Application Manager for Application Diagnostics for version 7.1 and earlier and in the ITCAM for Applications for version 7.2 and later.
ITCAM Agent for WebSphere DataPower Appliance	ITCAM for Applications
Monitoring Agent for WebSphere Message Broker	ITCAM for Applications
Monitoring Agent for WebSphere MQ	ITCAM for Applications Knowledge Center
ITCAM Extended Agent for Oracle Database	ITCAM for Applications
ITCAM Monitoring Agent for SAP HANA Database	ITCAM Monitoring Agent for SAP HANA Database Reference
ITCAM Web Response Time Agent	IBM Tivoli Composite Application Manager for Transactions

If coexisting agents are monitoring the same resources, the following scenariois not supported:

- Version 6 or 7 agents are integrated with the Hybrid Gateway to display data from both agents in the Cloud APM console. For example, if version 6 or 7 agents are connected to the same Cloud APM server through the Hybrid Gateway, do not use the version 8 IBM Integration Bus agent and the version 6 or 7 Monitoring Agent for WebSphere Message Broker to monitor the same broker on your system.

If a Tivoli Monitoring agent, which is integrated with the Hybrid Gateway to display data in the Cloud APM console, is monitoring a resource and you want your Cloud APM agent to monitor that resource, complete the following steps:

1. Remove the Tivoli Monitoring agent from any applications that include it.
2. Remove the Tivoli Monitoring agent from the Tivoli Monitoring managed system group that Cloud APM is configured to use.
3. Wait at least 24 hours and then install the Cloud APM agent and add it to an application.

When multi-instance agents that coexist on the same computer are integrated with the Hybrid Gateway and monitor the same resources, use different names for each instance to display data from both agents in the Cloud APM console.

For agents with a data collector, two agents of the same type are supported. Deep-dive diagnostics, resource, and transaction tracking data is displayed in the Cloud APM console. Resource data is displayed in the Tivoli Enterprise Portal. The following agents share a data collector:

Monitoring Agent for HTTP Server

The HTTP Server agent is a Cloud APM agent and the ITCAM Agent for HTTP Servers is an IBM Tivoli Monitoring agent. If you have both agents in your environment, you can configure both data collectors on the same HTTP Server for both agents. For more information about the HTTP Server agent, see "Configuring HTTP Server monitoring" on page 265.

Microsoft .NET agent

For more information about Microsoft .NET agent coexistence, see "Enabling transaction tracking in agent coexistence environment" on page 526.

WebSphere Applications agent

For more information about WebSphere Applications agent coexistence, see "Configuring WebSphere Applications agent" on page 854 and "Configuring the data collector for agent coexistence environment" on page 855.

Hybrid Gateway

To view monitoring data and events for your IBM Tivoli Monitoring and OMEGAMON agents in the Cloud APM console, you must create a managed system group and install the IBM Cloud Application Performance Management Hybrid Gateway in your Tivoli Monitoring domain, and configure communications in the Cloud APM console **Hybrid Gateway Manager**. Review the background information to help you plan for installing and configuring one or more Hybrid Gateways in your Tivoli Monitoring and Cloud APM environments.

For a video demonstration of installing and configuring the Hybrid Gateway, and then viewing data from your hybrid environment in the Cloud APM console, see Integrating with Tivoli Monitoring - Hybrid Gateway.

Where to install the Hybrid Gateway

You can install the Hybrid Gateway in one or more Tivoli Monitoring domains: one hub Tivoli Enterprise Monitoring Server per domain. For details about where to install the Hybrid Gateway, see the Preparing to install the Hybrid Gateway information. For Hybrid Gateway system requirements, which includes Tivoli Enterprise Portal Server, see the Hybrid Gateway Software Product Compatibility report, **Prerequisites** tab.

Tivoli Monitoring and OMEGAMON agents in the Cloud APM console

After you select the "My Components" predefined application or a defined application in the Application Performance Dashboard that includes Tivoli Monitoring or OMEGAMON managed systems (or both), you can see a summary status dashboard of all managed systems and you can see a detailed dashboard of a single managed system instance. You can also create dashboard pages in **Custom Views** tab.

You can view situation events for these agents in the **Events** tab. However, you cannot create new thresholds for Tivoli Monitoring and OMEGAMON agents in the Threshold Manager. Instead, create new situations in Tivoli Monitoring.

Not all possible Tivoli Monitoring and OMEGAMON events are available in the Events dashboard. Only events from agent nodes that can be added to an application are displayed. For example, for the Tivoli Monitoring agent for WebSphere Application Servers, events that are associated with a particular server instance are displayed, but events from the agent as a whole are not displayed.

View up to 1500 managed systems from each Tivoli Monitoring domain

The maximum number of managed systems, including subnodes, that you can view from a Tivoli Monitoring domain is 1500. By default, the limit is 200 systems. You can plan for supporting an increased amount of systems. For instructions, see "Planning for a large number of managed systems" on page 941.

The limit for all Tivoli Monitoring domains must be within the maximum supported by Cloud APM. For more information see "Architecture overview" on page 47.

Resource monitoring only

Resource monitoring is available for your Tivoli Monitoring agents. For more information about resource monitoring, see "Offerings and add-ons" on page 49 and "Capabilities" on page 56. If you have the IBM Cloud Application Performance Management, Advanced subscription, transaction tracking and diagnostics dashboards are not available for managed systems from your Tivoli Monitoring environment.

Tivoli Authorization Policy Server affects the availability of Tivoli Monitoring managed systems

For Tivoli Monitoring environments that include the Tivoli Authorization Policy Server, the managed systems that are available through the Hybrid Gateway are affected by the authorization policies. For more information, see Using role-based authorization policies in the Tivoli Monitoring Knowledge Center.

Supported Tivoli Monitoring and OMEGAMON agents

For a Tivoli Monitoring agent to be available for the Hybrid Gateway, it must also be supported in Cloud APM, with the exception of the iOS and OMEGAMON agents. The available Tivoli Monitoring agents and versions are listed here:

Product name	Product code	Supported version
IBM iOS agent	KA4	v6.30: 06.30.00.00 and later
ITCAM for Microsoft Applications: Microsoft SQL Server agent	KOQ	6.3.1.8, or later
ITCAM Agent for WebSphere Applications	KYN	07.20.00
TCAM Extended Agent for Oracle Database	KRZ	06.31.02.00
ITCAM Agent for DB2	KUD	07.10.00
IBM Tivoli Monitoring: Linux OS Agent	KLZ	v6.2.3: 06.23.01.00 and later v6.3.0: 06.30.01.00 and later
IBM Tivoli Monitoring: Windows OS Agent	KNT	v6.2.3: 06.23.01.00 and later v6.3.0: 06.30.02.00 and later
IBM Tivoli Monitoring: UNIX OS Agent	KUX	v6.2.3: 06.23.01.00 and later v6.3.0: 06.30.02.00 and later
ITCAM Agent for WebSphere MQ[1]	KMQ	07.10.01
ITCAM Agent for WebSphere Message Broker[1]	KQI	07.10.01

[1] Not available if you are using the Cloud APM, Base offering. For a list of agents in the Cloud APM, Base and Advanced offerings, see Capabilities .

For a list of OMEGAMON agents that you can display in the Cloud APM console, see the Getting started topic for your release in the IBM OMEGAMON for Application Performance Management topic collection on IBM Knowledge Center.

Preparing to install the Hybrid Gateway

To install the IBM Cloud Application Performance Management Hybrid Gateway, you must first ensure that your environment is set up correctly. Review the information to help you plan your Hybrid Gateway installation.

Where to install the Hybrid Gateway

The Hybrid Gateway must be installed on a Red Hat Enterprise Linux v6.2 (or later) x86-64 system that has a network connection with IBM Tivoli Monitoring and IBM Cloud Application Performance Management.

The Hybrid Gateway can be installed on the same system as your Tivoli Enterprise Portal Server or on a separate system from the Tivoli Enterprise Portal Server if the systems are running on Red Hat Enterprise Linux. However, the Hybrid Gateway cannot be installed on the same system as your Cloud APM server.

A Tivoli Monitoring domain has one hub Tivoli Enterprise Monitoring Server. When your Tivoli Monitoring environment consists of multiple domains, you can install the Hybrid Gateway in more than one domain.

For system requirements related to the Hybrid Gateway, click the **Hardware** tab in the Software Product Compatibility Report for Hybrid Gateway.

Setting up the Tivoli Enterprise Portal Server for the Hybrid Gateway

For Tivoli Monitoring environments where the portal server has a heavy load, you should install a separate dedicated portal server to service the requests from the Hybrid Gateway. If you set up a separate portal server:

- You can use the same host for the portal server and the Hybrid Gateway if the portal server is running Red Hat Enterprise Linux.
- Ensure that the separate portal server has the application support for the agents whose data is displayed in the Cloud APM console.
- Ensure that Tivoli Enterprise Portal clients are not connected to the separate portal server to complete administrative tasks such as creating custom workspaces, creating situations, and creating managed system groups.

The Tivoli Enterprise Portal Server must be at V6.3 Fix Pack 6 or higher. If your portal server is at an earlier version, the integrated Tivoli Monitoring agents might not be available for adding to an application in the Cloud APM console.

The IBM Tivoli Monitoring dashboard data provider must be enabled on the Tivoli Enterprise Portal Server. For details, see Verifying the dashboard data provider is enabled in the IBM Tivoli Monitoring topic collection on IBM Knowledge Center.

The TCP ports that must be opened on the Hybrid Gateway

The following TCP ports must be open on the Hybrid Gateway. For each port, one side sends a request and the other side provides a response. The side that initiates the connection is indicated.

- The Hybrid Gateway initiates a unidirectional connection with the Cloud APM server on port 443 and sends HTTPS requests.

  If the Hybrid Gateway uses a pass-through forwarding proxy to connect to the Cloud APM server, configure the Hybrid Gateway to use the proxy port instead of port 443 for unidirectional connections that it initiates with the Cloud APM server. For instructions, see “Using a forward proxy to communicate with the Cloud APM server” on page 937.
- If you use HTTP to communicate with the portal server, open port 15200. If you use HTTPS, open port 15201. The Hybrid Gateway initiates a unidirectional connection with the portal server on either port 15200 or 15201. To use a custom port, update the value of the **Portal Server Port** setting. For more information, see “Hybrid Gateway Manager” on page 943.

  Alternatively, if the Hybrid Gateway uses a pass-through forwarding proxy to connect to the portal server, configure the Hybrid Gateway to use the proxy port instead for unidirectional connections that it initiates with the portal server. Set the value of the **Pass-through Proxy Port** setting. For more information, see “Hybrid Gateway Manager” on page 943.
- For the Hybrid Gateway to listen to inbound EIF events from the Tivoli Enterprise Monitoring Server, open port 9998. The monitoring server initiates a unidirectional connection with the Hybrid Gateway on port 9998. The installation utility displays a warning if this port is not open.

Root privileges required to run the Hybrid Gateway installation script

You must run the Hybrid Gateway installation script with root privileges. For a list of supported operating systems, see IBM Software Product Compatibility Reports for IBM Cloud Application Performance Management - Agents V8.1.

Installing the Hybrid Gateway

Download and install the IBM Cloud Application Performance Management Hybrid Gateway to view managed systems from your IBM Tivoli Monitoring domain in the Cloud APM console.

Before you begin

Review and complete the required preparation tasks in Preparing to install the Hybrid Gateway.

Procedure

Complete the following steps to install the Hybrid Gateway in your Tivoli Monitoring domain:

1. Download the Hybrid Gateway package.

   The `APM_Hybrid_Gateway_Install.tar` file contains the Hybrid Gateway and installation script.

   a) Sign in to your account and go to Products and services on IBM Marketplace.

   b) Under **IBM Cloud APM**, click **More actions**.

   c) Click **Show additional packages**.

   d) Select **Hybrid Gateway**. If necessary, scroll down to find the entry.

   e) Click **Download**.

2. If needed, transfer the file to the system where the Hybrid Gateway will run.
3. Enter the following command to extract the files:

   ```
   tar -xf APM_Hybrid_Gateway_Install.tar
   ```

   The archive file contains a script that is used to deploy the Hybrid Gateway. The installation script is extracted into the directory and Hybrid Gateway files are extracted into subdirectories.

4. Change to the Hybrid Gateway directory and run the installation script with root user privileges:

   ```
   cd APM_Hybrid_Gateway_Install_version
   ./install.sh
   ```

   where *version* is the current version, such as 8.1.4.0.

   A prerequisite scan of your environment starts and takes a few moments to complete. If any requirements are missing, a message directs you to a log file with the reason for the failure. A prerequisite, such as insufficient disk space, stops the installation. You must address the failure, and start the installation again. You can also turn off the prerequisite check as described in Bypassing the prerequisite scanner.

5. After the system passes the prerequisite scan, respond to the prompt to accept the license agreement by selecting 1 for yes.

   A message then instructs you to log in to the Cloud APM console and configure the Hybrid Gateway before continuing. The default profile name, which is derived from the host name, is also displayed.

6. Press Enter to accept the default name or enter a profile name.

   If you already created a profile for this Tivoli Monitoring domain, use the identical name that you provided in the Hybrid Gateway Manager. If you haven't yet created a profile, you can accept the default name or provide a new name, but be sure to keep track of the name because you must use this name when you create the profile later. (See "Configuring the Hybrid Gateway using the Cloud APM console" on page 939.)

   After you press Enter, the Hybrid Gateway installation continues.

Results

The Hybrid Gateway is installed in the `/opt/ibm/hybridgateway` directory and starts automatically. The installation log file is at `/opt/ibm/hybridgateway/logs/install-hybridgateway-`*`timestamp`*`.log`. The Hybrid Gateway log files are in the `/opt/ibm/wlp/usr/servers/hybridgateway/logs` directory. Be aware that until the connection to the Tivoli Enterprise Portal Server is configured, connection failures are logged.

What to do next

- You can configure the Hybrid Gateway to use a forward proxy to communicate with the Cloud APM server. For instructions, see "Using a forward proxy to communicate with the Cloud APM server" on page 937.
- You can check the status of the Hybrid Gateway with the following command: `install_dir/hybridgateway/bin/hybridgateway.sh status`. For more options, see "Managing the Hybrid Gateway" on page 943.
- If you haven't already created the managed system group for the Hybrid Gateway, follow the instructions in "Creating the managed system group" on page 937.
- If you haven't already created a Hybrid Gateway profile for the Tivoli Monitoring domain, follow the instructions in "Configuring the Hybrid Gateway using the Cloud APM console" on page 939.
- If your Tivoli Monitoring environment has more than one hub domain, you can install the Hybrid Gateway in other domains. Repeat the steps in this procedure to install the Hybrid Gateway in another Tivoli Monitoring domain.

Using a forward proxy to communicate with the Cloud APM server

You can configure the IBM Cloud Application Performance Management Hybrid Gateway to use a forward proxy to communicate with the Cloud APM server.

Procedure

1. On the host where you installed the Hybrid Gateway, edit the `/opt/ibm/wlp/usr/servers/hybridgateway/bootstrap.properties` file:
   - If the Hybrid Gateway uses HTTP to communicate with the Cloud APM server, add the lines:

     ```
     http.proxyHost=proxy_host
     http.proxyPort=proxy_port
     ```

   - If the Hybrid Gateway uses HTTPS to communicate with the Cloud APM server, add the lines:

     ```
     https.proxyHost=proxy_host
     https.proxyPort=proxy_port
     ```

   where *proxy_host* is the host name or IP address of the proxy, accessible from the Hybrid Gateway host, and *proxy_port* is the port of the proxy.
2. Restart the Hybrid Gateway.

Creating the managed system group

Use the Object group editor in the Tivoli Enterprise Portal client to create a managed system group of managed systems that you want to view in the Cloud APM console.

Before you begin

- The types of IBM Tivoli Monitoring and OMEGAMON agents that you can include in the managed system group must be one of the supported agents. For example, for Tivoli Monitoring, some supported agents are Monitoring Agent for Oracle Database or Monitoring Agent for Linux OS.

  For the current list of supported Tivoli Monitoring agents, see "Supported Tivoli Monitoring and OMEGAMON agents" on page 934. For a list of OMEGAMON agents that you can display in the Cloud APM console, see the Getting started topic in the IBM OMEGAMON for Application Performance Management topic collection on IBM Knowledge Center.
- The Tivoli Monitoring and OMEGAMON agents must be connected to the same IBM Tivoli Monitoring infrastructure. If your environment has multiple Tivoli Monitoring domains, create a managed system group for each hub Tivoli Enterprise Monitoring Server for which a Hybrid Gateway is installed.
- By default, you can add up to 200 managed systems to the managed system group for viewing from the Tivoli Monitoring domain in the Application Performance Dashboard. You can increase the limit to as many as 1500 systems by taking several planning steps. For more information, see "Planning for a large

number of managed systems” on page 941. If you have multiple Hybrid Gateways for Tivoli Monitoring a environment with multiple hubs, the managed system group for each domain must be within the maximum supported by Cloud APM. For more information see “Architecture overview” on page 47.

- The default for handling subnodes changed in the Cloud APM March 2017 release. In previous releases, if you had agents with subnodes such as the WebSphere Applications agent, you had to assign the managing node to the managed system group and all the subnodes were included automatically. Although, you assigned one managing node to the managed system group, any subnodes were included in the count towards the managed system maximum.

  In the Cloud APM March 2017 release and later, subnodes whose metric data that you want to display in the Cloud APM console must be specifically assigned to the managed system group. The managing agent is discovered automatically if any of its subnodes are clearly assigned to the managed system group. For monitoring applications based on subnodes, Cloud APM might need to query the managing agent for information that is required to clearly identify the monitoring resources that appear in the Cloud APM dashboard navigator. It is for this reason that, with the current version of discovery mode, the managing agent is included automatically and at least one associated subnode is assigned to the managed system group configured for the Hybrid Gateway to use. The current discovery mode supports precise control over the subnode resources that can be viewed in the Cloud APM console and is better aligned with how Cloud APM applications are constructed, particularly for applications that involve large sets of subnode resource instances. Always use the current default discovery mode when you are integrating OMEGAMON agents with Cloud APM.

  You can specify which discovery mode version the Hybrid Gateway uses by assigning the appropriate value to an external property called `MSN_DISCOVERY_MODE`, which is processed by the Hybrid Gateway during initialization. To control which discovery mode is used by the Hybrid Gateway, add the `MSN_DISCOVERY_MODE` property (or change its current value) in the following properties file on the system where the Hybrid Gateway is installed and then restart the Hybrid Gateway.

  ```
  HG_install_dir/wlp/usr/servers/hybridgateway/bootstrap.properties
  ```

  The possible values are for the `MSN_DISCOVERY_MODE` property are:

  - `MSN_DISCOVERY_MODE=1` forces the Hybrid Gateway to use the original agent discovery mode where all subnodes are discovered automatically for any managing agent that is assigned to the Tivoli Monitoring managed system group.
  - `MSN_DISCOVERY_MODE=2` forces the Hybrid Gateway to use the new default agent discovery mode where only the subnodes that are clearly assigned to the managed system group are queried by the Hybrid Gateway. The associated managing agent or agents are discovered automatically.

- If you prefer to create the managed system group with the **`tacmd createsystemlist`** and **`tacmd editsystemlist`** IBM Tivoli Monitoringcommands, see the *IBM Tivoli Monitoring Command Reference* (https://www.ibm.com/support/knowledgecenter/SSTFXA_6.3.0/com.ibm.itm.doc_6.3/cmdref/itm_cmdref.htm) for information about how to run the commands.

Procedure

Complete these steps to create a managed system group in the Tivoli Enterprise Portal client.

1. Start the Tivoli Enterprise Portal client with a user ID and password that has full access to all the managed system types (**Allowed Applications** is set to **All Applications** for the user ID.)
2. Click **Object group editor**.
3. Expand the **Managed system** object and select **All Managed Systems** to combine multiple agent types (such as Windows OS and Oracle) in the managed system group. If you prefer that the managed system group contain only one monitoring agent type, such as Linux OS or WebSphere Applications, select the agent type.
4. Click **Create new group** and enter a name for the managed system group.

   The name can consist of letters and numbers, and must have no spaces, punctuation, or special characters other than the underscore (_).

   After you click **OK**, the new managed system group is displayed in the managed system folder.

5. Select the managed systems from the **Available Managed Systems** list and ◀ to move them to the **Assigned** list.

   You can select multiple managed systems by holding down the Ctrl key while you are clicking each managed system. After you select a managed system, you can use press Shift+click to select all managed systems between this selection and the first selection.

6. After you added managed systems to the group, click **OK** to save your changes and close the Object group editor.

What to do next

- After you create the managed system group and installed the Hybrid Gateway, you must configure the Hybrid Gateway in the Cloud APM console.
- For the configuration, specify the managed system group name that you created, the Tivoli Enterprise Portal user ID that is allowed access to all agent types, and the host name and port of the Tivoli Enterprise Portal Server.
- For installation instructions, see "Installing the Hybrid Gateway" on page 936.
- For configuration instructions, see "Configuring the Hybrid Gateway using the Cloud APM console" on page 939.

Configuring the Hybrid Gateway using the Cloud APM console

Use the **Hybrid Gateway Manager** page in the Cloud APM console to configure the IBM Cloud Application Performance Management Hybrid Gateway to connect to the Tivoli Enterprise Portal Server and to specify the managed system group.You can create a Hybrid Gateway profile for each hub Tivoli Enterprise Monitoring Server in your environment.

Procedure

Complete the following steps to configure the Hybrid Gateway in the Cloud APM console.

1. If you are not already logged in to the Cloud APM console, log in now.

   (See "Starting the Cloud APM console" on page 957.)

2. Click **System Configuration** > **Hybrid Gateway Manager**.

   The page displays with a table of any the Hybrid Gateways that were configured for your Tivoli Monitoring domains. If a profile with a blank name is displayed, it is for the Hybrid Gateway that was installed before the August 2017 release. For more information, see "Profile Name" on page 943.

3. Click ⊕ **Add** to open the **Add Hybrid Gateway** window, enter a new name in the **Profile Name** field, and click **Add**.

   If you already installed the Hybrid Gateway on the Tivoli Monitoring domain, be sure to use the same name that you provided or accepted during Hybrid Gateway installation. The profile name entered during installation and the name you enter here must be an exact match.

   The **Edit Hybrid Gateway** window opens.

4. In the **Managed System Group Name** field, enter the name of the manage system group for the Hybrid Gateway.

   This is the name that you used in "Creating the managed system group" on page 937.

5. Specify the address, port, and web communications protocol of the Tivoli Enterprise Portal Server:

Option	Description
Portal Server Host Name	Enter the portal server host IP address or fully qualified host or domain name.
Portal Server Port	Enter the port number that is used by the portal server for web communications. The default port is 15200 for HTTP or 15201 for HTTPS. A value of 0 sets the port to the default 15200 for HTTP or 15201 for HTTPS.

Option	Description
Portal Server Protocol	Select the **HTTP** Internet Protocol or the secure **HTTPS** Internet Protocol to connect to the portal server.

6. Complete the **Portal Server User Name** and **Portal Server Password** fields with the logon user name and corresponding password for starting the Tivoli Enterprise Portal client.

   The user ID must have access to all monitoring agent types (**Allowed Applications** is set to **All Applications**), such as the `sysadmin` ID. For more information, see Administer Users in the Tivoli Monitoring IBM Knowledge Center.

7. If access to the portal server goes through a proxy server, specify the address, port, and web protocol:

Option	Description
Pass-Through Proxy Host Name	Enter the IP address or fully qualified name of the proxy host system.
Pass-Through Proxy Port	Enter the port number of the proxy host system.
Pass-Through Proxy Protocol	Select the protocol that is used for communications through the proxy: HTTP or HTTPS

Results

After you click **Save**, a connection is established with the Hybrid Gateway service and the managed systems from your Tivoli Monitoring domain are discovered. The managed system group is polled every 5 minutes for resource monitoring data.

What to do next

- You must configure Tivoli Monitoring to interact with Cloud APM. For instructions, see "Configuring Tivoli Monitoring to integrate with Cloud APM" on page 940.
- You can repeat these steps to add a profile for each Tivoli Monitoring domain that you want to monitor managed systems from Cloud APM.
- You can manage existing profiles with the **Hybrid Gateway Manager** tools:

  Select a Hybrid Gateway and click ✎ **Edit** to open the **Edit Hybrid Gateway** window.

  Select a Hybrid Gateway that you no longer want and click ⊖ **Delete**. After you confirm that you want to delete the Hybrid Gateway, the profile is permanently removed.

  Click a column heading to sort the table by that column; Ctrl + Click another column to add a secondary sort.

  Click inside the filter text box and type the beginning of the value to filter by. As you type, the rows that do not fit the criteria are filtered out. To clear the filter, click the ✕ in the filter box or press the Backspace key.

Configuring Tivoli Monitoring to integrate with Cloud APM

To integrate your IBM Tivoli Monitoring domain with Cloud APM, you must complete such tasks as configuring the Tivoli Enterprise Portal Server and configuring the hub Tivoli Enterprise Monitoring Server.

Procedure

For each Tivoli Monitoring domain that you have a Hybrid Gateway installed on, complete these steps:

1. Configure the Tivoli Enterprise Portal Server to enable the dashboard data provider.

   The data provider is required to integrate Cloud APM with the Hybrid Gateway. For instructions, see the following topics:

   - Windows Windows: Installing the portal server (step 16c).
   - Linux AIX Configuring the portal server on Linux or AIX: command-line procedure (step 14).

If you are using the Hot Standby feature, you must specify a domain override. The Hybrid Gateway uses the domain name to collect data from both hub monitoring servers, regardless of the hub to which the portal server is connected.

2. If you want to view situation events from Tivoli Monitoring agents in the Cloud APM console, configure the hub monitoring server for one of the following scenarios:

   - To send events to the Hybrid Gateway only.
   - To send events to the Hybrid Gateway and additional EIF receivers such as Netcool/OMNIbus servers.

   Complete one of the following steps depending on your applicable scenario.

   a) To configure the hub Tivoli Enterprise Monitoring Server to send events to the Hybrid Gateway only, complete the steps in the Configuring the hub monitoring server to forward events topic.

      Specify port number 9998 for the **ServerPort** parameter.
      Cloud APM does not forward Tivoli Monitoring events to Netcool/OMNIbus. If you want to view Tivoli Monitoring events in Cloud APM and in Netcool/OMNIbus, you must configure Tivoli Monitoring to send the events to both systems.

   b) To configure the hub Tivoli Enterprise Monitoring Server to send events to the Hybrid Gateway and another EIF receiver such as a Netcool/OMNIbus server, configure the default EIF receiver using the steps in the Configuring the hub monitoring server to forward events topic.

      The topic also provides information about creating additional EIF destinations by using the **tacmd createEventDest** command. Specify port 9998 as the EIF port number for the Hybrid Gateway destination.

   c) Configure any situations that exist for the agents in the Hybrid Gateway managed system group to ensure that the situation events are sent to the EIF destination for the Hybrid Gateway. For instructions, see the Specifying which situations forward events to Netcool/OMNIbus topic.

What to do next

Review the Application Performance Dashboard to confirm that the managed systems from your Tivoli Monitoring domain are passed through the Hybrid Gateway:

1. Click **Performance** > **Application Performance Dashboard** to open the **All My Applications** dashboard.
2. In the summary box for "My Components", click **Components** to open the status summary dashboard for all component-managed systems (except for the WebSphere Applications agent). If you don't have a "My Components" application, add an application as described in “Managing applications” on page 1073.
3. Look for managed systems from your Tivoli Monitoring domain, indicated by an **ITM** (IBM Tivoli Monitoring) domain icon in the status summary group widget title. If any managed systems are missing, go to the Cloud Application Performance Management Forum and search on "Hybrid Gateway".

You can create applications with managed systems from your Tivoli Monitoring domains and include managed systems from your Cloud APM domain. For more information, see “Managing applications” on page 1073.

Planning for a large number of managed systems

The maximum number of managed systems that you can view from your IBM Tivoli Monitoring domain is 1500. If you include an agent that has subnodes in the managed system group that you created for the Hybrid Gateway profile, all the subnodes, as well as the agent, count towards the limit. By default, this limit is 200 managed systems, but you can take a number of planning steps to extend the limit.The limit for all Tivoli Monitoring domains must be within the maximum supported by Cloud APM. For more information see “Architecture overview” on page 47.

- Set the value of the Tivoli Enterprise Portal Server **KFW_REPORT_NODE_LIMIT** environment variable to a number greater than or equal to the number of managed systems for the Hybrid Gateway. The default value is 200. For instructions, see Tivoli Enterprise Portal Server configuration settings.

If the managed systems exceed this setting, the Tivoli Enterprise Portal Server `KfwServices` message log displays a message similar to the following example:

```
56C6246F.0000-10:ctreportmanager.cpp,2864,"CTReport::Manager::executeDefinitionDual") Query is targeting 1497 nodes which exceeds the current limit of 200 nodes.
```

- If you are viewing a large number of systems, performance might degrade depending on the type of agents, the network latency between the Hybrid Gateway and the managed systems, and the size of the environment monitored by each agent (amount of data collected and posted). To avoid this effect, select only the agents that provide necessary data and ensure fast networking connectivity between the monitored systems and the Hybrid Gateway host.
- As the network latency increases, the time to collect data from a given number of agents increases. The Hybrid Gateway attempts to gather data from each agent every 5 minutes. If the time to collect data from all agents exceeds 5 minutes, the Hybrid Gateway misses data samples, and therefore metrics are unavailable for some of the managed systems on the Application Performance Dashboard pages.
- To compensate for very slow network speeds, you can try increasing the number of threads used by the Hybrid Gateway to gather data samples. The **`MAX_COLLECTOR_THREADS`** parameter of the Hybrid Gateway `bootstrap.properties` file controls the thread number. The default value is 50.

Uninstalling the Hybrid Gateway

If you no longer want to view the IBM Tivoli Monitoring managed systems in the Cloud APM console, uninstall the IBM Cloud Application Performance Management Hybrid Gateway.

Procedure

1. In the Hybrid Gateway *`install_dir`*`/hybridgateway/bin` directory (such as `/opt/ibm/hybridgateway/bin`), run the following command:

   ```
   ./hybridgateway.sh uninstall
   ```

   The Hybrid Gateway is removed and a message confirms that it has been uninstalled successfully. If you have any applications in the Cloud APM console that include hybrid agents, the hybrid agents continue to appear until the monitoring infrastructure processes their removal.
2. In the Cloud APM console, Click **System Configuration** > **Hybrid Gateway Manager**.
3. Select the Hybrid Gateway profile that you no longer want and click **Delete**.

   After you confirm that you want to delete the Hybrid Gateway, the profile is permanently removed.

What to do next

- To remove any hybrid agent managed systems from an application in the Cloud APM console, follow the instructions in "Managing applications" on page 1073 for editing an application.
- If, instead of successful removal of the software, you get an error message similar to the one shown in this example, review the log file for the possible causes:

  ```
  error: Failed dependencies:
  ibm-java-x86_64-jre is needed by (installed) smai-kafka-00.08.00.00-1.el6.x86_64
  Uninstallation failed. The uninstaller was unable to remove some of the components,
  please inspect the
   log file ("/tmp/hybridgateway/logs/uninstall-hybridgateway-20150228080551.log")
  for more information.
  ```

  The error shown in the example occurred because the ibm-java-x86_64-jre is required by an externally installed package on the system. The installer does not remove the JRE because it would likely render the other package nonfunctional. As a workaround, uninstall the products with the dependency on ibm-java-x86-64-jre before uninstalling the Hybrid Gateway.

Managing the Hybrid Gateway

Use the commands available for the IBM Cloud Application Performance Management Hybrid Gateway service to start or stop it, to check the status, to uninstall the Hybrid Gateway, and to collect the log files if instructed by IBM Support.

About this task

These steps assume that the Hybrid Gateway installation directory is `/opt/ibm/`. On the system where the Hybrid Gateway is installed, take any of the following steps from the command prompt:

Procedure

- To start the Hybrid Gateway service, enter **`/opt/ibm/hybridgateway/bin/hybridgateway.sh start`**.
- To stop the Hybrid Gateway service, enter **`/opt/ibm/hybridgateway/bin/hybridgateway.sh stop`**.
- To check the status of the Hybrid Gateway service, enter **`/opt/ibm/hybridgateway/bin/hybridgateway.sh status`**.
- To uninstall the Hybrid Gateway, enter **`/opt/ibm/hybridgateway/bin/hybridgateway.sh uninstall`**.

  See also "Uninstalling the Hybrid Gateway" on page 942.
- To check the Hybrid Gateway log files, go to `/opt/ibm/wlp/usr/servers/hybridgateway/logs`.
- To collect the Hybrid Gateway log files for IBM Support, enter **`/opt/ibm/hybridgateway/collectLogs.sh`**.

  The log files are collected and a message shows the location of the compressed log files and asks you to return them to IBM Support.

Hybrid Gateway Manager

Configure the IBM Cloud Application Performance Management Hybrid Gateway for viewing monitoring data from your IBM Tivoli Monitoring domain in the Cloud APM console. You can create a Hybrid Gateway profile for each hub Tivoli Enterprise Monitoring Server in your environment.

After you click **System Configuration** > **Hybrid Gateway Manager**, the page is displayed with a list of the defined Hybrid Gateways.

The page has a table of all the hybrid gateways that were configured for your Tivoli Monitoring domains, and has tools for managing the Hybrid Gateway profiles:

- ⊕ **Add** opens the **Add Hybrid Gateway** window for naming the new profile. After you enter a name and click **Add**, the **Edit Hybrid Gateway** window opens.
- Select a hybrid gateway and click ✎ **Edit** to open the **Edit Hybrid Gateway** window.
- Select a hybrid gateway that you no longer want and click ⊖ **Delete**. After you confirm that you want to delete the hybrid gateway, the profile is permanently removed.
- Click a column heading to sort the table by that column; Ctrl + Click another column to add a secondary sort.
- Click inside the filter text box and type the beginning of the value to filter by. As you type, the rows that do not fit the criteria are filtered out. To clear the filter, click the ✕ in the filter box or press the Backspace key.

The mandatory fields that you must populate to configure the Hybrid Gateway are marked with an asterisk (*) in the **Edit Hybrid Gateway** window.

Profile Name

The given name for the Hybrid Gateway profile, which can be up to 128 letters, numbers and underscores (_).

The profile name is requested during Hybrid Gateway installation. If you already installed the Hybrid Gateway on the Tivoli Monitoring domain, use the name that you provided or accepted during Hybrid Gateway installation.

Older versions of the Hybrid Gateway don't use a named profile to access their configuration data. If you installed the Hybrid Gateway before the Cloud APM August 2017 release, you have a special, unnamed (blank) profile name. Only one older version of the Hybrid Gateway is allowed to connect to the Cloud APM server. If you configured the earlier version Hybrid Gateway, the unnamed profiled shows the configured values. If you did not configure the earlier version Hybrid Gateway, the unnamed profile shows the default values. You can keep the unnamed profile, or delete and add it again later as needed, and it can be used only for the March 2017 (or earlier) version Hybrid Gateway.

Managed System Group Name
The Tivoli Enterprise Portal Server managed system group that you created for viewing supported monitoring agents in the Cloud APM console. Any monitoring agent types that are not supported by your Cloud APM offering are not shown in the console regardless of their inclusion in the managed system group.
For guidance and limitations when creating the managed system group for hybrid enablement, see .

Portal Server Host name
The Tivoli Enterprise Portal Server host IP address or fully qualified domain name.

Portal Server Port
The port number used by the Tivoli Enterprise Portal Server for communications. The default port is `15200` for HTTP or `15201` for HTTPS. A value of 0 sets the port to the default `15200` for HTTP or `15201` for HTTPS.

Portal Server Protocol
Determines whether to use the HTTP Internet protocol or the secure HTTPS protocol to connect to the Tivoli Enterprise Portal Server. Default: `http`.

Portal Server User Name
The user name for starting the Tivoli Enterprise Portal client. This user ID must have access to all monitoring agent types (**Allowed Applications** is set to **All Applications**). For more information, see Administer Users in the Tivoli Monitoring Knowledge Center.)

Portal Server User Password
The password that is associated with the Tivoli Enterprise Portal logon user name.

Pass-Through Proxy Host Name
Used if the Tivoli Enterprise Portal Server communicates through a pass-through proxy server. Enter the IP address or fully qualified name of the proxy host system.

Pass-Through Proxy Port
Used if the Tivoli Enterprise Portal Server communicates through a pass-through proxy server. Enter the port number for communicating with the proxy.

Pass-Through Proxy Protocol
Used if the Tivoli Enterprise Portal Server communicates through a pass-through proxy server. Enter the protocol used for communications through the proxy. Default: `http`.

The Tivoli Monitoring agents that you are viewing in the Cloud APM console are in your IBM Tivoli Monitoring environment. You can view them in the Application Performance Dashboardpages, but you cannot create thresholds for these agents in the **Threshold Manager**.

Integrating with OMEGAMON

You can view data and events for your OMEGAMON application components in the Cloud APM console by purchasing the z Systems Extension Pack and using the Hybrid Gateway to connect one or more deployed OMEGAMON agents to Cloud APM.

Before you begin

- To use the z Systems Extension Pack, you must have either the IBM Cloud Application Performance Management, Advanced or IBM Cloud Application Performance Management, Base offering.

- One or more licensed OMEGAMON agents must be running on z Systems LPARS that are being monitored.
- The OMEGAMON agents are connected to the IBM Tivoli Monitoring infrastructure.

For a list of OMEGAMON agents that you can display in the Cloud APM console, see the Getting started topic for your release in the IBM OMEGAMON for Application Performance Management topic collection on IBM Knowledge Center.

Procedure

To integrate OMEGAMON with Cloud APM, complete the following steps:

1. After the z Systems Extension Pack is added to your Cloud APM product, complete the following Hybrid Gateway tasks:
   a) Install the Hybrid Gateway.
   b) Create the managed system group that you want to view in the Cloud APM console.
   c) Configure the Hybrid Gateway in the Cloud APM console so that you can connect the Hybrid Gateway to the Tivoli Enterprise Portal Server and specify a managed system group.

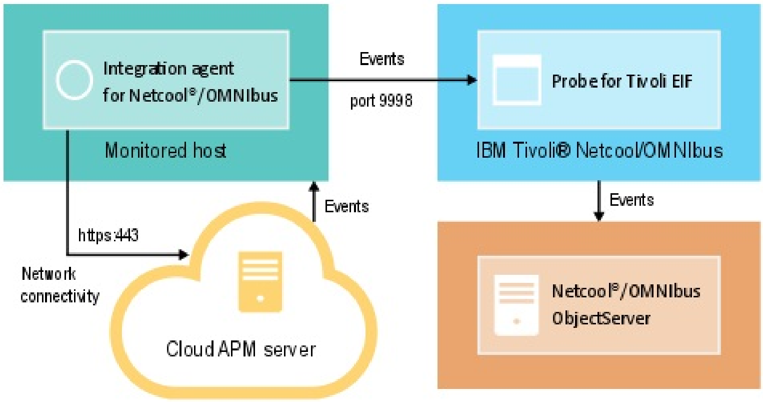
   For more information, see the required topics in the "Hybrid Gateway" on page 933 section.
2. To view the status of your applications in the dashboard, log in to the Cloud APM console from your browser. For more information, see "Starting the Cloud APM console" on page 957.

Integrating with Netcool/OMNIbus

You can forward events from IBM Cloud Application Performance Management into your on-premises IBM Tivoli Netcool/OMNIbus event manager.

Procedure

1. To view the Integration Agent for Netcool/OMNIbus, and how it integrates in Cloud APM with the Probe for Tivoli EIF to forward events to Netcool/OMNIbus, see the following configuration:



   The Integration Agent for Netcool/OMNIbus automatically connects with the Cloud APM server. This connectivity allows events to flow from the server into the network without any inbound network connections.
2. Configure the integration for Netcool/OMNIbus.

Installing and configuring the Integration Agent for Netcool/OMNIbus

To install the Integration Agent for Netcool/OMNIbus, you must download an archive file from the IBM Marketplace website, extract the agent installation files, and then start the installation script. After installation, the agent starts automatically, but must be configured.

About this task

Only one instance of the Integration Agent for Netcool/OMNIbus can forward events from a single instance of a Cloud APM service subscription to the Netcool/OMNIbus event manager at a time.

Procedure

1. Download the Cloud APM Integration archive file that includes the Integration Agent for Netcool/OMNIbus:
   a) Sign in to your account and go to Products and services on IBM Marketplace.
   b) Under **IBM Performance Management**, click **More actions**.
   c) Click **Show additional packages**.
   d) Select **IBM Performance Management OMNIbus Integration on Cloud**. If necessary, scroll down to find this item.
   e) Click **Download**.
2. Save the file to a staging directory of your choosing. Install the agent on any system that has network connectivity to the Tivoli Netcool/OMNIbus Probe for Tivoli Event Integration Facility (EIF). If needed, transfer the installation archive file to the systems to be monitored. The archive file contains the agent and installation script.
3. Extract the installation file:

   Linux

   a. Open a terminal shell session on the Red Hat Enterprise Linux system.
   b. Go to the directory where the archive file is located.
   c. Extract the installation files by using the following command:

   ```
   tar -xf ./apm_integration_agents_xlinux_8.1.4.0.tar
   ```

   Windows

   Extract the `apm_integration_agents_win_8.1.4.0.zip` file.

   The installation script is extracted to a directory named for the archive file and version. For example, `IPM_Agent_Install_8.1.3.2`. Agent binary and configuration-related files are extracted into subdirectories within that directory.
4. Run the installation script with Administrator privileges from the directory that is named for the archive file and version.

   If you are installing the Integration Agent for Netcool/OMNIbus on the same system where your Probe for Tivoli EIF is located and the Probe for Tivoli EIF is using the default port of 9998, the Integration Agent for Netcool/OMNIbus is automatically configured to connect to your Probe for Tivoli EIF.

   Important: If you are installing the Integration Agent for Netcool/OMNIbus on a system that is different from the one where your Probe for Tivoli EIF is located, or if you are using a port number that is different from the default for the Probe for Tivoli EIF, you must configure the Integration Agent for Netcool/OMNIbus after the installation is complete.

   Complete the following steps to install the agent:

   Linux `installAPMAgents.sh`
   Windows `installAPMAgents.bat`

   You are prompted to install the Integration Agent for Netcool/OMNIbus.

A prerequisite scan of your environment starts and takes a few moments to complete. If any requirements are missing, a message directs you to a log file with the reason for the failure. A missing prerequisite, such as a missing library or insufficient disk space, stops the installation. You must address the failure, and start the installation again.

The agent is configured with the following default settings:

```
Tivoli EIF Probe Host: localhost
Tivoli EIF Probe Port: 9998
```

After installation, the Integration Agent for Netcool/OMNIbus is automatically started.

The monitoring agent is installed into the directory you specify (*install_dir*). The following default directories are used:

Linux `/opt/ibm/apm/agent`
Windows `C:\IBM\APM\`

5. If you are installing the Integration Agent for Netcool/OMNIbus on a system that is different from the one where your Probe for Tivoli EIF is located, or if the Probe for Tivoli EIF is using a port number that is different from the default port of 9998, the Integration Agent for Netcool/OMNIbus must be configured to connect to your Probe for Tivoli EIF.

   Note: If you installed the Integration Agent for Netcool/OMNIbus on the same system where your Probe for Tivoli EIF is located and the Probe for Tivoli EIF is using the default port of 9998, you do not need to complete this step.

   Linux Complete the following steps to configure the agent:

   a. Run the following command:

   ```
   install_dir/bin/omnibus-agent.sh config
   ```

   b. When prompted, provide your Probe for Tivoli EIF host name and port number.

      After configuration is complete, the Integration Agent for Netcool/OMNIbus is automatically started.

   Linux Alternatively, you can use the following steps to review and change your configuration settings.

   a. Open the response file `install_dir/samples/omnibus_silent_config.txt` in a text editor.
   b. Edit the file to set or change your configuration settings. Ensure that you uncomment the configuration lines.
   c. Save and close the response file.
   d. Reconfigure the agent. Run the following command, specifying the fully qualified path to the silent configuration file you edited:

   ```
   install_dir/bin/omnibus-agent.sh config install_dir/samples/omnibus_silent_config.txt
   ```

   e. Restart the agent to implement your changes:

   ```
   install_dir/bin/omnibus-agent.sh stop
   install_dir/bin/omnibus-agent.sh start
   ```

   Windows Complete the following steps to configure the agent:

   a. Open the `install_dir\samples\omnibus_silent_config.txt` response file in a text editor.
   b. Edit the file to specify your Probe for Tivoli EIF host name and port number. Ensure that you uncomment the configuration lines.
   c. Save and close the response file.
   d. Reconfigure the agent by specifying the fully qualified path to the silent configuration file you edited:

```
install_dir\BIN\omnibus-agent.bat config install_dir\samples\omnibus_silent_config.txt
```

e. Restart the agent to implement your changes:

```
install_dir\BIN\omnibus-agent.bat stop
install_dir\BIN\omnibus-agent.bat start
```

What to do next

Follow the instructions in Configuring the integration for Netcool/OMNIbus.

If you want to stop using the Integration Agent for Netcool/OMNIbus or you want to move the agent to a different system, uninstall the agent by using the following command:

Linux `install_dir/bin/omnibus-agent.sh uninstall`

Windows `install_dir\BIN\omnibus-agent.bat uninstall`

Configuring the integration for Netcool/OMNIbus

After you install the Integration Agent for Netcool/OMNIbus , you must copy the event rules to the Probe for Tivoli EIF and modify them. You must also update the Netcool/OMNIbus ObjectServer and the database schema.

Before you begin

Before you complete the integration steps, stop the Integration Agent for Netcool/OMNIbus by using the following commands:

Linux `install_dir/bin/omnibus-agent.sh stop`

Windows `install_dir\BIN\omnibus-agent.bat stop`

install_dir is the default `/opt/IBM/apm/agent` or `C:\IBM\APM` directory or the directory that you specified during the Integration Agent for Netcool/OMNIbus installation.

About this task

After the installation of the Integration Agent for Netcool/OMNIbus, the following configuration files are in the *`install_dir`*`/localconfig/i0/omnibus` and the *`install_dir`*`\localconfig\i0\omnibus` directory:

- `itm_apm_db_update.sql`
- `itm_event.rules`
- `itm_apm_event.rules`
- `multitier/collection_itm_apm.sql`
- `multitier/display_itm_apm.sql`
- `multitier/GATE_itm_apm.map`

where *install_dir* is the default `/opt/IBM/apm/agent` or `C:\IBM\APM` or the directory that you specified during the Integration Agent for Netcool/OMNIbus installation.

You must copy the files to the computer systems where the Netcool Probe for Tivoli EIF and Netcool/OMNIbus ObjectServers are installed. If your Netcool/OMNIbus environment is configured for the single tiered architecture, which is the default, you only need the first three files. However, if your Netcool/OMNIbus environment is configured for the multitiered architecture, you also need to deploy the files in the multitier directory.

You must install the V08.20.01.00 version of the Integration Agent for Netcool/OMNIbus to obtain the files in the multitier directory.

Important: You must complete these steps even if your Probe for Tivoli EIF and Netcool/OMNIbus ObjectServer are already integrated with IBM Tivoli Monitoring, Probe for Tivoli EIF, IBM SmartCloud®

Monitoring - Application Insight®, IBM SmartCloud Application Performance Management, or a previous version of Cloud APM.

Procedure

In this procedure, when you follow links to the IBM Tivoli Monitoring documentation, complete only the steps that are provided on the linked page.

1. Copy the Integration Agent for Netcool/OMNIbus `itm_event.rules` and `itm_apm_event.rules` files to the Probe for Tivoli EIF installation directory.

   The following directories are the default directories:

   Linux *install_dir*/tivoli/netcool/omnibus/probes/linux2x86
   Windows *install_dir*\Tivoli\Netcool\omnibus\probes\win32

   Where *install_dir* is the directory where you install the Probe for Tivoli EIF.

2. Open the Probe for Tivoli EIF `tivoli_eif.rules` file in a text editor and complete one of the following steps:

   - If you are an existing IBM Tivoli Monitoring customer and have completed the OMNIbus integration already, add this line to your `itm_event.rules` file: `include "itm_apm_event.rules"`.
   - If you have not set up the OMNIbus integration already, uncomment the include statement for the `itm_event.rules` file.

   For detailed steps, see Updating the rules files of the EIF probe in the IBM Tivoli Monitoring documentation.

3. Update the Netcool/OMNIbus ObjectServer database schema by loading the `itm_apm_db_update.sql` file into the database if you configured a single tiered Netcool/OMNIbus architecture, which is the default architecture. If you configured a multitiered Netcool/OMNIbus architecture then load the `itm_apm_db_update.sql` file into the database of each Netcool/OMNIbus ObjectServer in the aggregation tier.

   Linux

   ```
   $OMNIHOME/bin/nco_sql -user user_name -password password
   -server server_name < itm_apm_db_update.sql
   ```

   Example:

   ```
   $OMNIHOME/bin/nco_sql -user smadmin -password passw0rd -server NCOMS <
   /tmp/apm/itm_apm_db_update.sql
   ```

   Windows

   ```
   itm_apm_db_update.sql | %OMNIHOME%\..\bin\isql -U user_name
   -P password -S server_name
   ```

   Example:

   ```
   \temp\apm\itm_apm_db_update.sql | %OMNIHOME%\..\bin\isql -U smadmin
   -P passw0rd -S NCOMS
   ```

   The following error messages might be displayed when you are running the scripts and these messages are harmless:

   - `Object exists` and `Attempt to insert duplicate row`, if the scripts were run previously (for example, for integration with an earlier version of Cloud APM or with Tivoli Monitoring).
   - `ERROR=Object not found on line 4 of statement "-- A workspace table for the ITM event clear automation..."` at or near `itm_event_clear`.
   - `ERROR=Object not found on line 1 of statement "delete from alerts.itm_problem_events;..."` at or near `itm_problem_events`.

- ERROR=Object not found on line 1 of statement "drop table alerts.itm_problem_events;..." at or near itm_problem_events.

4. Repeat step 5 so that the file is loaded into the Object Server twice to ensure that all dependencies are loaded correctly.
5. If you have a multitiered Netcool/OMNIbus architecture configured, perform the following steps:
   a) Copy the GATE_itm_apm.map file to the computer systems where the Netcool/OMNIbus uni-directional gateways are installed. You need to update all uni-directional gateways that transfer data between the Netcool/OMNIbus ObjectServers.

   b) Copy the mappings in the GATE_itm_apm.map file into the gateway map definition file that also uses the .map extension.

   The attributes should be added to the STATUSMAP mapping entry in the gateway map definition file. Your gateway map file might contain the following comment block to identify where to add these custom attributes:

   ```
   ##############################################
   #
   # CUSTOM alerts.status FIELD MAPPINGS GO HERE
   # ############################################
   ```

   c) Copy the collection_itm_apm.sql file to the computer systems where the Netcool/OMNIbus ObjectServers in the collection tier are installed.

   d) Update the Object Server database in the collection tier with the following command, which pipes the SQL command set into the SQL command-line tool and performs the updates to the ObjectServer database.

   - Windows

     ```
     type path_to_file\collection_itm_apm.sql | %OMNIHOME%\..\bin\redist\isql
     -U username
     -P password
     -S server_name
     ```

     where:

     $OMNIHOME
     Is the system-defined variable defining the installation location of OMNIbus.

     username
     Is the OMNIbus Object Server user name.

     password
     Is the OMNIbus Object Server password.

     server_name
     Is the OMNIbus Object Server name defined for process control.

     path_to_file
     Is the fully qualified path to the specified SQL file.

   - UNIX

     ```
     $OMNIHOME/bin/nco_sql -user username
             -password password
             -server server_name
             < path_to_file/collection_itm_apm.sql
     ```

     where:

     $OMNIHOME
     Is the system-defined variable defining the installation location of OMNIbus.

     username
     Is the OMNIbus Object Server user name.

     password
     Is the OMNIbus Object Server password.

server_name
Is the OMNIbus Object Server name defined for process control.

path_to_file
Is the fully qualified path to the specified SQL file.

e) If the Netcool/OMNIbus architecture includes a display tier, copy the `display_itm_apm.sql` file to the computer systems where the Netcool/OMNIbus ObjectServers in the display tier are installed. Update the Object Server database in the display tier with the following command, which pipes the SQL command set into the SQL command-line tool and performs the updates to the ObjectServer database.

- Windows

```
type path_to_file\display_itm_apm.sql | %OMNIHOME%\..\bin\redist\isql
-U username
-P password
-S server_name
```

where:

$OMNIHOME
Is the system-defined variable defining the installation location of OMNIbus.

username
Is the OMNIbus Object Server user name.

password
Is the OMNIbus Object Server password.

server_name
Is the OMNIbus Object Server name defined for process control.

path_to_file
Is the fully qualified path to the specified SQL file.

- UNIX

```
$OMNIHOME/bin/nco_sql -user username
        -password password
        -server server_name
        < path_to_file/collection_itm_apm.sql
```

where:

$OMNIHOME
Is the system-defined variable defining the installation location of OMNIbus.

username
Is the OMNIbus Object Server user name.

password
Is the OMNIbus Object Server password.

server_name
Is the OMNIbus Object Server name defined for process control.

path_to_file
Is the fully qualified path to the specified SQL file.

f) Restart the Netcool/OMNIbus uni-directional gateways between the collection and aggregation tiers.

g) If the Netcool/OMNIbus architecture includes a display tier, restart the uni-directional gateways between the aggregation and display tiers.

6. Start (or restart) the Probe for Tivoli EIF.
7. Restart the Integration Agent for Netcool/OMNIbus by using the following commands:

Linux `install_dir/bin/omnibus-agent.sh start`

Windows `install_dir\BIN\omnibus-agent.bat start`

Integrating with Operations Analytics - Log Analysis

When your environment includes IBM Operations Analytics - Log Analysis, you can integrate it to enable searching through application logs in the Cloud APM console.

About this task

Integrating with your installed Log Analysis application involves configuring the Cloud APM server with the URL. For more information about Log Analysis, see the IBM Operations Analytics - Developers Community.

You must provide the top level URL for your Log Analysis installation, for example:

```
https://loganalysis.example.com:9987/Unity
```

The Log Analysis URL must be accessible from the hosts where users work with the Cloud APM console. It is not necessary to make it accessible from the open Internet.

Procedure

1. In the Cloud APM console, click **System Configuration** > **Advanced Configuration**.
2. Select the **UI Integration** category.
3. In the **Log Analysis URL** field, enter the URL that is used to launch your Log Analysis application.

Results

The Log Analysis application is integrated and the feature is enabled for you to search through application logs from the Application Performance Dashboard.

Note:

Single sign-on is not supported from the Cloud APM console to the Log Analysis application.

What to do next

Select **Performance** > **Application Performance Dashboard** . Optionally select an application, then use the search box to search log files. By default, entries for the last hour are searched, but you can change this time period. If you select an application, only the logs on servers associated with this application are searched. For detailed instructions, see "Searching log files" on page 1057.

Integrating with Operations Analytics - Predictive Insights

When you integrate IBM Cloud Application Performance Management with Operations Analytics - Predictive Insights, Operations Analytics - Predictive Insights analyzes the metric data collected by Cloud APM and generates alarms when it identifies anomalies in the data.

Anomalies are displayed as events in the Cloud APM Dashboard, as described in "Investigating anomalies with Operations Analytics - Predictive Insights" on page 1085. You can then drill down to the Operations Analytics - Predictive Insights User Interface to view more details on an anomaly.

When you add Operations Analytics - Predictive Insights to a Cloud APM subscription, it is automatically configured to collect and analyze performance metrics. No further configuration is required. To add Operations Analytics - Predictive Insights to a Cloud APM subscription, go to IBM Support and open a Service Request.

You can integrate the following Cloud APM agents with Operations Analytics - Predictive Insights:

- Monitoring Agent for Db2
- Monitoring Agent for IBM Integration Bus
- Monitoring Agent for JBoss
- Monitoring Agent for Linux OS
- Monitoring Agent for Oracle Database

- Response Time Monitoring Agent
- Monitoring Agent for UNIX OS
- Monitoring Agent for VMware VI
- Monitoring Agent for WebSphere Applications
- Monitoring Agent for WebSphere MQ
- Monitoring Agent for Windows OS
- Monitoring Agent for Tomcat

Integrating with Alert Notification

For more flexibility beyond the basic email forwarding of Cloud APM, you can integrate with the Alert Notification product to extend Cloud APM capabilities for notifying users when problems occur.

Integration with Alert Notification gives you granular control over who receives notifications and how they are received. For example, each user can decide whether they want to receive email, SMS, or voicemail. Notifications can also be routed to Slack. Different users can receive different types of notifications based on time of day, day of week, and so on. Each user can decide what types of alerts they want to receive. For example, a database administrator might want to receive only database alerts that are warning or higher severity.

Before you begin

Alert Notification is automatically integrated for you in your IBM Cloud Application Performance Management subscription and your trial subscription.

About this task

Events are automatically configured to send to Alert Notification. Then, you can create notification policies for your subscription to determine the alerts for which you want to receive notifications.

You can add monitored applications to resource groups. For every saved resource group, you can configure one or several email addresses. When the performance of any managed system in a group exceeds a threshold, an email notification is sent to the addresses that are configured for the group.

Alert Notification sends email notification for any events that open on the managed systems that are assigned to the group.

Procedure

Complete Steps “1” on page 953 and “2” on page 953 for any Resource Group for which you want to route alerts based on the resource group.

1. If you want to route notifications based on resource groups, complete the following substeps:

   a) In the Cloud APM console, click **System Configuration** > **Resource Group Manager**.

   b) Select a resource group and click **Edit** to open the **Resource Group Editor**.

   Important: You must save a new or edited resource group before you open Alert Notification. After you save, the Resource Group Editor closes and you must reopen it if you want to continue to use it.

   If you do not save, after you configure email notification in Step “2” on page 953 and return to the Resource Group Editor, you might receive an error message that informs you synchronous update is not allowed.

2. Complete the following steps to configure email notification:

   a) In the APM console Resource Group Editor, click the **Configure email notification** URL to open the IBM Alert Notification application in a new browser tab or window.

   This action automatically creates a new policy based on the resource group you selected in Step “1” on page 953.

⚠ **Attention:** If your browser does not allow pop-up windows, the Alert Notification window is prevented from opening. You must set the browser to allow the Alert Notification window to open it for configuring email notification for a resource group.

b) In Alert Notification, configure Users and Groups and associate their email addresses with resource groups for receiving event notifications by email.

   You can set up your own Alert Notification policies and notifications for the type of alerts that you want to receive. For example, a Linux Administrator might want to receive email, SMS messages, or voicemail for all Linux systems. Then, they might want to turn off high severity alerts that are critical. A policy allows them to filter all events that are generated by the Linux OS agent. For information about using the Notifications Editor in the Alert Notification application, see Creating notification policies.

What to do next

For more information about Alert Notification, see IBM Alert Notification Documentation.

Integrating with Control Desk

You can configure your events from Cloud APM to automatically open tickets in IBM Control Desk.

About this task

You can integrate IBM Cloud Application Performance Management by using the on-premises version or the Cloud version of IBM Control Desk.

Procedure

Use one of the following procedures:

- To open tickets in your on-premises IBM Control Desk V 7.6, complete the following steps:
  1. Configure your IBM SmartCloud Notes® email account for Control Desk to use an IMAP mail client. During configuration, you must ensure that you select **Enable IMAP Access Now**. For more information, see Enabling IMAP access in the IBM Connections Social Cloud Knowledge Center.
  2. In the Cloud APM console, click **System Configuration** > **Advanced Configuration** and then set the following parameters:

     Target Email Addresses
     Specify your SmartCloud Notes email address that is used to create Service Request tickets.

     Email Subject Line
     Specify a subject line for the email, such as `PMaaS Event`.
  3. Go to Marketplace support. and select **Service Request** to submit a support ticket to complete you enablement.

     Provide the following information in your ticket:

     – SmartCloud Notes email address

       For example, `user@ibmserviceengage.com`.

     – SmartCloud Notes email password

     – SmartCloud Notes email fully qualified server name

       For example, `imap.notes.na.collabserv.com`.

     – SmartCloud Notes email port number

       For example, 993.

     – Customer IBM Control Desk URL

       Format the link as follows: `https://<subscriber-id>.sccd.ibmserviceengage.com/maximo_t4hj/webclient/login/login.jsp?welcome=-true`

4. To configure the email listener to parse the email and to handle it appropriately when you want to assign tickets to other groups in your IBM Control Desk on Cloud, see Configuring e-mail listeners.

- To open tickets in your IBM Control Desk Cloud environment, complete the following steps:

  1. Go to Marketplace support. and select **Service Request** to submit a support ticket to complete your enablement.

     Provide the following information in your ticket:

     – SmartCloud Notes email address

       For example, `user@ibmserviceengage.com`.

     – SmartCloud Notes email password

     – SmartCloud Notes email fully qualified server name

       For example, `imap.notes.na.collabserv.com`.

     – SmartCloud Notes email port number

       For example, 993.

     – Customer IBM Control Desk URL

       Format the link as follows: `https://`*`<subscriber-id>`*`.sccd.ibmserviceengage.com/maximo_t4hj/webclient/login/login.jsp?welcome=-true`

  2. To configure the email listener to parse the email and to handle it appropriately when you want to assign tickets to other groups in your IBM Control Desk on Cloud, see Configuring e-mail listeners.

Integrating with IBM Cloud

You can view monitoring information for your applications within the IBM Cloud environment by using selected data collectors.

When configured to collect data from an IBM Cloud application, a data collector enables the integration of monitoring capabilities with IBM Cloud. Data collectors transfer resource monitoring and diagnostics data about your IBM Cloud applications to the Cloud APM server. The Cloud APM server receives and processes monitoring information that is collected by the data collectors. The following types of IBM Cloud applications can be monitored:

- Liberty applications
- Node.js applications
- Python applications
- Ruby applications

After proper configurations of a data collector, you can view monitoring data on the Cloud APM console. For configuration instructions, see “General procedure for configuring data collectors” on page 184.

Integrating with IBM Cognos Analytics

You can run your installed IBM Cloud APM reports from the IBM Cognos Analytics user interface. Cognos Analytics 11.x is the replacement for Tivoli Common Reporting (TCR), which is at end of service.

For more information on installing the Cloud APM reports in a Cognos Analytics 11.x environment, see the IBM Support technote at https://www.ibm.com/support/pages/node/6257869

Integrating with IBM Agent Builder

You can create, modify, debug, and package agents using Agent Builder that extend the monitoring capabilities of an IBM Tivoli Monitoring or IBM Cloud Application Performance Management environment. A custom agent uses either of these environments to monitor any type of in-house or customized software.

For details, see the IBM Agent Builder User's Guide.

Chapter 9. Administering

Starting the Cloud APM console

Log in to the Cloud APM console from your browser to review the health status of your applications in the dashboards.

Before you begin

- Activate your account by using the link that is provided in the confirmation email you received after your initial sign-up for the service.
- To ensure that the user interface is not truncated, use a minimum resolution of 1280 x 1024.
- For optimal performance, use one of the supported browsers. For a list of the supported browsers, open the IBM Cloud Application Performance Management Software Product Compatibility Report, select the Prerequisites tab, and scroll down to Web Browsers.

Procedure

1. To access your Cloud APM console, use the link that is provided in the email alerting you that your service is ready.
2. You can also access your console from the IBM Marketplace website:
   a. Go to Products and services on the IBM Marketplace website.
   b. Log in with the user name and password that you used to register for the service.
   c. In the Cloud APM server row, click **Launch**.

Results

After you log in, the **Getting Started** page is displayed with learning options for **User Tasks** and **Administrator Tasks**, and links to **Community Resources**.

What to do next

- Familiarize yourself with the user interface elements by clicking the hypertext link to take a tour of the Cloud APM dashboard. Watch videos of the user tasks and administrator tasks to help you get started using and customizing your Cloud APM environment.
- Add applications for viewing dashboards of your resources in logical groupings such as Online Ordering. For instructions, see “Managing applications” on page 1073.
- Create thresholds to test for conditions that, when met, cause an event to open. For example, you can have a threshold that opens an event after storage capacity reaches 90%. For instructions, see “Threshold Manager” on page 967.
- Add and assign users to user groups and roles to control access to the Cloud APM console features and managed resources. For more information, see “Managing user access” on page 982.
- To learn about monitoring the IBM Java application stack and IBM integration stack, see “Scenarios” on page 90.
- If, instead of the Getting Started page or the Application Performance Dashboard, your browser goes to the IBM website, your user ID has no permissions to the Cloud APM console. You must request access from your administrator.
- If no metrics are shown for a data source, consult the Cloud Application Performance Management Forum on developerWorks®. Search the forum for "dashboard", reply to an entry to ask a related question, or create a new entry and describe the symptom.

- If you are starting the Cloud APM console from Internet Explorer 8, 9, or 10 and you get a `This page can't be displayed` error, you might need to enable the security option, TLS 1.2. For more information, go to the Cloud Application Performance Management Forum and search on "tls".

Thresholds and resource groups

Thresholds test for certain conditions, such as number of transactions per minute fewer than 100, and open an event when the conditions have been met. Use thresholds to monitor for real and potential issues with your monitored resources. Assign thresholds to resource groups for monitoring on all managed systems of the same type the belong to the group.

Background information

Review the background information to learn about thresholds, predefined thresholds for your agents, the resource groups that they are assigned to, and customizing thresholds.

Predefined thresholds

Your monitoring agents come with *predefined thresholds* that are enabled and started with the agent. The first time that you open the **Threshold Manager** after agent installation, the thresholds that are listed for the selected data source type are the predefined thresholds. These predefined thresholds are assigned to the default system resource group for the agent and shown in the **Assigned groups** column.

If you edit a predefined threshold, such as to change the name or condition, the threshold is no longer treated as a predefined threshold but considered a *custom threshold*. However, you can change the assigned resource group for a predefined threshold from the default system group to a user-defined group and it remains a predefined threshold.

If you prefer not to use the predefined thresholds, you can turn them off in the **Advanced Configuration** page (see “Thresholds Enablement” on page 1051). Disabling the predefined thresholds doesn't remove them from the **Threshold Manager**; it only removes their group assignment, rendering them inactive. After disabling the predefined thresholds, you can open the **Threshold Manager** and see that the **Assigned groups** column is empty for every predefined threshold (see “Examples of disabled thresholds” on page 959).

You can enable the threshold as a custom threshold by assigning it to any available resource group.

Custom thresholds

New thresholds that you create are custom thresholds, as indicated in the **Threshold Manager Origin** column. If you edit a predefined threshold, it also becomes a custom threshold and its origin changes from "Predefined" to "Custom".

Execute command

After an event is opened for a threshold that evaluates to true, you can have a command or script of commands run automatically. For example, you might want to log information, trigger an audible beep, or stop a job that is overusing resources when an event is opened. The command or script is run on the system of the monitoring agent that opened the event.

The command uses the following syntax:

```
&{data_set.attribute}
```

where *data_set* is the data set name and *attribute* is the attribute name as shown in the Threshold Editor. If the data set or attribute name contains a space, replace with an underscore. The *data_set* must be the same data set that you select in the Data set selection field.

The following example shows how you can pass the disk name parameter to your managed resource:

```
/scripts/clean_logs.sh &{KLZ_Disk.Disk_Name}
```

You can pass in one or more attributes from the data set. If specified, multiple attributes are passed into the command in order ($1, $2, and so on).

You must ensure the script or programs executed by the command are installed on the agent system since Cloud APM does not provide a mechanism to distribute scripts or programs. The command runs from the command line with the same user account that the agent was started with. Ensure the user that starts the agent has permission to execute the command. For example, if the agent is running as root, then root runs the command on the managed system.

The following options control how often the command is run:

Select □ **On first event only** if the data set returns multiple rows and you want to run the command for only the first event occurrence in the data sample. Clear the check box to run the command for every row that causes an event.

Select □ **For every consecutive true interval** to run the command every time the threshold evaluates to true. Clear the check box to run the command when the threshold is true, but not again until the threshold evaluates to false, followed by another true evaluation in a subsequent interval.

Resource groups

Resource groups represent a collection of managed systems and control how thresholds are distributed. You assign a threshold to the resource group that includes the managed systems where you want it to run.

All predefined thresholds have a default resource group assignment, which is the system defined group for the agent type, such as Db2 and Microsoft IIS.

You can create custom resource groups and select the managed systems to include in each group. You can have multiple agent types in a custom resource group; thresholds that are assigned to the group are distributed only to the managed systems of the same agent type. For example, a threshold that is created with Linux OS attributes and assigned to a resource group of Linux OS, MongoDB, and Python managed systems, is distributed to only the Linux OS managed systems.

For more information, see “Resource Group Manager” on page 962.

Application Performance Dashboard event status

The status severities that are shown in the Application Performance Dashboard indicate the highest event severity of the selected application, group, subgroup, and managed system instance.

After you select an application from the navigator or from a summary box in the **All My Applications** dashboard, a tabbed dashboard presents different facets of your application. The **Events** tab provides information about the events for the selected navigator item, as described in “Event Status” on page 1083.

Threshold changes affect other thresholds that are assigned to the same monitoring agent

After you create, modify, or delete a threshold definition or change the list of thresholds that are distributed to a monitoring agent, all sampled events are closed for the agents that the threshold is distributed to. After the event closure, the monitoring agents reopen events for any threshold conditions that evaluate to true. On the Cloud APM console, the closed events disappear from the console until they are reopened with a new **Timestamp** value. If you are receiving email notifications for events, you receive close event and open event email notifications.

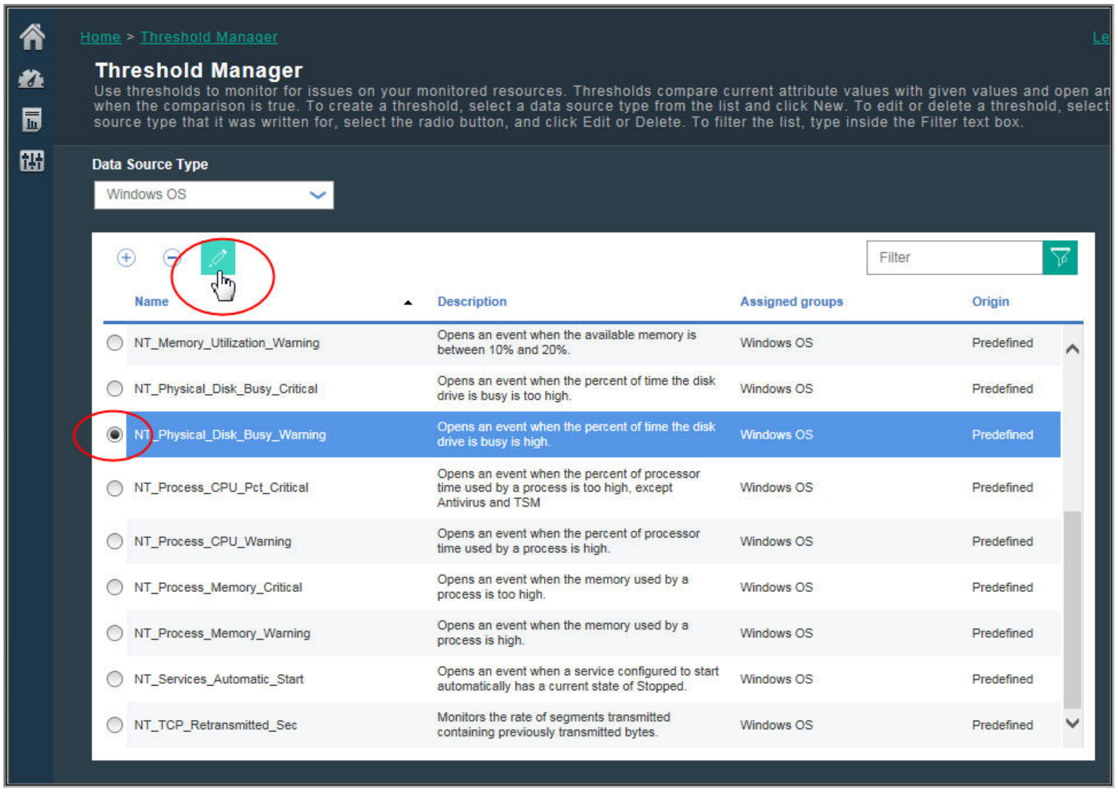
Consider, for example, that you have a custom resource group named Site Systems with Linux OS and WebSphere Applications thresholds and agents assigned. You create a new Linux OS threshold and assign it to Site Systems. Any open sampled events on the Linux OS agents that are assigned to Site Systems are closed. Then the sampled events are reopened if the threshold conditions are still true.

Examples of disabled thresholds

You can disable the predefined thresholds for all agents in your environment. You can also disable thresholds individually, whether predefined or custom. When a threshold is disabled it is not running on managed systems and no events are opened.You disable a threshold by removing its resource group assignment (or assignments). An **Advanced Configuration** setting is also available for disabling the predefined thresholds for all agents.

Disabling a single threshold

In this image, the threshold to be disabled is selected in the **Threshold Manager** and the user clicks ✎ **Edit**:



Home > Threshold Manager

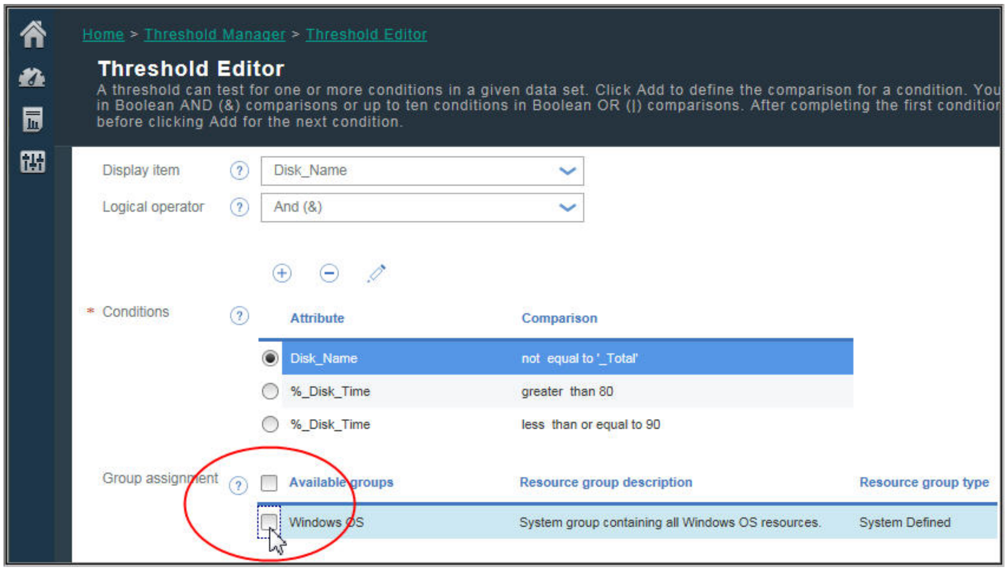
Threshold Manager

Use thresholds to monitor for issues on your monitored resources. Thresholds compare current attribute values with given values and open an
when the comparison is true. To create a threshold, select a data source type from the list and click New. To edit or delete a threshold, select
source type that it was written for, select the radio button, and click Edit or Delete. To filter the list, type inside the Filter text box.

Data Source Type: Windows OS

	Name	Description	Assigned groups	Origin
○	NT_Memory_Utilization_Warning	Opens an event when the available memory is between 10% and 20%.	Windows OS	Predefined
○	NT_Physical_Disk_Busy_Critical	Opens an event when the percent of time the disk drive is busy is too high.	Windows OS	Predefined
●	NT_Physical_Disk_Busy_Warning	Opens an event when the percent of time the disk drive is busy is high.	Windows OS	Predefined
○	NT_Process_CPU_Pct_Critical	Opens an event when the percent of processor time used by a process is too high, except Antivirus and TSM	Windows OS	Predefined
○	NT_Process_CPU_Warning	Opens an event when the percent of processor time used by a process is high.	Windows OS	Predefined
○	NT_Process_Memory_Critical	Opens an event when the memory used by a process is too high.	Windows OS	Predefined
○	NT_Process_Memory_Warning	Opens an event when the memory used by a process is high.	Windows OS	Predefined
○	NT_Services_Automatic_Start	Opens an event when a service configured to start automatically has a current state of Stopped.	Windows OS	Predefined
○	NT_TCP_Retransmitted_Sec	Monitors the rate of segments transmitted containing previously transmitted bytes.	Windows OS	Predefined

The threshold is opened in the Threshold Editor. The user clears the check box of the assigned resource group in the **Group assignment** field:



Home > Threshold Manager > Threshold Editor

Threshold Editor

A threshold can test for one or more conditions in a given data set. Click Add to define the comparison for a condition. You
in Boolean AND (&) comparisons or up to ten conditions in Boolean OR (|) comparisons. After completing the first conditior
before clicking Add for the next condition.

Display item: Disk_Name

Logical operator: And (&)

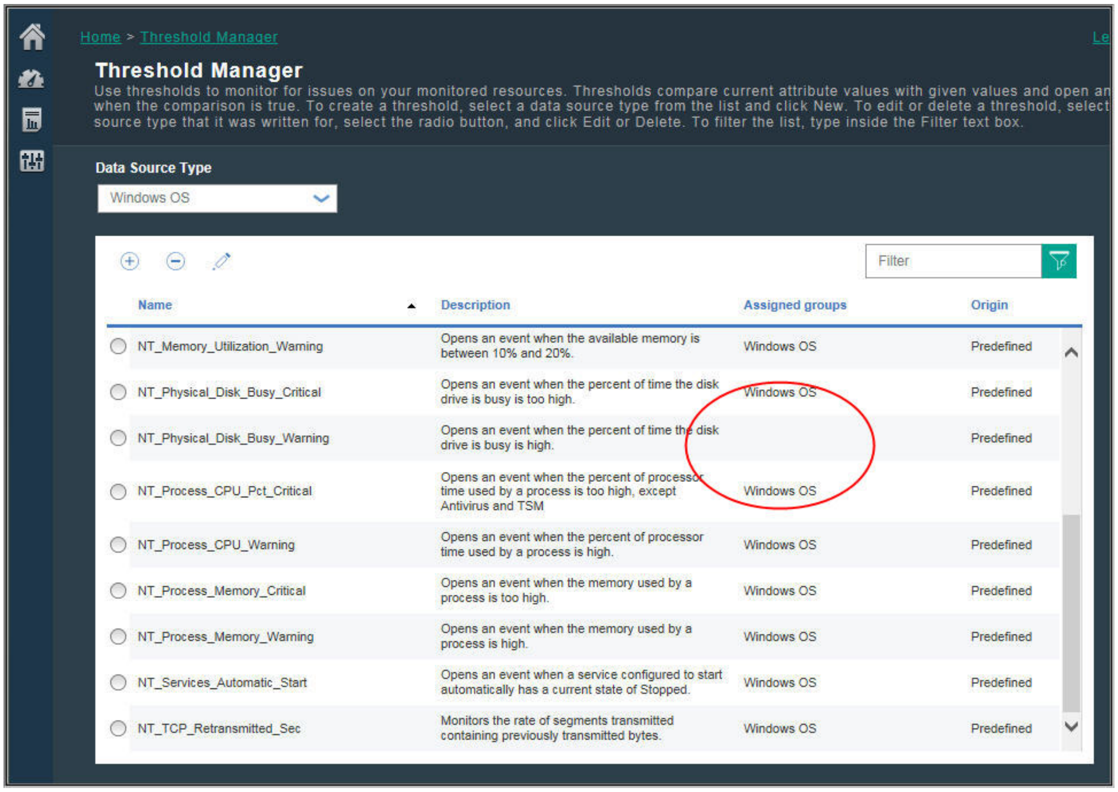
* Conditions

	Attribute	Comparison
●	Disk_Name	not equal to '_Total'
○	%_Disk_Time	greater than 80
○	%_Disk_Time	less than or equal to 90

Group assignment

□	Available groups	Resource group description	Resource group type
□	Windows OS	System group containing all Windows OS resources.	System Defined

After the user clicks **Save**, the **Threshold Manager** is displayed. The threshold is disabled, and the **Assigned groups** column is empty:



Home > Threshold Manager

Threshold Manager

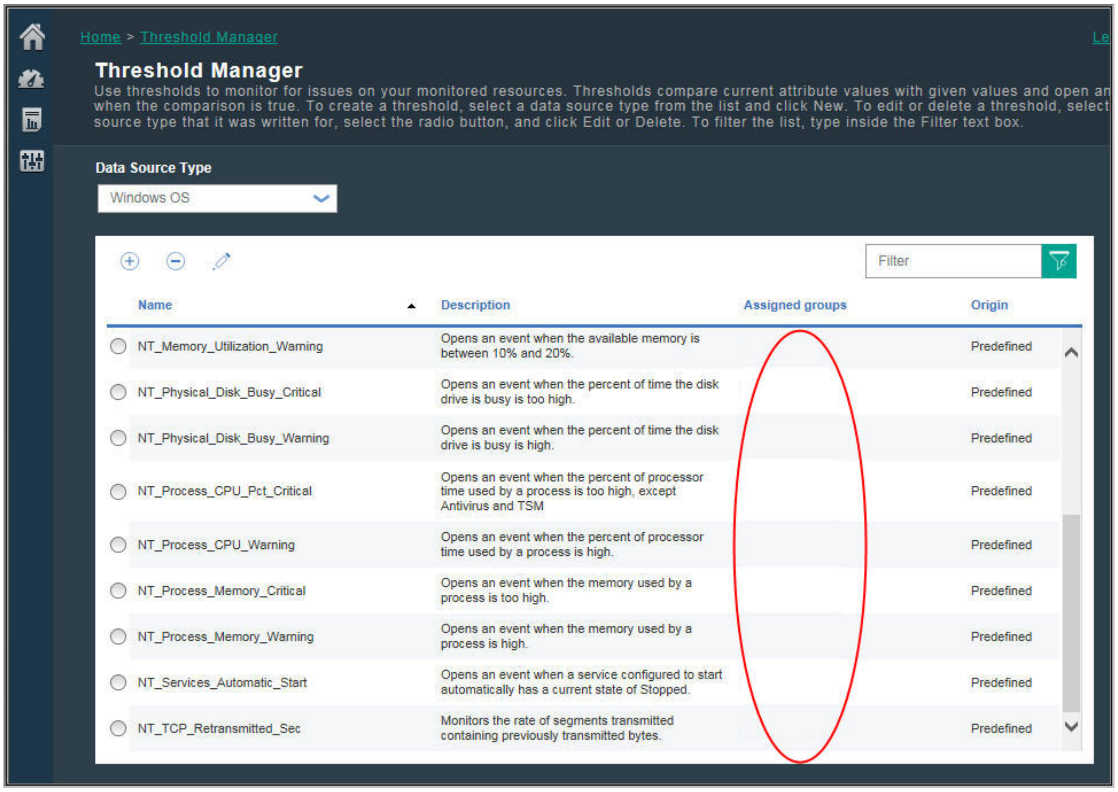
Use thresholds to monitor for issues on your monitored resources. Thresholds compare current attribute values with given values and open an when the comparison is true. To create a threshold, select a data source type from the list and click New. To edit or delete a threshold, select source type that it was written for, select the radio button, and click Edit or Delete. To filter the list, type inside the Filter text box.

Data Source Type: Windows OS

	Name ▲	Description	Assigned groups	Origin
○	NT_Memory_Utilization_Warning	Opens an event when the available memory is between 10% and 20%.	Windows OS	Predefined
○	NT_Physical_Disk_Busy_Critical	Opens an event when the percent of time the disk drive is busy is too high.	Windows OS	Predefined
○	NT_Physical_Disk_Busy_Warning	Opens an event when the percent of time the disk drive is busy is high.		Predefined
○	NT_Process_CPU_Pct_Critical	Opens an event when the percent of processor time used by a process is too high, except Antivirus and TSM	Windows OS	Predefined
○	NT_Process_CPU_Warning	Opens an event when the percent of processor time used by a process is high.	Windows OS	Predefined
○	NT_Process_Memory_Critical	Opens an event when the memory used by a process is too high.	Windows OS	Predefined
○	NT_Process_Memory_Warning	Opens an event when the memory used by a process is high.	Windows OS	Predefined
○	NT_Services_Automatic_Start	Opens an event when a service configured to start automatically has a current state of Stopped.	Windows OS	Predefined
○	NT_TCP_Retransmitted_Sec	Monitors the rate of segments transmitted containing previously transmitted bytes.	Windows OS	Predefined

Disabling all predefined thresholds

Turn off all predefined thresholds for all monitoring agents in **Advanced Configuration** page, as described in "Thresholds Enablement" on page 1051. When you next open the **Threshold Manager**, the **Assigned groups** column is empty for every predefined threshold, indicating that the thresholds are inactive:



Home > Threshold Manager

Threshold Manager

Use thresholds to monitor for issues on your monitored resources. Thresholds compare current attribute values with given values and open an when the comparison is true. To create a threshold, select a data source type from the list and click New. To edit or delete a threshold, select source type that it was written for, select the radio button, and click Edit or Delete. To filter the list, type inside the Filter text box.

Data Source Type: Windows OS

	Name ▲	Description	Assigned groups	Origin
○	NT_Memory_Utilization_Warning	Opens an event when the available memory is between 10% and 20%.		Predefined
○	NT_Physical_Disk_Busy_Critical	Opens an event when the percent of time the disk drive is busy is too high.		Predefined
○	NT_Physical_Disk_Busy_Warning	Opens an event when the percent of time the disk drive is busy is high.		Predefined
○	NT_Process_CPU_Pct_Critical	Opens an event when the percent of processor time used by a process is too high, except Antivirus and TSM		Predefined
○	NT_Process_CPU_Warning	Opens an event when the percent of processor time used by a process is high.		Predefined
○	NT_Process_Memory_Critical	Opens an event when the memory used by a process is too high.		Predefined
○	NT_Process_Memory_Warning	Opens an event when the memory used by a process is high.		Predefined
○	NT_Services_Automatic_Start	Opens an event when a service configured to start automatically has a current state of Stopped.		Predefined
○	NT_TCP_Retransmitted_Sec	Monitors the rate of segments transmitted containing previously transmitted bytes.		Predefined

Related concepts

"Background information" on page 958

Review the background information to learn about thresholds, predefined thresholds for your agents, the resource groups that they are assigned to, and customizing thresholds.

Related reference

"Threshold Manager" on page 967

Resource Group Manager

Your monitored environment might have multiple managed systems that can be categorized by their purpose. Such systems often have the same threshold requirements. Use the **Resource Group Manager** to organize managed systems into groups that you can assign thresholds to. You can also create resource groups that correlate with your role-based access control (RBAC) policies.

After you click **System Configuration** > **Resource Group Manager**, the page opens with a table of defined resource groups. Initially, one predefined system group is shown for each monitoring agent type that is installed, such as `Windows OS`. Each system group contains all the predefined thresholds for the agent.

Your access to the **Resource Group Manager** and resource groups is controlled by your user permissions. You must have View permission for a resource group to see it; you must have Modify permission to create, edit, or delete a resource group.

The table has tools for managing resource groups:

- ⊕**New** opens the **Resource Group Editor** for assigning managed systems and thresholds.
- Select a resource group to see the assigned resources and thresholds that are assigned to the group in the adjacent pane.
- Select a resource group and click **Edit** to open the **Resource Group Editor** for changing the managed system and threshold assignments.
- Select a resource group that you no longer want and click ⊖**Delete**. After you confirm the deletion, any thresholds that were assigned to the group must be assigned to another group if you want them to continue to run on your managed systems.
- You can click inside the filter text box and type the value to filter by. As you type, the rows that do not fit the criteria are filtered out. To clear the filter, click the × in the filter box or press the Backspace key.

The table displays the available resource groups:

Resource group name
Predefined groups are named for their agent type; custom groups are named by the author.

Resource group description
A predefined group is described as a *system group* for the monitored resource; custom groups are described by the author.

A system group, such as `Linux OS`, includes all the predefined thresholds for the agent and all managed systems where the agent is installed. You can edit a system group to assign or remove thresholds but you cannot assign or remove managed systems. Managed systems are automatically assigned to a system group of the same type, including any from your Tivoli Monitoring domain if you have a Hybrid Gateway configured.

Some system resource groups relate to agents that support subnodes. Depending on the agent type, the subnodes, the agent node, or both can be added to applications. If only the subnodes can be added to defined applications, you are not able to see events for any thresholds that were defined for the agent node. However, the events can be forwarded to an event manager such as Netcool/OMNIbus. As well, IBM Cloud Application Performance Management subscribers can set up Alert Notification.

Resource group type
Predefined groups are type *System Defined*. You have a predefined group for every type of agent that you have installed in your environment.

Custom groups that you or others in your environment create are type *User Defined*.

Resource Group Editor

After you click ⊕ **New** to add a group, or after you select a group and click ✎ **Edit** to edit a group, the **Resource Group Editor** is displayed with the following fields:

Group name
The name of the group is required. You can change an existing custom group name, and all references to the group are updated automatically after you save your changes.

Group description
Optional for custom groups. Add a description of the group organization. The description is displayed in the **Resource Group Manager**.

Resource assignment
All the managed systems that are available for adding to the group are shown in the agent list by their managed system name, host name, agent type, and their domain. You can click a column heading to sort the list by agent name, host name, type, or domain.

To populate the group, select the check box of one or more managed systems.

You can select □ **Show only selected resources** to hide the unassigned managed systems.

If you have configured the IBM Cloud Application Performance Management Hybrid Gateway, you can add managed systems from your IBM Tivoli Monitoring domain to user defined resource groups. You cannot add Tivoli Monitoring managed systems to system defined groups nor can you create thresholds for them.

Threshold assignment
All the thresholds that are predefined or were added through the **Threshold Manager** are shown in the threshold list by their name and agent type. You can click a column heading to sort the list.

To add a threshold to the group, select the check box next to the name; to remove a threshold from the group, clear the check box. You must have View permission for the **Threshold Manager** to add or remove thresholds. When adding thresholds to a system group, the available thresholds are limited to those whose data set is suitable for the system group.

The thresholds that you assign to the group are distributed to every managed system in the group of the same agent type. Although you can assign thresholds of any monitoring agent type to a group, the assigned thresholds are distributed only to managed systems of the same type that are members of the group. For example, if you assign the `MySQL_Process_Down` threshold to the group, it is included in the group, but is distributed only to the Monitoring Agent for MySQL managed systems that belong to the group.

You can select □ **Show only selected thresholds** to hide the unassigned thresholds. If you are filtering the list, click the × in the filter box to clear the filter and enable the check box.

You can also assign a resource group to a threshold from the **Threshold Manager**.

Configure email notification
Available with IBM Cloud Application Performance Management: Click **Configure email notification** to open the IBM Alert Notification application in a new browser tab or window. Use Alert Notification to create users and associate their email addresses with resource groups for receiving event notifications by email.

Attention: If your browser does not allow pop-up windows, the Alert Notification window is prevented from opening. You must set the browser to allow the Alert Notification window to open it for configuring email notification for a resource group.

After you click **Save**, the resource group is saved with the list of resource groups and displayed in the **Resource Group Manager** table.

Related tasks

"Integrating with Alert Notification" on page 953

"Exploring the APIs" on page 1049

Related reference

"Threshold Manager" on page 967

Tutorial: Defining a threshold

Thresholds are the alerting mechanism for potential and actual problems with your managed resources. Use the tutorial to learn the basic steps for defining a threshold to raise an alarm when the condition occurs.

About this task

This tutorial uses the Linux OS agent for showing you how to define a threshold in the **Threshold Manager** and view the raised alarm in the Application Performance Dashboard. Your user ID must have View permission for the **Threshold Manager** to complete these steps.

Procedure

1. From the navigation bar, click **System Configuration** > **Threshold Manager**.
2. Click the **Data Source Type** list box and select the **`Linux OS`** data type.

   The thresholds that were defined for the Linux OS agent are displayed in the table.
3. Click ⊕ **New** to open the **Threshold Editor** for defining the threshold.
4. Define a threshold to raise an alarm of ✓ Unknown severity when the average CPU is under 75%:
   a) In the **Name** field, enter `CPU_average_below_75_percent`.

   b) In the **Description** field, enter `Threshold tutorial`

   c) Leave the **Severity**, **Interval**, and **Required consecutive samples** fields at their default values.

   d) In the **Data set** field, select **`KLZ CPU Averages`**.

   e) In the **Conditions** field, click ⊕**New** and add the comparison in the dialog box that pops up:

      1) In the **Attribute** field, select **`CPU_Usage_Current_Average`**

      2) In the **Operator** field, select **`Less than`**

      3) In the **Value** field, enter 75

      After you click **OK**, the attribute and comparison are displayed in the **Conditions** field.

   f) In the **Group assignment** field, select the **`Linux OS`** system group.

   g) Click **Save** to complete the definition and return to the **Threshold Manager** page.

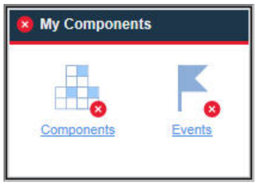
   `CPU_average_below_75_percent` is displayed in the list of thresholds that are defined for the Linux OS data source.

Results

You defined a threshold that raises an alarm when the average CPU usage on any of your Linux OS managed systems is under 75%.

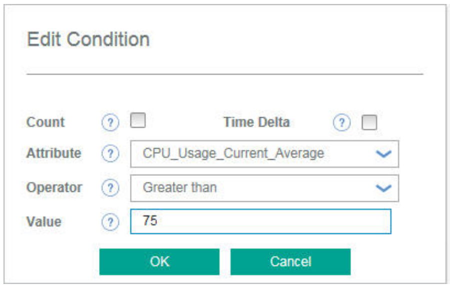
What to do next

- View the event:
  1. From the navigation bar, click **Performance** > **Application Performance Dashboard**.
  2. In the **My Components** summary box, click the **Events** link.

  3. In the **Events** tab that opens, look for the `CPU_average_below_75_percent` threshold in the list. It can take 1 or 2 minutes for the alarm to be raised. If the CPU average is over 75 percent, however, no alarm is raised.
- Edit the threshold:

1. From the navigation bar, click **System Configuration** > **Threshold Manager**.
2. Click the **Data Source Type** list box and select the **`Linux OS`** data type.
3. Select the `CPU_average_below_75_percent` threshold from the list and click **Edit**.
4. Change any of the fields for a condition that you want to be alerted of. For example, you might change **Name** to `CPU_high_warning`, change **Severity** to **`Warning`**, edit the condition to be **`Greater than 75`**, and add a condition: **`Less than 95`**. To edit the comparison, click **Edit**, change **Operator** to **`Greater than`** and click **OK**. To add the new condition, leave the **Logical operator** field at the default **`And (&)`**, click **New** for a new condition, select **Operator Less than**, and enter **Value** 95.



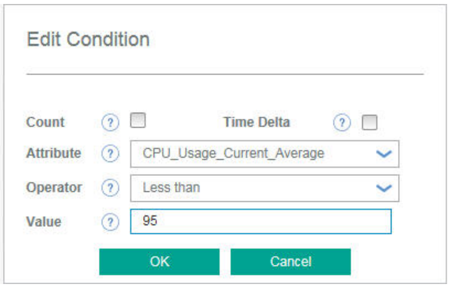
Edit Condition

Count ? □ Time Delta ? □

Attribute ? CPU_Usage_Current_Average

Operator ? Greater than

Value ? 75



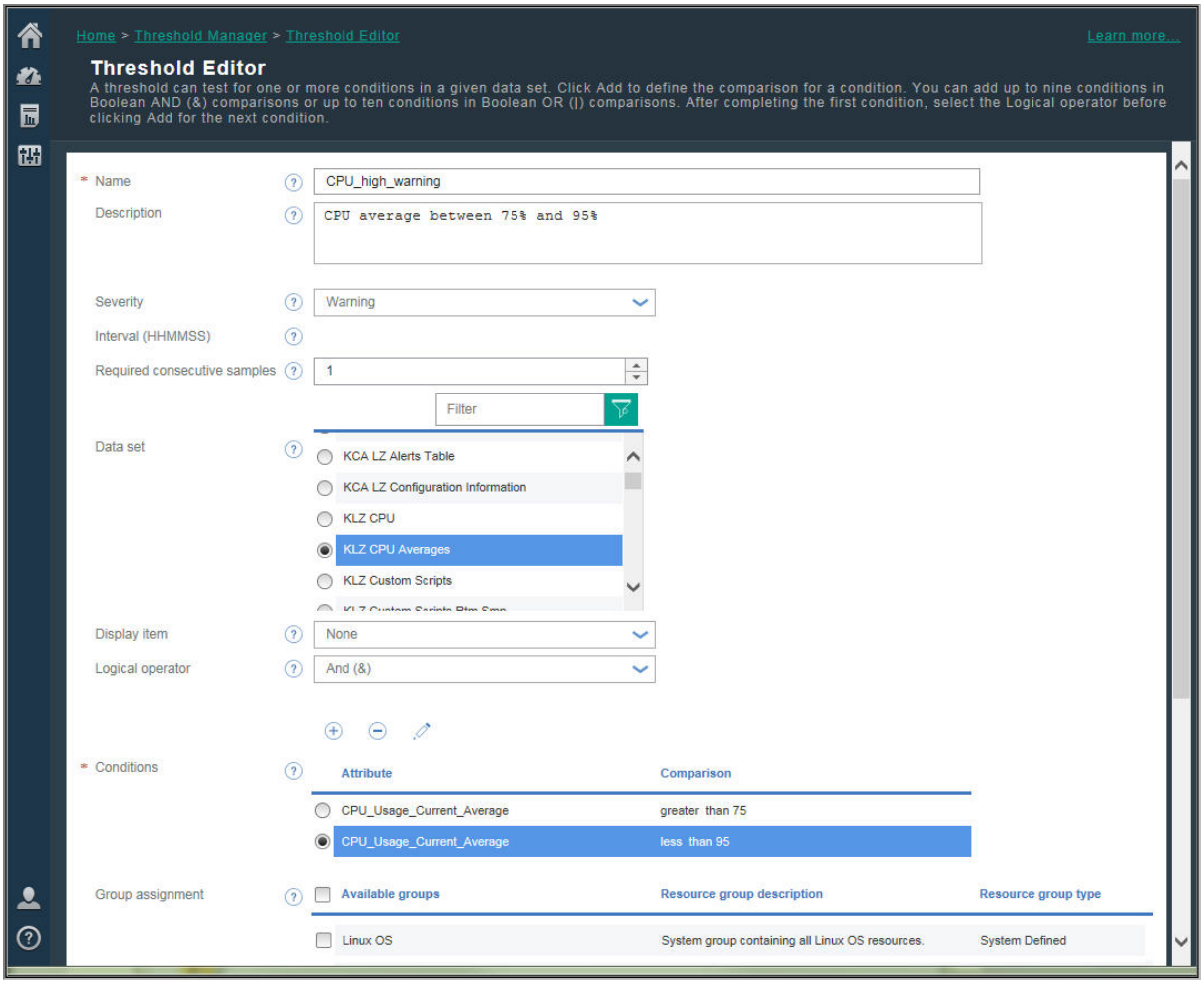
Edit Condition

Count ? □ Time Delta ? □

Attribute ? CPU_Usage_Current_Average

Operator ? Less than

Value ? 95



Threshold Editor

A threshold can test for one or more conditions in a given data set. Click Add to define the comparison for a condition. You can add up to nine conditions in Boolean AND (&) comparisons or up to ten conditions in Boolean OR (|) comparisons. After completing the first condition, select the Logical operator before clicking Add for the next condition.

* Name ? CPU_high_warning

Description ? CPU average between 75% and 95%

Severity ? Warning

Interval (HHMMSS) ?

Required consecutive samples ? 1

Data set ? Filter

- ○ KCA LZ Alerts Table
- ○ KCA LZ Configuration Information
- ○ KLZ CPU
- ● KLZ CPU Averages
- ○ KLZ Custom Scripts
- ○ KLZ Custom Scripts Rtm Smp

Display item ? None

Logical operator ? And (&)

* Conditions ?

	Attribute	Comparison
○	CPU_Usage_Current_Average	greater than 75
●	CPU_Usage_Current_Average	less than 95

Group assignment ?

□	Available groups	Resource group description	Resource group type
□	Linux OS	System group containing all Linux OS resources.	System Defined

- Review the predefined thresholds for your agents and adjust any comparison values as needed for your environment.
- Create new thresholds to raise alarms on other conditions that you want to be alerted of.

Related reference

"Threshold Manager" on page 967

Tutorial: Defining a threshold to run a command on the managed resource

You can use the **Threshold Editor** to pass certain parameters to agents. You can specify commands or a script of commands to run automatically when an event is triggered.

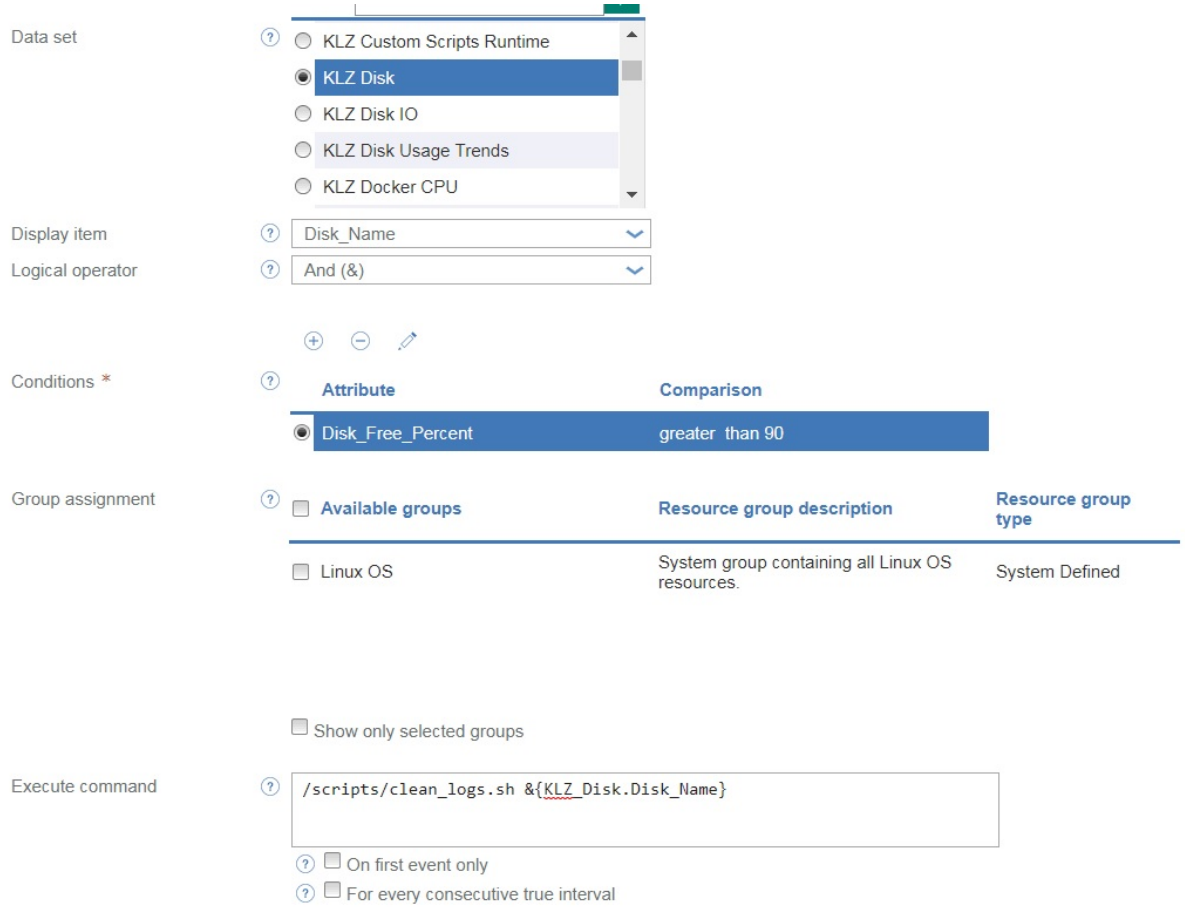
About this task

This tutorial shows you how to use the **Execute Command** field to pass a parameter to your IBM Cloud Application Performance Management agent.

Procedure

1. Open the **Threshold Manager** by selecting **System Configuration** > **Threshold Manager**.
2. Select *Linux OS* from the **Data Source Type** field.

   The thresholds that were defined for the Linux OS agent are displayed in the table.
3. Click ⊕ **New** to open the **Threshold Editor** for defining the threshold.
4. Define the threshold and conditions by specifying values for the various parameters, such as **Name**, **Severity**, and **Conditions**.
5. Select *KLZ Disk* from the **Data set** field.



Data set ?
○ KLZ Custom Scripts Runtime
● KLZ Disk
○ KLZ Disk IO
○ KLZ Disk Usage Trends
○ KLZ Docker CPU

Display item ? Disk_Name

Logical operator ? And (&)

Conditions * ?

	Attribute	Comparison
●	Disk_Free_Percent	greater than 90

Group assignment ?

□	Available groups	Resource group description	Resource group type
□	Linux OS	System group containing all Linux OS resources.	System Defined

□ Show only selected groups

Execute command ? /scripts/clean_logs.sh &{KLZ_Disk.Disk_Name}

? □ On first event only
? □ For every consecutive true interval

6. Enter the following command in the **Execute Command** field:

   ```
   /scripts/clean_logs.sh &{KLZ_Disk.Disk_Name}
   ```

   You must replace the space in the *KLZ Disk* data set name with an underscore.

   Note: The data set name referenced in the **Execute Command** field must be the same data set that is selected in step 5.

   The *`KLZ_Disk.Disk_Name`* is passed into the command script.

Results

Your script or command is set up to automatically run for your defined threshold. You must ensure the script or programs executed by the command are installed on the agent system since Cloud APM does not provide a mechanism to distribute scripts or programs.The command or script is run on the system of the monitoring agent that is monitoring the threshold condition. The user who starts the monitoring agent must have permission to execute the script or command.

Related reference

"Threshold Manager" on page 967

Threshold Manager

Use the **Threshold Manager** to review the predefined thresholds for a monitoring agent and to create and edit thresholds.Thresholds are used to compare the sampled value of an attribute with the value set in the threshold. If the sampled value satisfies the comparison, an event is opened. The event closes automatically when the threshold comparison is no longer true.

After you click **System Configuration** > **Threshold Manager**, the page is displayed with a table of the thresholds that were defined for the selected data source type.

The data types that display when you click the **Data Source Type** list box are for the types of monitoring agents and data collectors that are installed in your managed environment. Select the data type for which you want to create or view thresholds.

The table lists all the thresholds that were created for the selected data type, and has tools for managing thresholds:

- **New** opens the **Threshold Editor** for defining a threshold for the selected data type.
- Select a threshold and click **Edit** to open the **Threshold Editor** for editing the definition.
- Select a threshold that you no longer want and click **Delete**. After you confirm that you want to delete the threshold, it is removed from the list and from any resource groups that it was assigned to. Any open events for the threshold are closed.
- For a long list, you can click inside the filter text box and type the beginning of the value to filter by. As you type, the rows that do not fit the criteria are filtered out. To clear the filter, click the × in the filter box or press the Backspace key.

For more information about the predefined thresholds and custom thresholds that are displayed in the table and the significance of the resource group assignment (or lack thereof), see "Background information" on page 958. For a quick hands-on lesson, see "Tutorial: Defining a threshold" on page 964.

Threshold Editor

After you click **New** or select a threshold and click **Edit**, the Threshold Editor is displayed with the following fields:

Name

Enter a unique name for the threshold. The name must begin with a letter and can be up to 31 letters, numbers, and underscores, such as "Average_Processor_Speed_Warning". The threshold name is displayed in the Application Performance Dashboard **Events** tab and in certain dashboard tables.

Description

Optional. A description is useful for recording the purpose of the threshold that users can see in the **Threshold Manager**.

Severity

Select the appropriate event severity from the list: Fatal, Critical, Minor, Warning, or Unknown.

The severities are consolidated for display in the Application Performance Dashboard: Fatal and Critical events show as ⊗; Minor and Warning events show as ⚠; and Unknown events show as ✅ (see "Event Status" on page 1083).

Forward EIF Event?

If you configured event forwarding in the **System Configuration** > **Advanced Configuration** page (“Event Manager” on page 1050), open events are forwarded by default to the event destinations that you configured, for example, EIF event targets, Cloud Event Management, or Alert Notification. Change the setting to **No** if you do not want to forward events for this threshold to any event destinations.

If you configured event forwarding in the **System Configuration** > **Advanced Configuration** page (Event Manager), open events are forwarded to an EIF receiver by default. Change the setting to **No** if you do not want to forward events for this threshold to an EIF receiver.

To customize how thresholds are mapped to forwarded events, thus overriding the default mapping between thresholds and events forwarded to the event server, click **EIF Slot Customization**. For more information, see “Customizing an event to forward to an EIF receiver” on page 972.

Interval

Enter or select the time to wait between taking data samples in *HHMMSS* format, such as 00 15 00 for 15 minutes. For sampled-event thresholds, the minimum interval is 000030 (30 seconds) and the maximum is 235959 (23 hours, 59 minutes, and 59 seconds).

A value of 000000 (six zeros) indicates a *pure event* threshold. Pure events are unsolicited notifications. Thresholds for pure events have no sampling interval, thus they have no constant metric that can be monitored for current values. Pure events are closed after 24 hours or as set in the **Advanced Configuration** page **Pure Event Close Time** field in category “Event Manager” on page 1050.

Required consecutive samples

Specify how many consecutive threshold samples must evaluate to true before an event is generated: For any threshold with a setting of 1 and a sample that evaluates to true, an event is generated immediately; a setting of 2 means that two consecutive threshold samples must evaluate to true before an event is opened.

Data set

Select the data set (attribute group) for the type of data to be sampled. The attributes that are available for inclusion in the condition are from the chosen data set. If the threshold has multiple conditions, they must all be from the same data set.

To get a short description of a data set, hover the mouse over the name. You can get the complete description of the data set and attributes by clicking the "Learn more" link in the hover help. You can also click **Help** > **Help Contents** or **Help** > **Documentation** in the navigation bar, and open the help or download the reference for the monitoring agent.

Some agents are categorized as multi-node agents, which have subnodes for monitoring multiple agent resources. A multi-node agent might have data sets that can be used in a threshold but any events opened for the threshold do not display in the Application Performance Dashboard. A message notifies you of the limitation. Such events can be forwarded to the IBM Netcool/OMNIbus event manager.

Display item

Optional. For multiple row data sets only. After a row evaluation causes an event to open, no more events can be opened for this threshold on the monitored system until the event is closed. By selecting a display item, you enable the threshold to continue evaluating the other rows in the data sampling and open more events if other rows qualify. As well, the display item is shown in the **Events** tab of the Application Performance Dashboard so that you can easily distinguish among the rows for which events were opened. The list contains only the attributes that you can designate as display items.

Logical Operator

Ignore this field if your threshold has only one condition. If you are measuring multiple conditions, select one of the following operators before you click **New** to add a second or third (or more) condition:

And (&) if the previous condition and the next condition must be met for the threshold to be breached

Or (|) if either of them can be met for the threshold to be breached

A mix of logical operators is not supported; use either all And operators or all Or operators. The threshold can have up to nine conditions when the Or operator is used; up to 10 conditions when the And operator is used.

If you are using the `Missing` function (described later in the **Operator** section), you can use only the And operator in the formula.

Conditions

The threshold definition can logically include multiple simultaneous thresholds or conditions.

Click ⊕ **New** to add a condition. Select a condition and click ✎**Edit** to modify the expression, or click ⊖ **Delete** to remove the expression.

After you click ⊕ **New** or ✎**Edit**, complete the fields in the **Add Condition** or **Edit Condition** dialog box that opens:

Count □

For data sets that return multiple rows for each data sample, you can have each row that meets the criteria of the condition counted. An event is opened after the count **Value** is reached and any other conditions in the formula are met. For example, if the number of "zombie" processes exceeds 10, issue an alert.

In the following example, the condition is true when more than 10 rows are counted: **Attribute** Timestamp, **Operator** Greater Than, **Value** 10.

Select the **Count** check box, the **Attribute** to be counted, the relational **Operator**, and count **Value**.

If the formula has multiple conditions, each condition must use the **And** Boolean operator. **Count** and **Time Delta** are mutually exclusive: If you select the check box for one function, the other function is disabled. The attribute cannot be a system identifier, such as Server Name or ORIGINNODE, be specified as the **Display Item**, or be from a data set for which the threshold opens pure events.

Time Delta □

Use the **Time Delta** function in a condition to compare the sampled time stamp (such as recording time) with the specified time difference.

After you select the **Time Delta** check box, the Time Delta field is displayed for you to combine + (plus) or **-** (minus) with the number of Days, Hours, Minutes, or Seconds. Select **Sampled Time** or **Specific Time** as the Value to use in the comparison.

In the following Event Log example, the formula compares the time that the event was logged with the time stamp from the data sampling. If the event occurred seven days earlier, the comparison is true. If the relational operator was changed to Less Than or Equal, the comparison would be true after 8 days, 9 days, and so on:

Attribute Timestamp
Time Delta -7 Days
Operator Equal
Value Entry Time

Attribute

Select the attribute that you want to compare in this condition. To see a short description of the attribute, hover the mouse over the name in the list.

Operator

Select the relational operator for the type of comparison:

```
Equal
Not Equal
```

`Greater than`
`Greater than or Equal`
`Less than`
`Less than or equal`
`Regular expression contains`
`Regular expression does not contain`

`Regular expression contains` and `Regular expression does not contain` look for a pattern match to the expression. The easier it is to match a string with the expression, the more efficient the workload at the agent. The expression does not need to match the entire line; only the substring in the expression. For example, in `See him run`, you want to know if the string contains `him` You could compose the regular expression using `him` , but you could also use `.*him.*`. Or if you are looking for See, you could enter See, or you could enter ^See to confirm that it's at the beginning of the line. Entering `.*` wildcards is a less efficient search and raises the workload. For more information about regular expressions, see the developerWorks® technical library topic or search `regex` in your browser.

You can also select the `Missing` function, which compares the value of the specified metric with a list of values that you supply. The condition is true when the value does not match any in the list. This function is useful when you want notification that something is not present in your system. Requirements and restrictions:

1. The selected metric must be a text attribute: time and numeric attributes cannot be used.
2. Separate each value with a comma (,), for example, `fred,mary,jean`.
3. You can have only one `Missing` condition in a threshold.
4. `Missing` must be the last condition in the formula. If other conditions are required, enter them before you add the Missing function and use only the **And (&)** operator in the formula. Otherwise, all subsequent rows are disabled.

Value

Enter the value to compare by using the format that is allowed for the metric, such as 20 for 20% or 120 for 2 minutes.

Group assignment

Assign a resource group to distribute the threshold to the managed systems of the same type within the resource group. The resource groups that are available are the user defined groups that you have Modify permission for and the system groups (for the agent type) that you have View permission for. The available system groups are also limited to those that are suitable for the chosen data set.

A threshold with no group assigned is distributed to no monitored systems and remains stopped until it is distributed to a resource group.

A system group, such as `Linux OS` or `HTTP Server`, distributes the threshold to all managed systems where that agent is installed. By default, every predefined threshold is assigned to the system group for that agent. (You can disable all predefined thresholds in the **Advanced Configuration** page, as described in "Thresholds Enablement" on page 1051.)

The exception is managed systems from the IBM Tivoli Monitoring domain: Managed systems from the Tivoli Monitoring domain must be monitored with situations that were distributed in your Tivoli Monitoring environment.

To assign groups to the threshold, select the check box of one or more resource groups. If the list of assigned groups is long, you can select ☐ **Show only selected groups**.

If you do not see a resource group that you want to assign the threshold to, you can save the threshold definition, and click **OK** when prompted to confirm that you want to save the threshold without assigning it to a group. You can then create a new group in the **Resource Group Manager**, and assign a threshold to the new group in the **Resource Group Editor**. For more information, see "Resource Group Manager" on page 962.

Execute command

After an event is opened for a threshold that evaluates to true, you can have a command or script of commands run automatically. For example, you might want to log information, trigger an audible beep, or stop a job that is overusing resources when an event is opened. The command or script is run on the system of the monitoring agent that opened the event.

The command uses the following syntax:

```
&{data_set.attribute}
```

where *data_set* is the data set name and *attribute* is the attribute name as shown in the Threshold Editor. If the data set or attribute name contains a space, replace with an underscore. The *data_set* must be the same data set that you select in the Data set selection field.

The following example shows how you can pass the disk name parameter to your managed resource:

```
/scripts/clean_logs.sh &{KLZ_Disk.Disk_Name}
```

You can pass in one or more attributes from the data set. If specified, multiple attributes are passed into the command in order ($1, $2, and so on).

You must ensure the script or programs executed by the command are installed on the agent system since Cloud APM does not provide a mechanism to distribute scripts or programs. The command runs from the command line with the same user account that the agent was started with. Ensure the user that starts the agent has permission to execute the command. For example, if the agent is running as root, then root runs the command on the managed system.

The following options control how often the command is run:

Select □ **On first event only** if the data set returns multiple rows and you want to run the command for only the first event occurrence in the data sample. Clear the check box to run the command for every row that causes an event.

Select □ **For every consecutive true interval** to run the command every time the threshold evaluates to true. Clear the check box to run the command when the threshold is true, but not again until the threshold evaluates to false, followed by another true evaluation in a subsequent interval.

After you click **Save**, the threshold is applied to all monitored systems of the same data type within the assigned resource groups.

Tip: You can control event behavior and event forwarding through the **Event Manager** options in the **Advanced Configuration** page. See "Advanced Configuration" on page 1050.

Note: To see a list of the attributes that are suitable for inclusion in the threshold definition, create a table with the data set that you plan to use. .

Related concepts

"Background information" on page 958
Review the background information to learn about thresholds, predefined thresholds for your agents, the resource groups that they are assigned to, and customizing thresholds.

Related tasks

"Tutorial: Defining a threshold" on page 964

"Tutorial: Defining a threshold to run a command on the managed resource" on page 966
You can use the **Threshold Editor** to pass certain parameters to agents. You can specify commands or a script of commands to run automatically when an event is triggered.

"Integrating with Netcool/OMNIbus" on page 945

You can forward events from IBM Cloud Application Performance Management into your on-premises IBM Tivoli Netcool/OMNIbus event manager.

Customizing an event to forward to an EIF receiver

You can customize the threshold events that are sent to an Event Integration Facility (EIF) receiver, such as Netcool/OMNIbus ObjectServer, to Cloud Event Management or Alert Notification. Use the **EIF Slot Customization** window to customize the event content that is forwarded to the event destinations, thus overriding the default mapping. You can create map definitions for threshold events that sent to the Event Integration Facility receiver. Use the **EIF Slot Customization** window to customize how events are mapped to forwarded EIF events, thus overriding the default mapping. By customizing the message template, you can add information about the problem that was identified by the event and specific data from the event.By customizing the message template, you can add information about the problem that was identified by the event and include specific data from the event.

About this task

You can customize the EIF base slot, which is a predefined **msg** slot that sends the threshold formula to an event destination. You can also add one or more EIF custom slots to the event. If you are using the Netcool/OMNIbus ObjectServer, you must update the EIF probe rules file and the ObjectServer triggers if you want to see the custom slots in the Netcool/OMNIbus UI.

You can customize the EIF base slot, which is a predefined **msg** slot that sends the threshold formula to the EIF receiver. You can also add one or more EIF custom slots, which requires an update to the EIF receiver and the probe rules file.

Procedure

Complete these steps to customize how events for the current threshold are mapped to forwarded events:

1. If the **Threshold Manager** is not open, click **System Configuration** > **Threshold Manager**.
2. Click the **Data Source Type** list box and select the data type that you want to work with.
3. If this is a new threshold, click **New**; otherwise, select a threshold and click **Edit**.
4. To customize how events for this threshold are mapped to forwarded events, ensure that **EIF Forwarder** is set to `Yes`, click **EIF Slot Customization**, and take one of the following steps:
   - **EIF Base Slots**: To customize the base slot, select the radio button for **msg** and click **Edit**.
   - **EIF Custom Slots**: To add a custom slot, click **Add**; to edit a custom slot, select the radio button for the slot and click **Edit**.

   The **Edit Slot** or **Add Slot** window opens.
5. Complete the fields to customize the slot values:

Field	Description	Restriction
Slot name	The name of the EIF custom slot, which must begin with a character.	The EIF base slot is **`msg`** and cannot be changed.
Slot type	The type of EIF custom slot: **`String Type`** or **`Number Type`**.	The EIF base slot is **`String Type`** and cannot be changed.
Subtype	The value that is assigned to the slot, which corresponds to the slot type: • **`Mapped Attribute`** enables the **Mapped attribute** field for adding the value of the selected attribute at the time the event occurred	A **`Number Type`** slot can use only **`Mapped Attribute`**.

Field	Description	Restriction
	- **Literal Value** enables the **Literal value** field for adding text to the message template - **Literal Value + Mapped Attribute** enables the **Literal value** and **Mapped attribute** fields for adding text and attribute values to the message template, and enables the **Add** button for adding multiple text or attribute values (or both). A space is added after each literal value or attribute. Typical usage for the EIF base slot **msg**, is to specify a **Literal Value + Mapped Attribute** for the message template.	
Add	If you want to send multiple literal values or attribute values in the forwarded message, click **Add** to add another set of **Literal value** and **Mapped attribute** fields. Each time you select **Add**, these fields are added to the panel. To remove a set of **Literal value** and **Mapped attribute** fields, clear both fields before clicking **OK**. See Example.	Enabled only when **Subtype** is **Literal Value + Mapped Attribute**. Maximum of 6 sets of **Literal value** and **Mapped attribute** fields. If you're not able to see the fields you added, use the browser zoom out feature (Ctrl -) to shrink the layout to fit the dialog box.
Literal value	The text to include in the message template. For example, a literal value of `Memory utilization is high at` with the mapped attribute **%Memory Utilization**, is shown in the **Event Manager** user interface as `Memory Utilization is high at 97.3%`. The message template consists of fixed message text and variable substitution references, or symbols. The symbol refers to common or event slot data or a special reference to the threshold formula. Common slots are those slots that are included in all forwarded events, such as *threshold_name*; event slots are those slots that are specific to the threshold `msg`.	Disabled when **Subtype** is **Mapped Attribute**.
Mapped attribute	The attribute whose value you want to add to the message template. The attributes available are from the data set that was selected for the threshold. For example, for a threshold that monitors for high processor time, you might want to map the user time percentage attribute.	Maximum of 6 **Mapped attribute** fields. Disabled when **Subtype** is **Literal Value**. When the **Slot type** is **Number Type**, only numeric attributes are available.
Multiplier	The multiplier is the value that is defined after you customize the original mapped attribute number value by a multiplier: slot value = *attribute1* * *n*. For example, if you want to convert minutes to seconds in the EIF event, you would specify a multiplier of 60. The	Enabled only for numeric attributes (**Slot type** is **Number Type**).

Field	Description	Restriction
	multiplier value can be a fraction, expressed as a decimal, such as 0.5 or 5.4.	

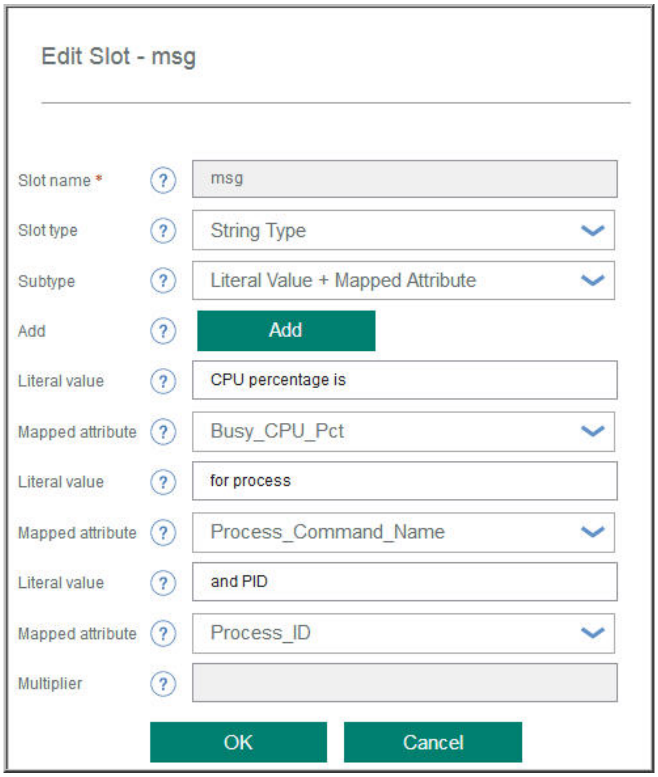
After you click **OK** to close the window, the **EIF Slot Customization** window lists the slot name and whether it is customized.

6. After you are finished editing the EIF base slot or adding, deleting, or editing EIF custom slots for the threshold, click **OK**.
7. After you are finished editing the threshold, click **Save**.

   For more information, see "Threshold Manager" on page 967.

Example

The `Linux_BP_ProcHighCpu_Critical` threshold tests for CPU consumption of 95% or higher. To add the Busy CPU percentage, the process command name, and the process ID to the summary message (contained in the msg slot), the msg slot was customized with three sets of **Literal value** and **Mapped attribute** fields:



Edit Slot - msg

Slot name * msg

Slot type String Type

Subtype Literal Value + Mapped Attribute

Add

Literal value CPU percentage is

Mapped attribute Busy_CPU_Pct

Literal value for process

Mapped attribute Process_Command_Name

Literal value and PID

Mapped attribute Process_ID

Multiplier

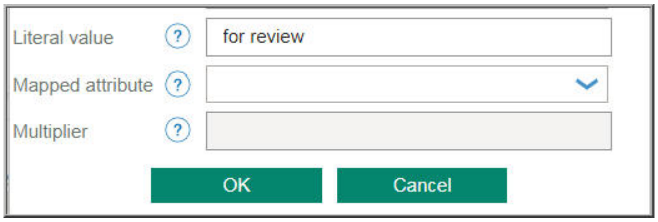
The message template looks like this:

```
CPU percentage is Busy_CPU_Pct for process Process_Command_Name and PID Process_ID
```

And the resulting message viewed in the Event Manager might look like this:

```
CPU percentage is 97 for process large.exe and PID 9876
```

You can also add the **Literal value** and **Mapped attribute** fields and leave one field empty. For example, to append "for review" to the message template, click **Add** and enter `for review` for **Literal value**.



Literal value for review

Mapped attribute

Multiplier

The message template now looks like this:

```
CPU percentage is Busy_CPU_Pct for process Process_Command_Name and PID Process_ID for review
```

And the resulting message viewed in the **Event Manager** might look like this:

```
CPU percentage is 96 for process big.exe and PID 5432 for review
```

What to do next

If you created new EIF custom slots, you must identify the new slots in the `alerts.status` table on your Netcool/OMNIbus ObjectServer, then update the `itm_apm_event.rules` configuration file that was installed during Netcool/OMNIbus integration with Cloud APM.

Adding EIF custom slots to the Netcool/OMNIbus ObjectServer database

When you add new EIF custom slots for thresholds, you must identify them in your EIF receiver before you can view forwarded events that use the custom slots. If you have Netcool/OMNIbus integrated with Cloud APM, update the `alerts.status` table to define the new slots.

About this task

When you configured Netcool/OMNIbus integration with Cloud APM, step "3" on page 949 had you load `itm_apm_db_update.sql`. The following procedure has you use the SQL interactive interface to update the `alerts.status` table in the `itm_apm_db_update.sql` database.

Procedure

Complete these steps on the Netcool/OMNIbus ObjectServer to define the new EIF custom slots that you created in the **Threshold Editor**:

1. Start the SQL interactive interface for editing the database:

   Linux

   ```
   $OMNIHOME/bin/nco_sql -user user_name -password password
   -server server_name > itm_apm_db_update.sql
   ```

   Example:

   ```
   $OMNIHOME/bin/nco_sql -user smadmin -password passw0rd -server NCOMS >
   /tmp/apm/itm_apm_db_update.sql
   ```

   Windows

   ```
   itm_apm_db_update.sql | %OMNIHOME%\..\bin\isql -U user_name
   -P password -S server_name
   ```

   Example:

   ```
   \temp\apm\itm_apm_db_update.sql | %OMNIHOME%\..\bin\isql -U smadmin
   -P passw0rd -S NCOMS
   ```

2. For each EIF custom slot, type the **ALTER TABLE** SQL command with the custom slot name and slot type in the following format, press Enter, then type go and press Enter:

   - For a string slot type,

     ```
     alter table alerts.status add CustomSlotName varchar(512);
     ```

   - For a number slot type,

     ```
     alter table alerts.status add CustomSlotName integer;
     ```

   where *CustomSlotName* is the name of the EIF custom slot exactly as it was entered in the **slotName** field of the **Add Slot** window in the **Threshold Editor**.

Example

The example shows the **`alter table`** commands to add **`BusinessApplication`** and **`GenericMetric`** custom slots.

```
alter table alerts.status add BusinessApplication varchar(512);
```

```
alter table alerts.status add GenericMetric integer;
```

What to do next

Update the `itm_apm_event.rules` configuration file that was installed as part of Netcool/OMNIbus integration with Cloud APM. For more information, see "Adding EIF custom slots to the EIF receiver event rules" on page 976.

Adding EIF custom slots to the EIF receiver event rules

If you defined new EIF custom slots for thresholds, you must update the rules file to identify the new slots to the EIF receiver.

About this task

These steps have you update the `itm_apm_event.rules` file on the Netcool/OMNIbus Probe for Tivoli EIF to identify each new EIF custom slot. If you are using another EIF receiver, update the rules files as required by the receiver.

Procedure

1. On the system where the Probe for Tivoli EIF is installed, change to the installation directory.

   Linux cd *install_dir*/tivoli/netcool/omnibus/probes/linux2x86

   Windows cd *install_dir*\Tivoli\Netcool\omnibus\probes\win32

   where *install_dir* is the default `/opt/IBM/` or `C:\IBM\` or the directory that you specified when the probe was installed.
2. Make a backup copy of the `itm_apm_event.rules` file.
3. Open the Probe for Tivoli EIF `itm_apm_event.rules` file in a text editor.

   The file has three parts to it that you are editing to add the custom EIF custom slot (or slots) that you created.
4. Append the **`if`** statements with a new statement for each EIF custom slot that uses the following format:

   ```
       if(exists($CustomSlotName))
       {
           if(regmatch($CustomSlotName, "^'.*'$"))
           {
               $SourceType = extract($CustomSlotName, "^'(.*)'$")
           }
       }
   ```

   where *CustomSlotName* is the name of the EIF custom slot exactly as it was entered in the **slotName** field of the **Add Slot** window.
5. Append the list of @ entries with a new row for each EIF custom slot that uses the following format:

   ```
   @CustomSlotName=$CustomSlotName
   ```

   where *CustomSlotName* is the name of the EIF custom slot.
6. Append the list of `$tmpEventData` entries with a new row for each EIF custom slot that uses the following format:

   ```
   $tmpEventData = nvp_remove( $tmpEventData, "CustomSlotName")
   ```

where *CustomSlotName* is the name of the EIF custom slot.

7. Save and close the `itm_apm_event.rules` file.
8. Restart the Probe for Tivoli EIF to implement your updates.

Results

The rules file is updated and the Probe for Tivoli EIF can now process threshold events that use the new EIF custom slots and forward the event details to the Netcool/OMNIbus ObjectServer.

Example

Here is an excerpt from the `itm_apm_event.rules` after it was edited to add these EIF custom slots: **`BusinessApplication`** and **`GenericMetric`** (shown in italic).

```
# ----------------------------------------------------------------
    if(exists($SourceID))
    {
        if(regmatch($SourceID, "^'.*'$"))
        {
            $SourceID = extract($SourceID, "^'(.*)'$")
        }
    }
    ...
    ...
    ...
    if(exists($ManagedSystemGroups))
    {
        if(regmatch($ManagedSystemGroups, "^'.*'$"))
        {
            $SourceType = extract($ManagedSystemGroups, "^'(.*)'$")
        }
    }
    if(exists($BusinessApplication))
    {
        if(regmatch($BusinessApplication, "^'.*'$"))
        {
            $SourceType = extract($BusinessApplication, "^'(.*)'$")
        }
    }
         if(exists($GenericMetric))
    {
        if(regmatch($GenericMetric, "^'.*'$"))
        {
            $SourceType = extract($GenericMetric, "^'(.*)'$")
        }
    }

 @SourceID=$SourceID
 @URL=$ManagementURL
 @Service=$Service
 @SourceType=$SourceType
 @SubscriberID=$TenantID
 @APMHostname=$apm_hostname
 @ManagedSystemGroups=$ManagedSystemGroups
 @BusinessApplication=$BusinessApplication
 @GenericMetric=$GenericMetric

 # -----------------------------------------------------------------
 # -  RTC 66157
 # -----------------------------------------------------------------
 if ( exists ( $appl_label ) )
 {
    if ( match($appl_label, "PI:A:S"))
    {
        @Class = 87723
    }
 }
 # -----------------------------------------------------------------
 # -  RTC 48775 - APM FP5 agents do not populate data in email of EMaaS Basic
 # -----------------------------------------------------------------
 if (match( $situation_eventdata, "~" ) )
 {
    # Dump all fields into the ITMEventData field
    $tmpEventData = nvp_add($*)
    # Remove the duplicated fields
    $tmpEventData = nvp_remove( $tmpEventData, "appl_label")
    $tmpEventData = nvp_remove( $tmpEventData, "control")
    ...
    ...
    ...
    $tmpEventData = nvp_remove( $tmpEventData, "ManagedSystemGroups")
    $tmpEventData = nvp_remove( $tmpEventData, "EventSeqNo")
    $tmpEventData = nvp_remove( $tmpEventData, "BusinessApplication")
    $tmpEventData = nvp_remove( $tmpEventData, "GenericMetric")
      @ITMEventData = $tmpEventData
```

Sending email in response to an event

When your managed environment includes IBM Alert Notification, you can provide email notification when application performance exceeds thresholds.

About this task

To configure email notification, you must enable IBM Alert Notification as described in the Alert Notification topic collection on IBM Knowledge Center. Then add monitored applications to resource groups. For every resource group, you can configure one or several email addresses. When the performance of any application in a group exceeds a threshold, you get an email notification to the addresses that are configured for the group.

Procedure

1. Click **System Configuration** > **Resource Group Manager**.
2. Either click **New** to create the resource group for which you want to configure email notification or select an existing group and click **Edit**.

   The **Resource Group Editor** opens.
3. Click **Configure email notification** to open the IBM Alert Notification application in a new browser tab or window. Use Alert Notification to create users and associate their email addresses with resource groups for receiving event notifications by email.

Using the Resource Group Management Service API

Use the Resource Group Management Service API to manage the lifecycle of groups of managed systems from the command line.

About this task

Complete resource group tasks such as creating, viewing, updating, and deleting groups of managed systems. Add and remove individual systems from custom groups. View a list of systems that you added to a specific custom resource group, and view a list of systems that are automatically added to the built-in groups such as the system resource group.

You can create scripts for automating such tasks as defining resource groups and assigning agents to these resource groups. The resource groups can be targets of threshold distributions and or access control policies.

The following operations are described in API Explorerand in the Example at the end of this topic.

- Return all resource groups, agents, or a specific resource group or agent.
- Create a custom resource group or update the definition of an existing group
- Delete a specified custom resource group
- Add agents to a custom resource group
- Remove agents from a custom resource group

Procedure

Complete these steps to define and change custom resource groups with the Resource Group Management Service API. System resource groups and agents cannot be modified.

1. Complete step 1 to step 9 in the Exploring the APIs topic.

   Step 10 and step 11 provide additional details.
2. Click **USE** and select a key, for example, **Key1**.

   Note: Click **Hide** to show your client-Id and client secret. Make a note of them because, if you are making API calls with external tools outside of API Explorer, they are needed. Then, click **Show** to hide them.
3. Populate all required headers, denoted with an asterisk.

   X-IBM-Service-Location

   * header is the geographic location of your subscription such as na for North America

   Authorisation

   * header is your base64-encoded string of the IBM ID and password. When you encode the IBM ID and password in the based64-encoder tool, the format must be *IBMid:password*, for example, `Basic YXBtYWRtaW46YXBtcGFzcw==`.
4. You must include a referer header in all POST, PUT, and DELETE requests. The value for the referer header is always:

   ```
   -H 'Referer: https://api.ibm.com'
   ```
5. Scroll to locate and click **Test**.

Results

The changes that you make to custom resource groups in the API are effective immediately and displayed in the **Resource Group Manager** (see "Resource Group Manager" on page 962).

Example

This command returns the names, unique identifiers, status, hostname, version, and agent type for all agents:

```
GET /1.0/topology/mgmt_artifacts?_filter=entityTypes=Agent&_field=keyIndexName&
_field=online&_field=hostname&_field=version&_field=productCode&_field=description
```

This command returns a list of all Linux OS agents:

```
GET /1.0/topology/mgmt_artifacts?_filter=entityTypes=Agent&_filter=description=
"Linux OS"&_field=keyIndexName
```

This command returns a list of system and custom groups:

```
GET /1.0/topology/mgmt_artifacts?_filter=entityTypes:AgentGroup,
AgentSystemGroup&_field=keyIndexName&_field=displayLabel
```

This command returns the list of agents that are assigned to a group that has the unique identifier {id}:

```
GET /1.0/topology/mgmt_artifacts/{id}/references/to/contains
```

The following example uses the curl command to create a custom group.

```
POST /1.0/topology/mgmt_artifacts
```

Note: The body of the POST request must contain a JSON object that defines the group as shown by the **-d** parameter.

```
curl -X POST \
  https://api.ibm.com/perfmgmt/run/1.0/topology/mgmt_artifacts \
  -H 'Referer: https://api.ibm.com' \
  -H 'authorization: Basic REPLACE_BASE64_ENCODED_STRING' \
  -H 'content-type: application/json' \
  -H 'x-ibm-client-id: REPLACE_KEY_VALUE' \
  -H 'x-ibm-client-secret: REPLACE_KEY_VALUE' \
  -d '{
  "keyIndexName": "customGroup",
```

```
  "description": "Custom group description",
  "displayLabel": "customGroupLabel",
  "entityTypes": [
    "AgentGroup"
  ],
  "arbitraryStringProperty": "Your custom property value"
}'
```

This command adds an agent with unique identifier {otherid} to a custom group that has unique identifier {id}:

```
POST /1.0/topology/mgmt_artifacts/{id}/references/to/contains/{otherid}
```

This command removes an agent with unique identifier {otherid} from a custom group that has unique identifier {id}:

```
DELETE /1.0/topology/mgmt_artifacts/{id}/references/to/contains/{otherid}
```

Using the Threshold Management Service API

Use the Threshold Management Service API to to manage the lifecycle of monitoring thresholds from the command line.

About this task

Complete threshold manager tasks such as creating, viewing, updating, and deleting thresholds. Assign resource groups to these thresholds. View a list of all thresholds and resources assignments. View a list of all thresholds that are assigned to a specific resource group.

You can create scripts for automating such tasks as defining thresholds and assigning these thresholds to resource groups.

The following operations are described in API Explorerand in the Example at the end of this topic.

- Return all thresholds or get a specific threshold. You can filter the request with these attributes: **`label`**, which corresponds to the threshold name; **`_appliesToAgentType`**, which corresponds to the 2-character product code, and **`_uiThresholdType`**, which corresponds to the threshold type that is shown in the Threshold Manager and Resource Group editor pages of the Cloud APM console. You can use **`_offset`** or **`_limit`** when getting thresholds
- Create a threshold or update the definition of an existing threshold. You must include the X-HTTP-Method-Override header and set to PATCH for update request
- Delete a specified threshold
- Return all resource assignments or a specific resource assignment, which shows the thresholds that are assigned to each resource group. You can filter the request with these attributes: **`resource._id`** and **`threshold._id`**; and use these supported operators are **=** (equal) and **`!=`** (not equal)
- Create a resource assignment, which assigns a single threshold to a single resource group
- Delete a resource assignment, which removes a single threshold from a single resource group

Procedure

1. Complete step 1 to step 9 in the Exploring the APIs topic.

   Step 10 and step 11 provide additional details.
2. Click **USE** and select a key, for example, **Key1**.

   Note: Click **Hide** to show your client-Id and client secret. Make a note of them because, if you are making API calls with external tools outside of API Explorer, they are needed. Then, click **Show** to hide them.
3. Populate all required headers, denoted with an asterisk.

   X-IBM-Service-Location

   * header is the geographic location of your server such as na for North America

Authorization

* header is your base64-encoded string of the IBMid and password. When you encode the IBMid and password in the based64-encoder tool, the format must be *IBMid:password*. For example `Basic YXBtYWRtaW46YXBtcGFzcw==`.

4. Scroll to locate and click **Test**.

Example

This command returns all the thresholds that are registered with the server:

```
GET /threshold_types/itm_private_situation/thresholds
```

This command returns the information for the threshold with the label (name) `My_threshold`.

```
GET /threshold_types/itm_private_situation/thresholds?_filter=label%3DMy_threshold
```

This command returns all the thresholds for agent type LZ, which is the component code for the Linux OS agent.

```
GET /threshold_types/itm_private_situation/thresholds?_filter=_appliesToAgentType%3DLZ
```

This command has the same output as the previous command but the agent name as it appears in the Cloud APM console is given.

```
GET /threshold_types/itm_private_situation/thresholds?_filter=_uiThresholdType%3DLinux OS
```

This command returns all the resource groups that threshold 123 is assigned to:

```
GET /resource_assignments?_filter=threshold._id=123
```

The following example uses the curl command to create a new threshold.

```
POST /1.0/thresholdmgmt/threshold_types/itm_private_situation/thresholds
```

Remember: The body of the POST request must contain a JSON object that defines the threshold as shown by the **-d** parameter. Example:

```
curl -X POST\
 https://api.ibm.com/perfmgmt/run/1.0/thresholdmgmt/threshold_types/itm_private_situation/
thresholds  \
  -H 'authorization: Basic REPLACE_BASE64_ENCODED_STRING' \
  -H 'content-type: application/json' \
  -H 'x-ibm-client-id: REPLACE_KEY_VALUE' \
  -H 'x-ibm-client-secret: REPLACE_KEY_VALUE' \
  -d '{
  "label": "Your_Linux_Threshold_Name",
  "description": "Your Linux Threshold Definition",
  "configuration": {
    "type": "json",
    "payload": {
      "formulaElements": [
        {
          "function": "*MKTIME",
          "metricName": "KLZ_CPU.Timestamp",
          "operator": "*EQ",
          "threshold": "1455767100000",
          "timeDelta": {
            "operator": "+",
            "delta": "3",
            "unit": "Hours"
          }
        }
      ],
      "period": "011500",
      "periods": "3",
      "severity": "Fatal",
      "matchBy": "KLZCPU.CPUID",
      "operator": "*OR",
      "actions": [
    {
     "name": "command",
     "command": "ps -ef",
     "commandFrequency": "Y",
     "commandWhen": "Y",
     "commandWhere": "N"
    }
   ]
  }
 }
}'
```

Managing user access

Use the Role Based Access Control features in Cloud APM to grant users the access privileges they require for their role.

Security in Cloud APM is based on roles. A role is a group of permissions that control the actions you can perform in Cloud APM. You can create customized roles in Cloud APM. You can assign permissions to customized roles, or you can assign more permissions to existing default roles. You can assign users and user groups to existing default roles or to customized roles. You can assign users and user groups to multiple roles. Permissions are cumulative, a user or user group is assigned all the permissions for all the roles they are assigned to.

User authentication in Cloud APM is managed through IBM Marketplace, or can be managed through Collaborative Operations if you have a subscription.

User authentication in Performance Management requires an IBMid. Create an IBMid by selecting the **Create an IBMid** link on the **Sign in to IBM** page. To access the **Sign in to IBM** page, go to the **Products and Services** (http://ibm.biz/my-prodsvcs) page on IBM Marketplace and you will be directed to **Sign in to IBM** page (you might need to log out and return to the **Products and Services** (http://ibm.biz/my-prodsvcs) page).

Once you have an IBMid you can log in to Cloud APM either directly or from Products and Services. Authorization to access an instance of Cloud APM is managed through **Products and Services** or through Collaborative Operations if you have a subscription. By default, the user who requests a trial or purchases a subscription has administrator privileges.

The owner of the Cloud APM subscription is the default user in Cloud APM. This is the user who requests a trial or purchases a subscription. This default user is a member of the Role Administrator role and has administrator privileges that allow this user to add new users. Subsequent users are by default added to the Monitoring User role.

If you are not a member of a role and you attempt to log in to Cloud APM, you receive a **Not Authorized** message.

To add a Cloud APM user:

1. Go to **Products and Services** on IBM Marketplace and expand the **IBM Performance Management** widget.
2. Click **Manage Authorizations** and enter an IBM id or email address to the **Add new user** or **Search existing users** fields. Click **Add user** to add the user.

There is no limit to the number of users you can add to a Cloud APM subscription.

Note: Support is not offered for user groups in Cloud APM.

For more information on roles, see "Roles and permissions" on page 982.

Roles and permissions

A *role* is a group of permissions that control the actions you can perform in Cloud APM. Use the Role Based Access Control page to manage users and roles or alternatively use the Authorization API to complete role-based access control tasks from the command line.For more information, see "Exploring the APIs" on page 1049.

Cloud APM has four default roles:

Role Administrator

This role is intended for users whose primary job function is to create access control policies for Cloud APM. This role has all permissions. If you change the default user, the new default user is automatically a member of the Role Administrator role. This role cannot be edited. Role Administrators are prevented from removing themselves from the Role Administrator role. This restriction removes the risk of accidentally removing all users from the Role Administrator role.

Monitoring Administrator

This role is intended for users whose primary job function is to use Cloud APM to monitor systems. Monitoring Administrators perform tasks such as adding monitoring applications, creating thresholds, adding groups of resources, and distributing the thresholds to these resource groups. This role can be edited.

System Administrator

This role is intended for users whose primary job function is to perform administration tasks for the Cloud APM system. System Administrators perform tasks such as configuring the Event Manager, or configuring the Hybrid Gateway. This role can be edited.

Monitoring User

This role is intended for users whose primary job function is to configure and maintain the health and state of systems that are monitored by Cloud APM. This role can be edited.

The following table describes the permissions that you can assign to roles, and the four available default roles and associated permissions:

Table 239. Roles and permissions

	Role Administrator		**Monitoring Administrator**		**System Administrator**		**Monitoring User**	
	View	**Modify**	**View**	**Modify**	**View**	**Modify**	**View**	**Modify**
System configuration permissions								
Advanced Configuration	✓	N/A	—	N/A	✓	N/A	—	N/A
Agent Configuration	✓	N/A	✓	N/A	—	N/A	—	N/A
Informational Pages	✓	N/A	✓	N/A	✓	N/A	✓	N/A
Search Provider	✓	N/A	✓	N/A	—	N/A	—	N/A
Usage Statistics	✓	N/A	✓	N/A	—	N/A	—	N/A
Resource permissions								
Application Performance Dashboard	✓	✓	✓	✓	✓	—	✓	—
Applications	✓	✓	✓	✓	—	—	✓	—
Individual Application	"Application and resource group permissions" on page 987							
Diagnostics Dashboard	✓	N/A	—	N/A	—	N/A	—	N/A
Resource Group Manager	✓	N/A	✓	N/A	—	N/A	—	N/A
Indivdual Resource group	"Application and resource group permissions" on page 987							
Resource Groups	✓	✓	✓	✓	—	—	—	—
Synthetic Script Manager	✓	N/A	—	N/A	—	N/A	—	N/A
Threshold Manager	✓	N/A	✓	N/A	—	N/A	—	N/A

Where

✓ indicates that members of this role have this permission

— indicates that members of this role do not have this permission

N/A indicates that this permission does not exist

Note: Although **`Usage Statistics`** is displayed in the list of **System configuration permissions**, it is no longer applicable to Cloud APM.

The following table describes the actions that are associated with each permission:

Table 240. Permissions

Permission	Description
Advanced Configuration	If you have view permission, you can perform the following tasks: • View **System Configuration** > **Advanced Configuration** in the menu bar. • Make and save changes in the **Advanced Configuration** window. • View **System Configuration** > **Hybrid Gateway Manager** in the menu bar. • Make and save changes in the **Hybrid Gateway Manager** window.
Agent Configuration	If you have view permission, you can perform the following tasks: • View **System Configuration** > **Agent Configuration** in the menu bar. • Make and save changes in the **Agent Configuration** window.
Informational Pages	If you have view permission, you can perform the following task: • View **Getting Started** and **Help** in the menu bar. **Note:** When the **Getting Started** page opens, if you clear **Show this page at startup**, for subsequent logins, you see a permission denied error. However, you are still able to navigate to the **Getting Started** page and any other areas that you have permission to.
Search Provider	If you have view permission, you can perform the following tasks: • View **System Configuration** > **Configure Search Providers** in the menu bar. • Make and save changes in the **Configure Search Providers** page.

Table 240. Permissions (continued)

Permission	Description
Application Performance Dashboard	If you have view permission, you can perform the following tasks: • View **Performance** > **Application Performance Dashboard** in the menu bar. • View the Application Performance Dashboard and the **My Components** and **My Transactions** predefined applications. **Note:** To determine what permissions are required to see systems in the My Components application, see "Application and resource group permissions" on page 987. **Note:** The My Transactions application is displayed only if you are using Web Site Monitoring. All Web Site Monitoring synthetic transactions are displayed in the My Transactions application. • Open custom dashboard pages in the Custom Views tab. • Create views in the Attribute Details tab and save them for your own use. If you have modify permission, you can perform the following tasks: • View **Performance** > **Application Performance Dashboard** in the menu bar. • View the Application Performance Dashboard and the **My Components** and **My Transactions** predefined applications. **Note:** To determine what permissions are required to see systems in the My Components application, see "Application and resource group permissions" on page 987. **Note:** The My Transactions application is displayed only if you are using Web Site Monitoring. All Web Site Monitoring synthetic transactions are displayed in the My Transactions application. • Create and save custom dashboard pages in the Custom Views tab. • Create views in the Attribute Details tab and share them with others. • View the **Actions**>**Edit** option in component pages, this option enables you to edit the threshold values and other settings of the group widgets that display in the Components dashboard.
Applications	If you have view permission, you can perform the following tasks: • View applications in the Application Dashboard. If you have modify permission, you can perform the following tasks: • View applications in the Application Dashboard • Create, modify, and delete applications with the ⊕ ⊖ ✎ tools in the Application Dashboard.
Individual Application	See "Application and resource group permissions" on page 987.
Resource Group Manager	If you have view permission, you can perform the following task: • View **System Configuration** > **Resource Group Manager** in the menu bar.

Table 240. Permissions (continued)	
Permission	**Description**
Resource Groups	If you have view permission, you can perform the following tasks: • View resource groups and the systems in them in the Resource Group Manager if you also have the Resource Group Manager view permission. • View the systems in the My Components predefined application if you also have the Application Performance Management view permission or modify permission • View the systems in the **Add Application** window if you also have permission to modify applications. If you have modify permission, you can perform the following tasks: • View resource groups and their contents in the Resource Group Manager if you also have Resource Group Manager view permission. • View the systems in the My Components predefined application if you also have Application Performance Management view or modify permission. • View the systems in the **Add Application** window if you also have permission to modify applications. • Create, modify, and delete resource groups in the Resource Group Manager if you also have Resource Group Manager view permission. To assign thresholds to a resource group, you also need to be a member of a role that has view permission for Threshold Manager. **Note:** The Resource Group Manager is used to organize monitored systems into groups, so that thresholds can be assigned to these groups. If you do not have view permission to the Threshold Manager, you are not able to see the thresholds that are assigned to Resource Groups. If you assign the modify Resource Groups permission to a role, you also need to assign the view Threshold Manager permission to the role.
Individual Resource Group	See "Application and resource group permissions" on page 987.
Threshold Manager	If you have view permission, you can perform the following tasks: • View **System Configuration** > **Threshold Manager** in the menu bar. • Create, modify, and delete thresholds in the Threshold Manager. • View and edit resource group assignment for thresholds in the Threshold Manager if you have appropriate permissions for the resource group (or groups). • Alternatively, view and edit thresholds assignment for resource groups in the Resource Group Manager if you have appropriate permissions for the Resource Group Manager and resource group (or groups), and view permission for the Threshold Manager.
Synthetic Script Manager	If you have view permission, you can perform the following tasks: • Create, modify, and delete synthetic transactions in the Synthetic Transaction Manager. **Note:** To work with synthetic transactions in the Synthetic Transaction Manager, you also need to be a member of a role that has view permission for **Agent Configuration**.

Table 240. Permissions (continued)

Permission	Description
Diagnostics Dashboard	If you have view permission, the **Diagnose** button is enabled on the diagnostic dashboards for the WebSphere Applications agent, Node.js agent, Ruby agent, and Microsoft .NET agent. Click the **Diagnose** button to drill-down to diagnostics dashboards.

Application and resource group permissions
Permissions can be assigned to individual applications and resource groups.

Application permissions

In Cloud APM, an application is a group of components and the instances within those components. Use the **Add Application** window to define an application. For more information on how to define an application, see Managing Applications.

To select **Performance** > **Application Performance Dashboard** in the Cloud APM console, you must be assigned the view or modify permission for the Application Performance Dashboard. This permission also allows you to see the **My Components** and **My Transactions** predefined applications. The **My Transactions** application is displayed only if you are using Web Site Monitoring. To see other custom applications, you must have view permission or modify permission for either all applications or for an individual application.

Note: If an application is renamed, permissions are not retained; you must reassign view and modify permissions.

View

View permission for an application is dominant over any other permissions. To view an application, you do not need to be a member of a role that has view permission for each component and component instance within the application. The following table describes the actions that you can perform if you have view permission for an application:

Table 241. View permission for an application

Action	Permission available
View all the supporting components within that application.	✓
View the application and its components in the navigation tree.	✓
View the application components in My Components.	✓
View customized dashboard pages that are associated with the application.	✓
Add or remove components from the application.	—
Assign thresholds to the components of the applications.	—
View the supporting components of an application in the Resource Group Manager.	—

Modify
If you are a member of a role that has modify permission for an individual application, you can

- Delete the application.
- Create customized dashboard pages in the Custom Views tab. See "Custom views" on page 1086.

- Add or remove components and component instances by using the **Edit Application** window. The components and component instances that are available to you in the **Edit Application** window are filtered based on your role permissions. The following components will be available:
  - Components that you directly have permission to access in system resource groups and custom resource group
  - Components that you indirectly inherited permissions from, based on other applications that you have modify permission to

Resource Group permissions

Use Resource groups to group components together by their type or purpose. For more information on how to create resource groups, see "Resource Group Manager" on page 962.

To select **System Configuration** > **Resource Group Manager**, you must be assigned the Resource Group Manager view permission. To view resource groups in the **Resource Group Manager** or to view resource group members in the **My Components** application, you also must be assigned view or modify permission for all resource groups or for individual resource groups.

There are two different types of resource groups: custom resource groups and system resource groups.

Custom defined resource groups

Create custom resource groups in the Resource Group Manager. Use custom resource groups to group resources together based on their purpose.

The following table describes the actions that you can take if you have the view permission for a custom resource group:

Table 242. View permission for a custom resource group

Action	Permission available
View the custom resource group and the resources in it in Resource Group Manager.	✓
View resources that are part of the custom resource group in the **Add Application** window if you also have modify permission for applications.	✓
View resources that are part of the custom resource group in the **My Components** predefined application if you also have one of the Application Performance Dashboard permissions.	✓
Add resources to the custom resource group.	–
Delete resources from the custom resource group.	–

The following table describes the actions that you can take if you have the modify permission for a custom resource group:

Table 243. Modify permission for a custom resource group

Action	Permission available
Assign thresholds to the custom resource group in Threshold Editor. **Note:** To assign thresholds, you also need to be a member of a role that has view permission for Threshold Editor.	✓

Table 243. Modify permission for a custom resource group (continued)

Action	Permission available
Add resources to the custom resource group.	✓
Delete resources from the custom resource group.	✓

System resource groups

System resource groups are automatically defined as part of your Cloud APM environment setup. System resource groups cannot be created manually, deleted, or customized. Only the view permission is available for system resource groups, the modify permission is not available.

System resource groups are defined for each resource type at the time that the resource becomes known to the Cloud APM server. A system resource group exists for each resource type that is connected to the Cloud APM server.

Cloud APM agents are an example of a resource. For example, the first time you download, install, and start a Db2 agent, a system resource group called Db2 is created. This group contains all Db2 agents that are subsequently added to the Performance Management environment.

The system resource group for each resource type contains all the resources of that type including IBM Tivoli Monitoring resources. If your environment has both IBM Tivoli Monitoring and IBM Cloud Application Performance Management, you can install the IBM Cloud Application Performance Management Hybrid Gateway to provide a view of agents from both domains. System defined resource groups contain agents from both domains. For more information, see "Integrating with IBM Tivoli Monitoring V6.3 " on page 929.

Some system resource groups are based on subnode agents. While you can assign thresholds to system resource groups that are based on subnode agents, events are not displayed in the Application Performance Dashboard. Thresholds are assigned to system resource groups based on subnode agents for event forwarding. System resource groups based on subnode agents have the following description in the Resource Group Manager: 'members of this group cannot be added to an application and do not have events displayed in the Performance Management console'. For more information, see "Resource Group Manager" on page 962.

The following table describes the actions you can take if you have view permission for a system resource group:

Table 244. View permission for a system resource group

Action	Permission available
View the system resource group in Resource Group Manager.	✓
View resources that are part of the system resource group in the **Add Application** window if you also have modify permission for applications.	✓
View resources that are part of the system resource group in the **My Components** predefined application if you also have one of the Application Performance Dashboard permissions.	✓
Assign thresholds to the system resource group in Threshold Editor.	✓
Add resources to the system resource group.	—

Table 244. View permission for a system resource group (continued)

Action	Permission available
Delete resources from the system resource group.	–

Working with roles, users, and permissions
Use the Role Based Access Control page to work with roles, users, and permissions.

Before you begin

Alternatively use the Authorization API to complete role-based access control tasks from the command line. For more information, see "Exploring the APIs" on page 1049.

Note: User groups are not supported in Cloud APM.

Procedure

- To filter the list of roles, users, or user groups showing in the Role Based Access Control page, complete the following steps:
  a) Select **System Configuration> Role Based Access Control**.
  b) Click inside the × **Filter** text box and type the partial or full text to filter by.
     As you type, any rows that do not contain what you typed in the filter box are removed from the table.
  c) To remove the quick filter, delete the value or click the "x".
  d) To apply the filter, click .
- To create a new customized role, complete the following steps:
  a) Select **System Configuration> Role Based Access Control**.
  b) In the **Roles** tab, click ⊕. The **Role Editor** page is displayed.
  c) In the **Assign Users to Roles** tab, select the **User Groups** tab, or the **Individual Users** tab and select the users and user groups you want to add to the role.
  d) In the **Assign Permissions to Roles** tab, select the **System Configuration Permissions** or the **Resource Permissions** tab, and select the permissions that you want to assign to the role.
  e) Click **Save**.
- To edit an existing default or customized role, complete the following steps:
  a) Select **System Configuration> Role Based Access Control**.
  b) In the **Roles** tab, click ✎. The **Role Editor** page is displayed.
  c) In the **Assign Users to Roles** tab, click the **User Groups** tab, or the **Individual Users** tab and select the users or user groups you want to add to the role.
  d) In the **Assign Permissions to Roles** tab, select the **System Configuration Permissions** or the **Resource Permissions** tab, and select the permissions that you want to assign to the role.
  e) Click **Save**.
- To delete a role, complete the following steps:
  a) Select **System Configuration> Role Based Access Control**.
  b) In the **Roles** tab, select the role that you want to delete and click −. A confirmation message is displayed, click **OK**.

  Note: When you delete a role, the users that are members of that role are not deleted. They are still available in the **Individual Users** tab, and can be assigned to another role by a Role Administrator.
- To view the permissions for a user group, complete the following steps:

a) Select **System Configuration> Role Based Access Control.**

b) Select the **User Groups** tab. The list of roles that are assigned to each user group is displayed.

Note: All users in the user group are assigned the roles for the user group.

- To view the permissions for an individual user, complete the following steps:

  a) Select **System Configuration> Role Based Access Control**.

  b) Select the **Individual Users** tab. The list of roles that are assigned to each user is displayed.

  Note: Only the roles that are assigned directly to a user are displayed. If a user is a member of a user group and roles are assigned to the user group, then the user group's roles are not displayed in the list of roles for the individual user on the **Individual Users** tab. However, when the user logs in to the Cloud APM console, they have the permissions that are assigned directly to them and the permissions of any user groups that they are a member of.

- To edit the permissions for an individual user or user group, complete the following steps:

  a) Select **System Configuration> Role Based Access Control**.

  b) In the **Individual User** or **User Groups** tab, select the user or user group you want to edit, and click . The **Individual User Editor** page or **User Group Editor** page is opened.

  c) Select the role or roles that you want to assign to the user or user group.

  d) Click **Save**.

- To create a csv file that summarizes the permissions for a user or user group, complete the following steps:

  a) In the **Individual User** or **User Groups** tab, select the required user or user group, and click . The **Individual User Editor** or **User Group Editor** page is opened.

  b) Click **Export Summary**.

  c) Select **Save File**, click **OK**.

  A csv file that summarizes the permission for the user or user group is saved to the specified location.

Results

Role and permission assignment takes effect immediately when you click **Save**.

Accessing and using the Role-Based Access Control Service API

Use the Role-Based Access Control Service API to manage the lifecycle of role-based access control policies from the command line.

About this task

Complete role-based access tasks such as creating, viewing, updating, and deleting roles. Add and delete a set of users or user groups from a specific role. Grant permissions to a specific role. View a list of roles, users, user groups, and permissions that are defined in the system.

You can create scripts for automating such tasks as defining new roles and assigning users, user groups, and permissions to these roles.

Procedure

1. Complete step 1 to step 9 in the Exploring the APIs topic.

   Step 10 and step 11 provide additional details.
2. Click **USE** and select a key, for example, **Key1**.

   Note: Click **Hide** to show your client-Id and client secret. Make a note of them because, if you are making API calls with external tools outside of API Explorer, they are needed. Then, click **Show** to hide them.
3. Populate all required headers, denoted with an asterisk.

X-IBM-Service-Location

* header is the geographic location of your subscription such as na for North America

Authorisation

* header is your base64-encoded string of the IBM ID and password. When you encode the IBM ID and password in the based64-encoder tool, the format must be *IBMid:password*, for example, `Basic YXBtYWRtaW46YXBtcGFzcw==`.

4. You must include a referer header in all POST, PUT, and DELETE requests. The value for the referer header is always:

   ```
   -H 'Referer: https://api.ibm.com'
   ```

5. Scroll to locate and click **Test**.

Example

The following example uses the curl command to create a new role.

```
POST /1.0/authzn/roles
```

Note: The body of the POST request must contain a JSON object that defines the role as shown by the **-d** parameter.

```

curl -X POST \
  https://api.ibm.com/perfmgmt/run/1.0/authzn/roles  \
  -H 'Referer: https://api.ibm.com' \
  -H 'authorization: Basic REPLACE_BASE64_ENCODED_STRING' \
  -H 'content-type: application/json' \
  -H 'x-ibm-client-id: REPLACE_KEY_VALUE' \
  -H 'x-ibm-client-secret: REPLACE_KEY_VALUE' \
  -d '{
    "description": "Your Role Description",
    "id": "/authzn/roles/Your_Role_Id",
    "label": "Your Role Name"
}'
```

Administering your agents

Your IBM Cloud Application Performance Management installation has tools for managing your monitoring agents.

Some of these tools are also used during initial configuration of your managed systems: "Using agent commands" on page 177, "Agent Configuration page" on page 182, and "Using the IBM Cloud Application Performance Management window on Windows systems" on page 182.

Starting agents as a non-root user

If you want to start agents as different users, create a common group on the system and make each user a member of this group.

Before you begin

If you installed and configured your agent as the same non-root user and you want to start the agent as the same user, no special action is required. If you installed and configured your agent as a selected user and want to start the agent as a different user, create a common group on the system. Make all agent management users members of this common group. Transfer ownership of all agent files and directories to this group.

About this task

An autostart script is generated by an installation, upgrade, or configuration. This script (named `ITMAgentsN` or `rc.itmN`, depending on the UNIX operating system) contains an entry for each application in a particular installation. By default all agents are started with root user access. To update system startup scripts and start agents as a non-root user, you must edit the `install_dir/config/`

`kcirunas.cfg` file, which contains a superset of the XML syntax. Each **`productCode`** section in the `kcirunas.cfg` file is disabled by default. Activate a **`productCode`** section for your agent by removing the comment indicator from **`!productCode`**. Commented or deactivated sections are ignored. Uncommented or activated sections for applications that are not installed are ignored.

Procedure

1. Install your monitoring agents on Linux or UNIX as described in "Installing agents" on page 125 on AIX systems or "Installing agents" on page 135 on Linux systems.
2. Optional: Configure your monitoring agents on Linux or UNIX as necessary, see Chapter 7, "Configuring your environment," on page 159.
3. Run the `./secure.sh` script with the group name of the non-root user to secure the files and set the file group ownership to the files.
   For example: **`./secure.sh -g db2iadm1`**
4. To update the system startup scripts, complete the following steps:
   a) Update the `install_dir/config/kcirunas.cfg` file. Activate **`productCode`** sections for your agents. For agents that do not require an instance value, specify the product_code, instance, and user, where the *product_code* value is the two-letter code that is specified in Table 11 on page 178. For agents that do require an instance value, such as the Db2 monitoring agent (product code: ud), specify the product_code, instance, user, and name.
   For example:

   ```
   <productCode>ud</productCode>
   <instance>
   <name>db2inst1</name>
   <user>db2inst1</user>
   </instance>
   <instance>
   <name>db2inst2</name>
   <user>root</user>
   </instance>
   ```

   b) Run the following script with root user or sudo user access: `*install_dir*/bin/UpdateAutoRun.sh`

What to do next

For more information about the **`./secure.sh`** script, see Securing the agent installation files.

Use the same user ID for agent installation and upgrades.

Event thresholds for Transaction Monitoring

You can use event thresholds to immediately monitor your environment. You can also create customized event thresholds that test for certain conditions and raise an event when key performance indicators exceed the threshold.

Response Time Monitoring events

Response Time events are created when web transactions exceed a **Response Time** threshold.

After you click **System Configuration** > **Threshold Manager**, select **Response Time** as the **Data Source Type**. All event thresholds for the Response Time Monitoring environment are applied to all managed systems of the same type.

The following predefined thresholds are available for the Response Time Monitoring Agent.

Table 245. Response Time Monitoring thresholds

Threshold	Description	Formula
`Response_Time_Availability_Crit`	A high percentage of the web transactions failed.	If WRT Transaction Status.Percent_Failed is greater than 10 and WRT Transaction Status.Transaction_Definition_Name is not equal to 'Ignore Resources' then Response_Time_Availability_Crit is true
`Response_Time_Availability_Warn`	A moderate percentage of the web transactions failed.	If WRT Transaction Status.Percent_Failed is greater than 0 and WRT Transaction Status.Percent_Failed is less than 10 and WRT Transaction Status.Transaction_Definition_Name is not equal to 'Ignore Resources' then Response_Time_Availability_Warn is true
`Response_Time_Critical`	The percentage of the web transactions with a slow response time is high.	If WRT Transaction Status.Percent_Slow is greater than 5 and WRT Transaction Status.Percent_Available is equal to 100 and WRT Transaction Status.Transaction_Definition_Name is not equal to 'Ignore Resources' then Response_Time_Critical is true
`Response_Time_Warning`	The percentage of the web transactions with a slow response time is moderate.	If WRT Transaction Status.Percent_Slow is greater than 1 and WRT Transaction Status.Percent_Slow is less than 5 and WRT Transaction Status.Percent_Available is equal to 100 and WRT Transaction Status.Transaction_Definition_Name is not equal to 'Ignore Resources' then Response_Time_Warning is true

Good requests have a response time less than 10 seconds. *Slow requests* have a response time greater than 10 seconds. The 10 second value used to determine good vs slow response time is not configurable.

Transaction Tracking events

Transaction Tracking events are created when middleware transactions exceed a Transaction Tracking threshold.

To view the default Transaction Tracking thresholds, click **System Configuration** > **Threshold Manager**, and select **Transaction Tracking** as the **Data Source Type**.

Tip: You can create your own Transaction Tracking thresholds if required.

The following predefined thresholds are available for middleware transactions.

Table 246. Transaction Tracking thresholds

Threshold	Description	Formula
`Interaction_Avail_Critical`	A high percentage of the middleware interactions failed.	If KTE INTERACTION AGGREGATE DATA.PERCENTAGE_FAILED is greater than 10, then Interaction_Avail_Critical is true
`Interaction_Avail_Warning`	A moderate percentage of the middleware interactions failed.	If KTE INTERACTION AGGREGATE DATA.PERCENTAGE_FAILED is greater than 0 and KTE INTERACTION AGGREGATE DATA.PERCENTAGE_FAILED is less than or equal to 10, then `Interaction_Avail_Warning` is true
`Interaction_Time_Critical`	The percentage of the middleware interactions with a slow total time is high.	If KTE INTERACTION AGGREGATE DATA.PERCENTAGE_SLOW is greater than or equal to 5 and KTE INTERACTION AGGREGATE DATA.PERCENTAGE_FAILED is equal to 0, then Interaction_Time_Critical is true
`Interaction_Time_Warning`	The percentage of the middleware interactions with a slow total time is moderate.	If KTE INTERACTION AGGREGATE DATA.PERCENTAGE_SLOW is greater than 1 and KTE INTERACTION AGGREGATE DATA.PERCENTAGE_SLOW is less than 5 and KTE INTERACTION AGGREGATE DATA.PERCENTAGE_FAILED is equal to 0 then Interaction_Time_Warning is true
`Transaction_Avail_Critical`	A high percentage of the middleware transactions failed.	If KTE TRANSACTION AGGREGATE DATA.PERCENTAGE_FAILED is greater than 10, then Transaction_Avail_Critical is true
`Transaction_Avail_Warning`	A moderate percentage of the middleware transactions failed.	If KTE TRANSACTION AGGREGATE DATA.PERCENTAGE_FAILED is greater than 0 and KTE TRANSACTION AGGREGATE DATA.PERCENTAGE_FAILED is less than or equal to 10, then `Transaction_Avail_Warning` is true
`Transaction_Time_Critical`	The percentage of the middleware transactions with a slow total time is high.	If KTE TRANSACTION AGGREGATE DATA.PERCENTAGE_SLOW is greater than or equal to 5 and KTE TRANSACTION AGGREGATE DATA.PERCENTAGE_FAILED is equal to 0, then Transaction_Time_Critical is true

Table 246. Transaction Tracking thresholds (continued)

Threshold	Description	Formula
`Transaction_Time_Warning`	The percentage of the middleware transactions with a slow total time is moderate.	If KTE TRANSACTION AGGREGATE DATA.PERCENTAGE_SLOW is greater than 1 and KTE TRANSACTION AGGREGATE DATA.PERCENTAGE_SLOW is less than 5 and KTE TRANSACTION AGGREGATE DATA.PERCENTAGE_FAILED is equal to 0 then Transaction_Time_Warning is true

Good requests have a response time less than 10 seconds. *Slow requests* have a response time greater than 10 seconds. The 10 second value used to determine good vs slow response time is not configurable.

Creating thresholds to generate events for transaction monitoring

Use the Threshold Manager to create thresholds for transactions. Thresholds are used to compare the sampled value of an attribute with the value set in the threshold. If the sampled value satisfies the comparison, a transaction event is generated.

About this task

You can monitor when applications report specific conditions using thresholds. For more information about the default thresholds for Transaction Monitoring, see "Event thresholds for Transaction Monitoring " on page 993.

You can create extra thresholds to monitor other aspects of a transaction. For example, you can create a threshold to monitor if the middleware transaction event rate falls. Then, if the transaction event rate falls below that specified by your threshold, an event is generated.

Procedure

To create a threshold and associate it with one or more transactions, complete the following tasks:

1. On the Navigation Bar, click **System Configuration** > **Threshold Manager**. Set the **Data Source type** as **Transaction Tracking**.
2. Click Add to create a new threshold.
3. Set a severity for the event that exceeds this threshold.
4. To associate the threshold with a transaction, set the following values:
   - Data set - `KTE TRANSACTION AGGREGATE DATA`
   - Display item - `Resource_Value`
   - Logical operator - And `(&)`
5. Alternatively, to associate the threshold with an interaction, set the following values:
   - Data set - `KTE INTERACTION AGGREGATE DATA`
   - Display item - `Source_Resource_Value`
   - Logical operator - And `(&)`
6. Click **Add** to add a condition. In the **Add Condition** box, select an attribute and an operator, then enter a threshold value.

   For example, to add a threshold condition that generates a transaction event when the number of transactions per minute falls below 100, set the following values and click **OK**:

   - Attribute - `Transaction_Rate`

- Operator - `Less than`
- Value - `100`

Repeat this step to add more conditions to your threshold if required.

7. In the Group assignment section, select `Transaction Tracking` to assign your threshold to that resource group.
8. Click **Save**.

Results

You created a threshold and associated it with a transaction or interaction. When the threshold conditions are met, an event is generated. You can monitor events in the Events tab of the Application Performance Dashboard.

Example

To create thresholds for the Response Time Monitoring agent to monitor other aspects of a web transaction in addition to the defaults:

1. In the Threshold Manager, set the **Data Source type** as **Response Time**.
2. When you add the threshold, use the following settings:
   - Data set - `WRT Transaction Status`
   - Display item - `Application`
   - Logical operator - `And (&)`
   - Group assignment - `Web Response Time`

Managing OS agent events

You can configure the OS agent to manage events.

Event filtering and summarization

Use the event filtering and summarization options that you set in the configuration (`.conf`) file to control how duplicate events are handled by the OS agent.

When a log is monitored, an event can display multiple times quickly. For example, this repeated logging can occur when the application that produces the log encounters an error and it logs this error continuously until the threshold is resolved. When this type of logging occurs, an excessive number of events are sent to the Performance Management infrastructure. The volume of events has a negative impact on performance.

Note: The event detection and summarization procedures are supported only on events that are sent to Performance Management. You cannot complete these procedures on events that are sent to OMNIbus by EIF.

Detecting and filtering duplicate events

You can configure the OS agent to handle duplicate events.

To alleviate the problem of multiple duplicate events, you define what constitutes a duplicate event by using the `DupDetectionKeyAttributes` tag in the `.conf` file. In a comma-separated list, you include one or more defined Performance Management attributes that you want to use to determine whether an event is considered a duplicate. In the following example, events with the same message and the same `CustomSlot1` are to be considered duplicates:

```
DupDetectionKeyAttributes=msg,CustomSlot1
```

Duplicate events are detected from Performance Management attributes. Therefore, if you want detection of duplicates to be based on particular slots that you defined, complete the following steps:

1. Map the slot value to a Performance Management attribute.

2. Map that Performance Management attribute to the `DupDetectionKeyAttributes` tag in the `.conf` file.

Using the following example, where the important slots, `eventclass` and `eventid`, are mapped to *CustomSlot1* and *CustomSlot2*:

```
REGEX BaseAuditEvent
^([A-Z][a-z]{2} [0-9]{1,2} [0-9]{1,2}:[0-9]{2}:[0-9]{2}
 [0-9] {4}) [0-9] (\S+) (\S+) \
Microsoft-Windows-Security-Auditing (\S+) ([0-9]+) (.*)
timestamp $1
severity $2
eventclass $3 CustomSlot1
eventkeywords $4
eventid $5 CustomSlot2
msg $6
END
```

if you want to specify certain events as duplicate events, in the `.conf` file, map the Performance Management attributes to the `DupDetectionKeyAttributes` tag as shown here:

```
DupDetectionKeyAttributes=CustomSlot1,CustomSlot2
```

Note:

1. The `CustomSlot` attribute names are case-sensitive and so you must enter the names exactly as shown in the preceding example.
2. If you do not provide a list of attributes, the values are defaulted to `Class` and `Logname`.

The events where these attributes match are considered to be duplicate events by the agent.

Since the duplicate detection is global, it is good practice to pick a set of custom slots to use as keys and use them this way in all format statements. For example, use slots 1 - 3 for keys. If a format needs only one key but also needs more slots, use slot one to contain the name value and slots four to n to contain the other data.

Summary interval

The duplication detection procedure operates over a time period that is known as the Summary Interval.

Duplicate events are counted during this interval and then reset when the interval expires. The counter starts the count again beginning at 0 at the start of each new summary interval.

The agent sends a summary event for each event set that it monitors during the interval. The summary event contains the attribute values of the first event that matched. The summary event also contains a count that indicates how many duplicates of that event occurred during the summary interval.

The summary interval is set in the configuration (`.conf`) file as shown in the following example:

```
EventSummaryInterval=300
```

The value that is assigned to the summary interval is in seconds, so in this example, the summary interval is 5 minutes.

Filtering events

If event filtering is running, the EventFloodThreshold setting in the `(.conf)` file informs the agent when to send an event.

The following table shows the EventFloodThreshold values.

Table 247. EventFloodThreshold values

EventFloodThreshold values	**Description**
send_all	The *send_all* value is the default value. All events are sent even if these events are duplicate events.

Table 247. EventFloodThreshold values (continued)

EventFloodThreshold values	**Description**
send_none	The *send_none* value means that no individual events are sent. Only the summary events are sent.
send_first	Use the *send_first* value to send the first event as soon as it is encountered. If duplicates of that first event occur within a specified time, then subsequent duplicates of this first event are not sent. For more information, see "Summary interval" on page 998.
n integer	Use the *n* integer value to send only every *n* occurrence of an event (for example every fifth duplicate) during a specific time. For more information, see "Summary interval" on page 998.

Summarization attributes
The Event Type and Occurrence Count attributes are used to help summarize events.

When event summarization is enabled, the Event Type and Occurrence Count attributes become meaningful. The Event Type attribute indicates the type of the event, being either an *Event* or a *Summary Event*. General events that correspond to records found in the log on a one-to-one basis are tagged as *Event*. Summary events that are sent at the end of the Summary Interval, are tagged as *Summary Event*.

The Occurrence Count attribute indicates the total amount of duplicate records found in the log for the event. Summary events include this count because it shows the number of events received that matched the summary event during the previous summary interval.

Thresholds and Summary Events

Regardless of the filter value as described in "Filtering events" on page 998, you always get the summary events at the end of each summary interval, for any event that occurred at least once during that interval. If you are not expecting the summary events, your thresholds can be accidentally triggered. To avoid this accidental triggering of a threshold, include a clause in the threshold for `Event Type== Event` or `Event Type!= Summary Event`.

Windows Event Log
The OS agent uses the .conf file to monitor events from the Windows Event Log.

The OS agent continues to use the `WINEVENTLOGS` configuration `(.conf)` file option to monitor events from the Windows Event Log. The agent monitors a comma-separated list of event logs as shown in the following example:

```
WINEVENTLOGS=System,Security,Application
```

The OS agent also continues to use the `WINEVENTLOGS=All` setting. The `All` setting refers to the following standard event logs: Security, Application, System, Directory, Domain Name System (DNS), and File Replication Service (FRS) that come with Windows versions earlier than 2008. However, all the event logs on the system are not checked.

The `UseNewEventLogAPI` configuration file tag allows the event log (Windows Event Log 2008 or later) to access any new logs added by Microsoft, and any Windows event logs created by other applications or the user. The new logs are listed by the `WINEVENTLOGS` keyword.

In the following example, the `UseNewEventLogAPI` tag is set to `y`.

```
UseNewEventLogAPI=y
WINEVENTLOGS=Microsoft-Windows-Hyper-V-Worker-Admin
```

In this example, the `Microsoft-Windows-Hyper-V/Admin` is monitored on a Windows system that has the Hyper-V role.

In the Windows Event Log, each event has the following fields in this order:

- Date in the following format: month, day, time, and year
- Event category as an integer
- Event Level
- Windows security ID. Any spaces in the Windows security ID are replaced by an underscore if `SpaceReplacement=TRUE` in the configuration `(.conf)` file.

  Note: `SpaceReplacement=TRUE` is the default if you set `UseNewEventLogAPI` to `y` in the `(.conf)` file (designated that you are using the event log).
- Windows source. Any spaces in the Windows source are replaced by an underscore if `SpaceReplacement=TRUE` in the configuration `(.conf)` file.
- Windows event log keywords. Any spaces in the Windows event log keywords are replaced by an underscore if `SpaceReplacement=TRUE` in the configuration `(.conf)` file.

  Note: The keyword field that is described here is new to the Windows 2008 version of Event Log. It did not exist in the previous Event Log, and so its presence prevents you from reusing your old Event Log format statements directly. They must be modified to account for this additional field.
- Windows event identifier
- Message text

For example, when an administrative user logs on to a Windows 2008 system, an event is generated in the Security log indicating the privileges that are assigned to the new user session:

```
Mar 22 13:58:35 2011 1 Information N/A Microsoft-Windows-
Security-Auditing Audit_Success 4672 Special privileges assigned to new logon.
S-1-5-21-586564200-1406810015-1408784414-500    Account Name:
Administrator   Account Domain:     MOLDOVA     Logon ID:
0xc39cb8e    Privileges:        SeSecurityPrivilege
SeBackupPrivilege          SeRestorePrivilege
SeTakeOwnershipPrivilege            SeDebugPrivilege
SeSystemEnvironmentPrivilege            SeLoadDriverPrivilege
SeImpersonatePrivilege
```

To capture all events that were created by the `Microsoft-Windows-Security-Auditing` event source, you write a format statement as shown here:

```
REGEX BaseAuditEvent
^([A-Z][a-z]{2} [0-9]{1,2} [0-9]{1,2}:[0-9]{2}:[0-9]{2} [0-9]
{4}) [0-9] (\S+) (\S+) Microsoft-Windows-Security-Auditing (\S+)
([0-9]+) (.*)
timestamp $1
severity $2
login $3
eventsource "Microsoft-Windows-Security-Auditing"
eventkeywords $4
eventid $5
msg $6
END
```

For the previous example event, the following example indicates the values that are assigned to slots:

```
timestamp=Mar 22 13:58:35 2011
severity=Information
login=N/A
eventsource=Microsoft-Windows-Security-Auditing
eventid=4672
msg="Special privileges assigned to new logon.
S-1-5-21-586564200-1406810015-1408784414-500    Account Name:
Administrator   Account Domain:     MOLDOVA     Logon ID:
0xc39cb8e    Privileges:        SeSecurityPrivilege
SeBackupPrivilege          SeRestorePrivilege
SeTakeOwnershipPrivilege            SeDebugPrivilege
```

```
SeSystemEnvironmentPrivilege          SeLoadDriverPrivilege
SeImpersonatePrivilege
```

Because it is difficult to anticipate exactly what these events look like, a useful approach to writing your regular expressions is to capture the actual events in a file. Then, you can examine the file, choose the events that you want the agent to capture, and write regular expressions to match these events. To capture all events from your Windows Event Log, use the following steps:

1. Create a format file that contains only one pattern that does not match anything, as shown in the following example:

   ```
   REGEX NoMatch
   This doesn't match anything
   END
   ```

2. Add the following setting to the configuration `(.conf)` file:

   ```
   UnmatchLog=C:/temp/evlog.unmatch
   ```

3. Run the agent and capture some sample events.

Event mapping

The Tivoli Event Integration Facility (EIF) interface is used to forward situation events to Tivoli Netcool/OMNIbus, Tivoli Enterprise Console, or Operations Analytics - Log Analysis.

EIF events specify an event class, and the event data is specified as name-value pairs that identify the name of an event slot and the value for the slot. An event class can have subclasses. Performance Management provides the base event class definitions and a set of base slots that are included in all monitoring events. Agents extend the base event classes to define subclasses that include agent-specific slots. For OS agent log file events, the event classes correspond to the agent attribute groups, and the agent-specific slots correspond to the attributes in the attribute group.

For events that are generated by thresholds in the LFAProfiles attribute group, events are sent by using the ITM_KLO_LFAPROFILES event class. This event class contains the following slots:

- node: STRING
- timestamp: STRING
- subnode_msn: STRING
- subnode_affinity: STRING
- subnode_type: STRING
- subnode_resource_name: STRING
- subnode_version: STRING
- subnode_config_file: STRING
- subnode_description: STRING
- subnode_description_enum: STRING

For events that are generated by thresholds in the Log File RegEx Statistics attribute group, events are sent by using the ITM_KLO_LOG_FILE_REGEX_STATISTICS event class. This event class contains the following slots:

- node: STRING
- timestamp: STRING
- table_name: STRING
- attrib_name: STRING
- filter_number: INTEGER
- average_processor_time: REAL

- average_processor_time_enum: STRING
- total_processor_time: REAL
- total_processor_time_enum: STRING
- max_processor_time: REAL
- max_processor_time_enum: STRING
- min_processor_time: REAL
- min_processor_time_enum: STRING
- filter_count: REAL
- filter_count_matched: REAL
- filter_count_unmatched: REAL
- regex_pattern: STRING
- last_matched_time: STRING
- last_matched_time_enum: STRING
- last_unmatched_time: STRING
- last_unmatched_time_enum: STRING
- result_type: INTEGER
- result_type_enum: STRING

For events that are generated by thresholds in the Log File Status attribute group, events are sent by using the ITM_KLO_LOG_FILE_STATUS event class. This event class contains the following slots:

- node: STRING
- timestamp: STRING
- table_name: STRING
- file_name: STRING
- regex_pattern: STRING
- file_type: INTEGER
- file_type_enum: STRING
- file_status: INTEGER
- file_status_enum: STRING
- num_records_matched: INTEGER
- num_records_not_matched: INTEGER
- num_records_not_matched_enum: STRING
- num_records_processed: INTEGER
- current_file_position: REAL
- current_file_position_enum: STRING
- current_file_size: REAL
- current_file_size_enum: STRING
- last_modification_time: STRING
- last_modification_time_enum: STRING
- codepage: STRING

For events that are generated by thresholds in the LogfileEvents attribute group, events are sent by using the ITM_KLO_LOGFILEEVENTS event class. This event class contains the following slots:

- node: STRING
- timestamp: STRING
- klo_class: STRING
- logname: STRING
- eifevent: STRING
- klo_msg: STRING
- customslot1: STRING
- customslot2: STRING
- customslot3: STRING
- customslot4: STRING
- customslot5: STRING
- customslot6: STRING
- customslot7: STRING
- customslot8: STRING
- customslot9: STRING
- customslot10: STRING
- occurrence_count: INTEGER
- occurrence_count_enum: STRING
- event_type: INTEGER
- event_type_enum: STRING
- custominteger1: REAL
- custominteger1_enum: STRING
- custominteger2: REAL
- custominteger2_enum: STRING
- custominteger3: REAL
- custominteger3_enum: STRING
- remotehost: STRING

For events that are generated by thresholds in the LogfileProfileEvents attribute group, events are sent by using the ITM_KLO_LOGFILEPROFILEEVENTS event class. This event class contains the following slots:

- node: STRING
- timestamp: STRING
- klo_class: STRING
- logname: STRING
- eifevent: STRING
- klo_msg: STRING
- customslot1: STRING
- customslot2: STRING

- customslot3: STRING
- customslot4: STRING
- customslot5: STRING
- customslot6: STRING
- customslot7: STRING
- customslot8: STRING
- customslot9: STRING
- customslot10: STRING
- occurrence_count: INTEGER
- occurrence_count_enum: STRING
- event_type: INTEGER
- event_type_enum: STRING
- custominteger1: REAL
- custominteger1_enum: STRING
- custominteger2: REAL
- custominteger2_enum: STRING
- custominteger3: REAL
- custominteger3_enum: STRING
- remotehost: STRING

For events that are generated by thresholds in the Performance Object Status attribute group, events are sent by using the ITM_KLO_PERFORMANCE_OBJECT_STATUS event class. This event class contains the following slots:

- node: STRING
- timestamp: STRING
- query_name: STRING
- object_name: STRING
- object_type: INTEGER
- object_type_enum: STRING
- object_status: INTEGER
- object_status_enum: STRING
- error_code: INTEGER
- error_code_enum: STRING
- last_collection_start: STRING
- last_collection_start_enum: STRING
- last_collection_finished: STRING
- last_collection_finished_enum: STRING
- last_collection_duration: REAL
- average_collection_duration: REAL
- average_collection_duration_enum: STRING

- refresh_interval: INTEGER
- number_of_collections: INTEGER
- cache_hits: INTEGER
- cache_misses: INTEGER
- cache_hit_percent: REAL
- intervals_skipped: INTEGER

For events that are generated by thresholds in the pro Performance Object Status attribute group, events are sent by using the ITM_KLO_PRO_PERFORMANCE_OBJECT_STATUS event class. This event class contains the following slots:

- node: STRING
- timestamp: STRING
- query_name: STRING
- object_name: STRING
- object_type: INTEGER
- object_type_enum: STRING
- object_status: INTEGER
- object_status_enum: STRING
- error_code: INTEGER
- error_code_enum: STRING
- last_collection_start: STRING
- last_collection_start_enum: STRING
- last_collection_finished: STRING
- last_collection_finished_enum: STRING
- last_collection_duration: REAL
- average_collection_duration: REAL
- average_collection_duration_enum: STRING
- refresh_interval: INTEGER
- number_of_collections: INTEGER
- cache_hits: INTEGER
- cache_misses: INTEGER
- cache_hit_percent: REAL
- intervals_skipped: INTEGER

For events that are generated by thresholds in the Thread Pool Status attribute group, events are sent by using the ITM_KLO_THREAD_POOL_STATUS event class. This event class contains the following slots:

- node: STRING
- timestamp: STRING
- thread_pool_size: INTEGER
- thread_pool_size_enum: STRING
- thread_pool_max_size: INTEGER
- thread_pool_max_size_enum: STRING

- thread_pool_active_threads: INTEGER
- thread_pool_active_threads_enum: STRING
- thread_pool_avg_active_threads: REAL
- thread_pool_avg_active_threads_enum: STRING
- thread_pool_min_active_threads: INTEGER
- thread_pool_min_active_threads_enum: STRING
- thread_pool_max_active_threads: INTEGER
- thread_pool_max_active_threads_enum: STRING
- thread_pool_queue_length: INTEGER
- thread_pool_queue_length_enum: STRING
- thread_pool_avg_queue_length: REAL
- thread_pool_avg_queue_length_enum: STRING
- thread_pool_min_queue_length: INTEGER
- thread_pool_min_queue_length_enum: STRING
- thread_pool_max_queue_length: INTEGER
- thread_pool_max_queue_length_enum: STRING
- thread_pool_avg_job_wait: REAL
- thread_pool_avg_job_wait_enum: STRING
- thread_pool_total_jobs: INTEGER
- thread_pool_total_jobs_enum: STRING

Managing synthetic transactions and events with Website Monitoring

Create synthetic transactions that monitor the performance and availability of internal applications, external applications, and public-facing web applications at different locations.

Create a *synthetic transaction* in the Synthetic Script Manager. Generate simple scripts in the Synthetic Script Manager to test the availability of an application, or use Selenium IDE to record synthetic scripts that replicate different user actions with an application. Then, configure a synthetic transaction to play back your script at specific intervals and playback locations.

Important: Only existing users of the IBM Website Monitoring on Cloud add-on can use the Synthetic Playback agent and the Synthetic Script Manager. Website Monitoring has been replaced by IBM Cloud Availability Monitoring for the August 2017 release. For more information, see "About Availability Monitoring" on page 1024.

Your available playback locations are the locations where you installed the Monitoring Agent for Synthetic Playback and the 15 points of presence (PoPs) that are provided for monitoring public-facing web applications. PoPs are available for the following locations:

- Amsterdam
- Chennai
- Dallas
- Frankfurt
- Hong Kong
- London
- Melbourne
- Mexico

- Paris
- San Jose
- Sao Paolo
- Singapore
- Tokyo
- Toronto
- Washington

Create thresholds and resource groups to raise events and notify stakeholders when your applications are slow or unavailable. View performance data and generate historical reports in the Application Performance Dashboard.

If you monitor end-user response time for an application with the Response Time agent, you can view KPIs for both end-user and synthetic transactions in the Application Performance Dashboard. Add synthetic transactions as components to the application that you are monitoring with the Response Time agent.

Note: To work in Synthetics Script Manager, you must be a member of a role that has view permission for Synthetic Script Manager and Agent Configuration. For more information, see "Roles and permissions" on page 982.

Recording synthetic scripts
Record a synthetic script by using the Firefox web browser and the Selenium IDE add-on.With Selenium IDE, you can record user actions on a web page, such as loading a page, clicking a link, or selecting an object. When Selenium IDE is recording, it generates a command for each user action in a script. Then, using Synthetic Script Manager, you can configure scripts to simulate user behavior at your website, at set intervals and at different locations.

Before you begin

You must use the Firefox web browser when recording scripts

Selenium IDE is available only as a Firefox add-on. If Selenium IDE is not installed or running, complete the following steps:

1. Ensure that you are running a version of Firefox 60 or later that supports Selenium IDE 3.2.X or 3.3.X. If you have a later version of Selenium IDE, it is not supported; you must uninstall it and install version 3.2.X or 3.3.X.

   Note: By default, Selenium IDE is automatically updated after you install version 3.2.X or 3.3.X. Turn off automatic updates for Selenium IDE to prevent version upgrades.
2. Download and install Selenium IDE 3.2.X or 3.3.X from the **Selenium** home page (https://addons.mozilla.org/firefox/addon/selenium-ide/versions/). Allow Selenium IDE to install all plug-ins.
3. After Selenium IDE is installed, restart Firefox.
4. Navigate to the web page that you want to test and close any other tabs. To open Selenium IDE, click **Tools** > **Selenium IDE**. In the **Selenium IDE** window, ensure that the **Base URL** field contains the URL of the displayed web page. Selenium IDE starts recording all user actions on the displayed web page.

Selenium .side script format

Scripts created with newer versions of Selenium use the `..side` format. With Selenium IDE 3.2.X or 3.3.X, you can import older scripts that were created with the `.html` format and save to the `.side` format. For more information, see "Updating scripts from earlier Selenium IDE versions" on page 1010.

If you will be using Selenium `.side` scripts, you must first install these updates:

- IBM Cloud Application Performance Management V8.1.4.0 Synthetic Playback agent Interim Fix 5 or later on the systems where you installed the Synthetic Playback agent.
- Contact IBM to ensure your Cloud APM subscription has been updated to IBM IBM Cloud Application Performance Management, Private Cloud APM V8.1.4.0 Server Interim Fix 8 or later.
- If you are using an Availability Monitoring private point of presence (PoP), check that the synthetic PoP build number is APM_201903090832 or later by entering the **cat build.info** command from the PoP installation directory. Earlier build versions do not support .side format.

Interim fixes for Cloud APM V8.1.4.0 are available to download from IBM Support > Fix Central > IBM APM 8.1.4.0.

About this task

In this task, you perform user actions on a web page and use Selenium IDE to record these actions as commands in a simple script. You can use scripts to monitor the performance and availability of your web application in the Application Performance Dashboard.

Procedure

Complete the following steps to record a script of user actions on a web page:

1. Click **Record** to start recording a script. Perform user actions on your web page, such as clicking a link.

   For every user action on a web page, Selenium IDE records a command and adds it to a script.

   For example, complete the following actions to record when a user loads the IBM Marketplace web page and navigates to a free trial of Cloud APM, in a script:

   Table 248. Recorded user actions and Selenium IDE commands

User action	Commands added to script
To record when the Cloud APM web page on the IBM Marketplace website opens, open the IBM Marketplace web page. Right-click anywhere on the displayed web page and select **open**.	open
To ensure that the script checks that the web page loads, right-click the title text of the web page (IBM Cloud Application Performance Management) and click **Show All Available Commands** > **verifyTitle IBM Cloud Application Performance Management**.	verifyTitle
To record when the user clicks a link to view details about Cloud APM, click the **Details** link. The **Details** page loads.	clickAndWait
To ensure that the script checks that the **Details** page has loaded, right-click on the "Feature spotlights" heading and select **Show All Available Commands** > **verifyText css=h2.heading--TERTIARY**.	verifyText
To record when the user clicks a link to view details about how to purchase Cloud APM, click the **Purchase** link. The **Purchase** page loads.	clickandWait
To record when the user clicks a button to register for a free trial of Cloud APM, click the **Try Free** button.	click

2. In the Selenium IDE window, click **Record** to stop the recording. Click the **Save Project** tool, give your script a meaningful name, and save as a .side file (such as open_webpage.side).
3. In the Selenium IDE window, review your recorded script. Click the **Table** tab to display the script in a table format. In the Selenium IDE window, click **Play Current Test Case** to test the playback of the script that you recorded.

   In this example, Selenium IDE displays the script of user actions on the IBM Marketplace website, as described in step 1.

Table 249. Example of a Selenium IDE script recording of user actions on the IBM Marketplace website

Command	Target	Value
`open`	`/`	
`verifyTitle`	`IBM Cloud Application Performance Management`	
`clickAndWait`	`css=ul > #details > a`	
`verifyText`	`css=h2.heading--TERTIARY`	Feature spotlights
`clickAndWait`	`css=ul > #purchase > a`	
`click`	`link=Try Free`	

Results
You recorded a script that you can use to monitor the performance and availability of a web application.

What to do next

If you recorded a complex script, you can organize your script into simpler scripts, where each script represents a specific business process or user action on your web application.

Use the Synthetic Script Manager to upload your script file to a new or existing synthetic transaction.

Structuring complex scripts

Organize a complex script into multiple scripts; then, save scripts together in a collection of scripts called a *test suite*.

About this task

If you create a complex script, you can organize that script into simple scripts that represent different business or user processes on your web application. Save the scripts together as a test suite. You can then use these scripts to monitor the performance and availability of your web application in response to specific user actions in the Application Performance Dashboard.

There should be only one test suite and all tests should be added into it.

Important: It is good practice to organize complex scripts into separate scripts, where each script represents a typical user or business process that you want to monitor. For example, create separate scripts that record when a user logs in to a website, or searches for an item. If you organize your scripts according to user or business processes, you can then monitor the response of your web application to these specific processes in the Application Performance Dashboard.

Procedure

To organize your complex script into separate scripts, and save your scripts as a test suite, complete the following steps:

1. To create a separate script for each user process that is recorded in your script, click **Tests** > **+** in Selenium IDE. Give each script a meaningful name that describes the user process and save each script as a `.side` file, such as `load_homepage.side`.

   For more information, see "Recording synthetic scripts" on page 1007.

   Important: The name that you give to your script in Selenium IDE is the name that identifies the recorded business or user process that you monitor in the Application Performance Dashboard.

2. In Selenium IDE, open a complex script that you recorded previously. Organize your script commands into separate scripts, according to different user actions. **Cut** commands from the original complex script in the **Test Case** window and **Paste** commands into the different **Test Case** window.

For example, the complex script example in Recording synthetic scripts contains Selenium IDE commands for three different user processes.

- Open the Cloud APM home page on the IBM Marketplace website.
- Open the **Details** page on IBM Marketplace.
- Open the **Pricing** page and record when the user opens the registration page for a free trial.

The user actions are then organized into three different scripts.

Table 250. Sample script for opening the IBM Marketplace page (`load_homepage.side`)

Command	**Target**	**Value**
`open`	`/`	
`verifyTitle`	`IBM Cloud Application Performance Management`	

*Table 251. Sample script for opening the **Details** page on IBM Marketplace (`load_products.side`)*

Command	**Target**	**Value**
`clickAndWait`	`css=ul > #details > a`	
`verifyText`	`css=h2.heading--TERTIARY`	Feature spotlights

*Table 252. Sample script for opening the **Purchase** and trial registration pages on IBM Marketplace (`load_APM.side`)*

Command	**Target**	**Value**
`clickAndWait`	`css=ul > #purchase > a`	
`click`	`link=Try Free`	

3. To put individual test cases into a test suite, change to the **Test suite** window and add tests to the test suite according to the business logic sequence. Finally, click the **Save Project** tool to save the test suite and all tests in the test suite to a `.side` file.
   As an example, consider the logical sequence Load_URL, `Select Manage inventory`, `Select IBM Machine Type`. When we add these test cases to the test suite, we first check Load_URL, followed by `Select Manage inventory`, then `Select IBM Machine Type`

Results

You recorded a set of scripts that you can use to monitor the performance and availability of your web applications. Use the Synthetic Script Manager to upload your `.side` test suite of scripts to a new or existing synthetic transaction.

Updating scripts from earlier Selenium IDE versions

The supported Selenium IDE versions 2.2.X and 3.2.X use the `.side` format for recording synthetic scripts rather than the `.html` format used by older versions of Selenium IDE. If you have existing `.html` scripts, you can still use them. Scripts that were created with older versions of Selenium IDE might not work fully with the latest Firefox and Selenium drivers used by IBM Cloud Availability Monitoring. In some instances, you might want to edit the `.html` scripts, re-record them in the new `.side` format, or import the `.html` script and save to the `.side` format.

Procedure

- Exception: If you want to interact with the `Select2` element, do not use the **`select`** command (see https://github.com/SeleniumHQ/selenium-ide).

  The old script is

```
<td>select</td>
<td>id=country</td>
<td>label=United States</td>
```

It should be changed to

```
<tr>
   <td>runScript</td>
   <td>window.scrollTo(0,810)</td>
   <td></td>
</tr>
<tr>
   <td>click</td>
   <td>id=select2-country-container</td>
   <td></td>
</tr>
<tr>
   <td>click</td>
   <td>xpath=(//ul[@id='select2-country-results']/li[text() = 'United States'])</td>
   <td></td>
</tr>
```

- Limitation: `.side` scripts recorded with Selenium IDE 3.2.X or 3.3.X are supported; the **`linkText`** locator is not supported.

Managing synthetic transactions

Use the Synthetic Script Manager to create, configure, and delete synthetic transactions.

To display the Synthetic Script Manager, click the **System Configuration** icon and select **Synthetic Script Manager**. To work in Synthetics Script Manager, you must be a member of a role that has view permission for **Synthetic Script Manger** and **Agent Configuration**. For more information, see "Roles and permissions" on page 982.

You can view data about your synthetic transaction usage for public external facing web applications for the current month. View the number of performed playback instances and predicted playback instances based on your current configuration for the current month in the Monthly Playback Usage table in the Synthetic Script Manager.

Important: To create synthetic transactions for private internal web applications and private external web applications, you must install the Synthetic Playback agent at every location that contains an web application that you want to monitor.

You can perform the following tasks with the Synthetic Script Manager:

- Create and edit a synthetic transaction.
- Configure synthetic transaction variables.
- Delete a synthetic transaction.

Creating and editing a synthetic transaction

To view data about the performance and availability of a web application, you must first create a synthetic transaction in the Synthetic Script Manager.

About this task

Use the Synthetic Script Manager to create, edit, and configure a synthetic transaction. Enter the URL of a web application in the Synthetic Script Editor to generate a simple script for your synthetic transaction. To simulate complex user processes, upload a synthetic script to a synthetic transaction in the Synthetic Script Editor. Then, configure your synthetic transaction to run at regular intervals and at different locations.

Procedure

To create a transaction, or to edit an existing transaction, complete the following steps:

1. Optional: If the Synthetic Script Manager is not displayed, click the **System Configuration** icon and select **Synthetic Script Manager**.

2. To create a new transaction, click the **New** icon ⊕. To edit an existing transaction, click the **Edit** icon ✎.
3. In the **Synthetic Script Editor**, click the **Upload a Script** tab and enter a transaction name in the **Transaction Name** text box. Enter a description of your transaction in the **Description** text box.

   Important: Do not give your transaction the same name as a transaction that you deleted in the last 24 hours. If you give your transaction the same name as a recently deleted transaction, data from both transactions are attributed incorrectly to that transaction name in the Application Performance Dashboard.
4. To generate a simple script to test a web application, select **Enter the URL of web page to test** and enter a URL. The Synthetic Script Manager generates a simple synthetic script based on that URL.
5. To assign a previously created script file to your transaction, select **Upload script file**. Click **Upload Script** to browse for scripts on your system. Choose a script and click **Open**.

   Important: The synthetic script file must be one of the following file types:

   - `.html`
   - `.zip`

   Save simple individual scripts (test cases) as `.html` files. Compress test cases and test suites together in a `.zip` file.
6. To configure simultaneous or staggered playback of a synthetic transaction, click the **Schedule a Script** tab. Select **Simultaneous** to execute the transaction from all locations simultaneously, or select **Staggered** to execute the transaction from a different location at each interval.
7. To choose how often a script runs, click the **Schedule a Script** tab. Click the **Interval** text box and enter a number, based on how often you want to monitor your web application. Choose an interval length between `1` and `60` minutes.

   Note: Large or complex scripts can take longer to run. Choose a longer interval length for large or complex scripts.
8. To choose the playback locations for your script, click the **Schedule a Script** tab and then select the data center locations and agent installation locations where you want your script to run.
9. To set response time thresholds for synthetic transactions and subtransactions, click the **Advanced Settings** tab; then, click and expand a synthetic transaction to reveal all subtransactions. Double-click the response time threshold value and enter a value. Choose a value between `0` and `3600` seconds. If you do not want to set a threshold, enter `0`. The default response time threshold value is `10` seconds.

   Note: Some commands can take longer than others. Choose a response time threshold that is suitable for the command that you want to test. If your transaction tests how long a web page takes to open, choose a longer response time.
10. To finish creating or editing your transaction, click **Save**.

Results

You configured a synthetic transaction. The synthetic transaction is listed in the Synthetic Script Manager.

What to do next

You can view metrics and KPIs recorded by a synthetic transaction in the Application Performance Dashboard. You can also add transactions as components to an application, and view all synthetic transactions that are associated with that application.

Important: When you first add a synthetic transaction, a blank space might appear in the **Availability Over Time** group widget in the Application Performance Dashboard. The blank space disappears quickly, when the server receives the first playback results of the transaction.

Configuring synthetic transaction variables

Use the Synthetic Script Manager to update variable values, such as user names and passwords that are stored in synthetic scripts, without the need to edit the script files. The values of your variables can be unique for each playback location.Configure variables for your synthetic scripts when your web applications require different variable values at different locations. For example, if your web application does not allow the same login details at different locations, use the Synthetic Script Manager to provide different login details at each location. You can create variables in synthetic scripts by using the `store` command in the Selenium-IDE plug-in.

About this task

In this task, use the Synthetic Script Manager to configure variables that are stored in your synthetic script.

Procedure

To configure the variables of a synthetic transaction, complete the following steps:

1. If the Synthetic Script Manager is not displayed, click the **System Configuration** icon and select **Synthetic Script Manager**. Select a synthetic transaction from the list and click the **Edit** icon.
2. Select the playback locations for your synthetic transaction.
3. Click the **Advanced Settings** tab. If the synthetic script contains variables, you can edit these variables in the **Configure Variable Substitutions for Different Locations** window. To edit a variable, double-click the value. To finish, click **Save**.

   For example, the following script contains the variables *username* and *password*. The values of these variables, `user1` and `pass`, are saved by using the `store` command in Selenium-IDE. The variables have the same value at two locations, Dallas and San Jose.

   Table 253. Example of a script with variables

Command	Target	Value
`store`	`user1`	*username*
`store`	`pass`	*password*
`type`	`id=j_username`	*${username}*
`type`	`id=j_password`	*${password}*

   The values of the script variables are displayed in the **Configure Variable Substitutions for Different Locations** window. Change the value of *username* at the location `Dallas` from `user1` to `admin1` so that the synthetic transaction uses different login details at different locations.

   Table 254. Script variable values at different locations

Location	username	password
San Jose	user1	pass
Dallas	admin1	pass

Results

You configured the variables of a synthetic transaction. You can now use this synthetic transaction to test the performance and availability of a web application at different locations.

What to do next

You can view metrics and KPIs recorded by a synthetic transaction in the Application Performance Dashboard. You can also add transactions as components to an application, and view all synthetic transactions that are associated with that application.

Filtering URLs and domain names for your synthetic transactions
Use the Synthetic Script Manager to allow or block access to specific URLs and domains by adding rules to the whitelist and blacklist for your test.

About this task

You can control which dependencies and resources contribute to the response times of your tested web applications. Use the blacklist to filter out requests from specified domains to remove those requests from your measured response times. Use the whitelist to include requests from specified domains to add those requests to your measured response times. Use the blacklist and whitelist to filter out or include dependencies that are associated with your web application, such as third-party metrics.

The **Blacklist** field contains a list of rules that block access to specified URLs and domains.

The **Whitelist** field contains a list of rules that allow access to specified URLs and domains.

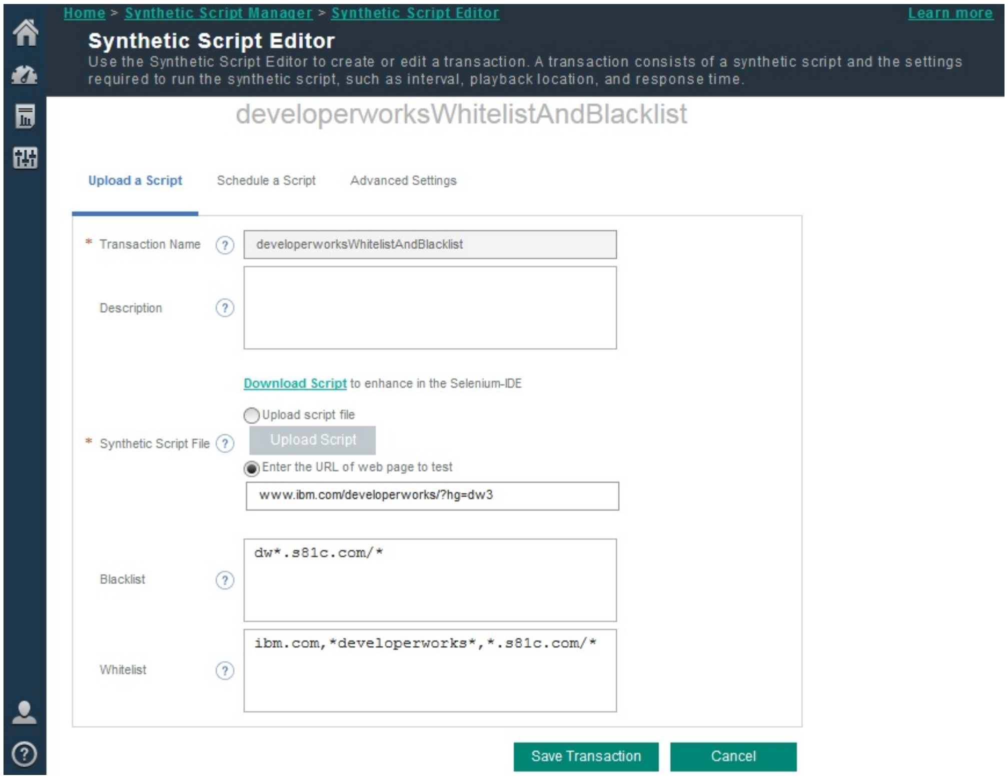
Use commas (,) to separate rules in your blacklist and whitelist. Use the wildcard symbol (*) to filter domain names and URLs. For example, `ibm.com, *.bluemix.net, *developerworks*, *.profile.*.cloundfront.net/*.png`.

Note: If you configure both a blacklist and whitelist for your synthetic transaction, the blacklist has higher priority.

Procedure

To add a rule to the blacklist or whitelist for your synthetic transaction, complete the following steps:

1. If the Synthetic Script Manager is not displayed, click the **System Configuration** icon and select **Synthetic Script Manager**.
2. To configure your blacklist or whitelist for an existing transaction, click the **Edit** icon. If no transactions are listed, click the **New** icon to create a new transaction.
3. In the **Synthetic Script Editor**, click the **Upload a Script** tab and enter a transaction name in the **Transaction Name** text box. Add rules to the **Blacklist** and **Whitelist** text boxes.



Synthetic Script Editor

Use the Synthetic Script Editor to create or edit a transaction. A transaction consists of a synthetic script and the settings required to run the synthetic script, such as interval, playback location, and response time.

developerworksWhitelistAndBlacklist

* Transaction Name: developerworksWhitelistAndBlacklist

Description:

Download Script to enhance in the Selenium-IDE

* Synthetic Script File:
○ Upload script file

● Enter the URL of web page to test
www.ibm.com/developerworks/?hg=dw3

Blacklist: dw*.s81c.com/*

Whitelist: ibm.com,*developerworks*,*.s81c.com/*

Results

Your synthetic transaction now filters out unwanted requests for your web application and allows access to other specific requests.

Hiding passwords in the Synthetic Script Manager

Store passwords as variables in your synthetic scripts to hide password values in the Synthetic Script Manager.

Before you begin

This procedure requires you to edit a synthetic script. Record a synthetic script by using Selenium IDE. For more information, see “Recording synthetic scripts” on page 1007.

About this task

Manually modify your synthetic scripts in the Selenium IDE to store your password as a variable. You can then create synthetic transactions with hidden passwords in the Synthetic Script Manager. Hidden passwords are displayed as asterisks in the Synthetic Script Manager.

Important: It is recommended that you store your passwords in synthetic scripts so that the password values are not displayed in the Synthetic Script Manager. Hidden passwords make your web applications more secure by preventing others from viewing passwords.

Procedure

1. Open the script that you want to modify in Selenium IDE. Use the `store` command to assign a password to the variable *password*, following the example that is described in this step; then, save the script.

   Important: You must store the password as the variable name *password* so that the password is not displayed in the Synthetic Script Manager.

   For example, the following synthetic script contains a user name *test@example.com* and a password value *ibm4value*.

   ```
   <tr>
      <td>type</td>
      <td>id=username</td>
      <td>test@example.com</td>
   </tr>
   <tr>
      <td>type</td>
      <td>id=password</td>
      <td>ibm4value</td>
   </tr>
   ```

   The following script shows how to assign the password value *ibm4value* to the variable *password* by using the `store` command.

   ```
   <tr>
      <td>store</td>
      <td>ibm4value</td>
      <td>password</td>
   </tr>
   <tr>
      <td>type</td>
      <td>id=username</td>
      <td>test@example.com</td>
   </tr>
   <tr>
      <td>type</td>
      <td>id=password</td>
      <td>$(password)</td>
   </tr>
   ```

2. Optional: To hide the password at script level, assign a blank value to the variable *password* by using the `store` command; then, save the script.

   You can set the password later in the Synthetic Script Manager.

For example, the following script shows how to assign a blank value to the variable *password* by using the `store` command.

```
<tr>
   <td>store</td>
   <td></td>
   <td>password</td>
</tr>
<tr>
   <td>type</td>
   <td>id=username</td>
   <td>test@example.com</td>
</tr>
<tr>
   <td>type</td>
   <td>id=password</td>
   <td>$(password)</td>
</tr>
```

3. Log in to the Cloud APM console and open the **Synthetic Script Manager**. Create a transaction and upload your script to that transaction. Click the **Advanced Settings** tab.

   The password for each location is hidden. You can change the password for each location. For more information, see "Managing synthetic transactions" on page 1011.

Deleting a synthetic transaction

Use the Synthetic Script Manager to delete synthetic transactions.

Procedure

To delete a synthetic transaction, complete the following steps:

1. If a synthetic transaction is assigned to an application, you must first remove the transaction from that application. On the Application Performance Dashboard, click and expand **All My Applications** and then click the application that is associated with the synthetic transaction that you want to delete.

   Click the **Edit** icon ✎. In the **Edit Application** window, remove the synthetic transaction component from the application. For more information, see Managing Applications.

   The synthetic transaction can now be deleted.

2. On the navigation bar, click the **System Configuration** icon and select **Synthetic Script Manager**.

   Select a synthetic transaction and then click the **Delete** icon ⊖. To confirm that you want to delete this synthetic transaction, click **OK**.

Results

The synthetic transaction is deleted.

Viewing synthetic transaction data in the Application Performance Dashboard

View synthetic transaction data in the Application Performance Dashboard. Associate synthetic transactions with a new or existing application, and view all associated synthetic transactions together in the Application Performance Dashboard.

About this task

You can view synthetic transaction data in the **My Transactions** window in the Application Performance Dashboard.

You can also create groups of synthetic transactions by associating your transactions with an application. Use the **Add Application** or **Edit Application** tool in the Application Performance Dashboard to add synthetic transactions as components to a new or existing web application. You can then view data for all synthetic transactions that are associated with that application together in the Application Performance Dashboard.

If you are already using the Response Time agent to monitor user response time for an application, you can add a synthetic transaction to that application. You can then view more metrics and KPIs for that application in the Application Performance Dashboard.

Procedure

- To view synthetic transactions, complete the following step:
  a) Click the **Performance** icon and select **Application Performance Dashboard**. In the **Applications** window, expand **All My Applications** and select **My Transactions**. In the **Groups** window, expand **Transactions** and select **Synthetic Transactions**.
  b) Click a synthetic transaction to view availability and performance data for that transaction, along with a graph of response times for transaction instances over a defined period.
- To associate synthetic transactions with an application, complete the following steps:
  a) Click the **Performance** icon and select **Application Performance Dashboard**. Select and edit an existing application or create a new application. For more information, see Managing applications.
  b) In the **Add Application** window, click the **Add Components** icon + and select **Synthetic Transactions** from the list of components. In the **Component Editor** window, select a synthetic transaction and click **Add** to associate the synthetic transaction with the application.
  c) Click **Back**. Click **Close** to close the **Component Editor** window. Click **Save**. To add another synthetic transaction as a component, repeat steps 1 - 3.

Results

You associated a synthetic transaction with an application. You can now view the application and its associated synthetic transactions in the Application Performance Dashboard. For more information, see Managing applications.

Note: When you associate a synthetic transaction with an application, the initial availability of that application is unknown. The status can take several minutes to update.

Managing synthetic events

Use the Threshold Manager and the Resource Group manager to configure thresholds and assign them to synthetic transactions. Synthetic events are generated when the value of a transaction attribute matches the condition that is defined in the threshold. You can monitor synthetic events in the Application Performance Dashboard.

Creating a threshold for synthetic transactions

Use the Threshold Manager to create thresholds for synthetic transactions. Thresholds are used to compare attribute values with the values set in the threshold. If the sampled value satisfies the comparison, an event is generated.

About this task

Thresholds allow users to monitor when applications report specific conditions. For example, you can create a threshold to monitor the time that a website takes to respond to a particular user command. If the website takes longer than the time specified by your threshold, a synthetic event is generated.

Procedure

To create a threshold and associate it with one or more synthetic transactions, complete the following steps:

1. On the navigation bar, click the **System Configuration** icon and select **Threshold Manager**. Set the **Data Source** type as **Synthetic Transaction**.
2. Create a threshold. For more information, see "Threshold Manager" on page 967.
3. To associate the threshold with a transaction, select **KSO TRANSACTION** as the **Data Set** and then select **TRANSNAME** as **Display Item**. For the **Logical operator**, select **And (&)**.

   Note: You must select **TRANSNAME** as the **Display Item**. If you do not select **TRANSNAME**, you cannot view synthetic events on the Application Performance Dashboard.
4. To add a condition, click the **New Condition** icon ⊕. In the New Condition box, select an **Attribute** and an **Operator**. Then, enter a threshold value for **Value**. To add this condition to the threshold, click **OK**.

For example, to add a threshold condition that generates a synthetic event when over 50% of transactions are slow, select **PSLOW** as **Attribute** and then select **Greater than** as **Operator**. To set the percentage of slow transactions to generate an event, enter `50` as the **Value**.

5. To define more threshold attributes, add more conditions to your threshold.
6. When you are finished, click **Save**. If you do not want to assign the threshold to a resource group, click **OK**.

Results

You created a threshold and associated it with a synthetic transaction. When the threshold conditions are met, an event is generated. You can monitor events in the Application Performance Dashboard, in the **Events** tab.

What to do next

You can group your synthetic transactions into resource groups.

Creating a resource group for synthetic transactions

Organize your synthetic transactions into a resource group and apply thresholds to all transactions in that resource group.

Before you begin

Create a threshold to apply to all synthetic transactions in your resource group.

About this task

You can organize your synthetic transactions into resource groups, and apply thresholds to every synthetic transaction in that resource group. Use the Resource Group manager to create a resource group and assign a threshold to that resource group. Then, assign one or more synthetic transactions subnodes to that resource group. The threshold that is associated with the resource group now applies to all associated synthetic transactions.

Procedure

To create a resource group for synthetic transactions, complete the following steps:

1. Click the **System Configuration** icon and select **Resource Group Manager**. Create a resource group, or edit an existing resource group. For more information, see "Resource Group Manager" on page 962.

To create a resource group for synthetic transactions, complete the following steps:

2. Give your resource group a name and description. Assign a threshold to your resource group in the **Threshold Assignment** table and click **Save**. In the Resource Group Manager, select your resource group again and click the **Edit** icon .
3. Associate your resource group with synthetic transaction subnodes from the **Resource Assignment** table, and click **Save**.

   The format of synthetic transaction subnodes is `SO:`*`TransactionName`*. For example, if you have a transaction `open_webpage`, the available subnode is called `SO:open_webpage`.

Results

You organized your synthetic transactions into a resource group, and you applied a threshold to every transaction in that resource group.

Creating critical thresholds for simultaneous and staggered synthetic transactions

Use the Threshold Manager to create critical thresholds for simultaneous and staggered synthetic transactions.

About this task

Create thresholds that notify stakeholders when consecutive staggered transactions fail, or when simultaneous transactions fail at all playback locations. For more information, see “Creating and editing a synthetic transaction” on page 1011.

Procedure

To create a critical threshold that creates an event when staggered transaction playback instances fail, complete the following steps:

1. Create a threshold for synthetic transactions in the Threshold Manager. For more information, see “Creating a threshold for synthetic transactions” on page 1017.
2. In the Threshold Manager, select **Critical** as the **Severity** and enter 1 minute as the threshold **Interval (HHMMSS)**. Use the following formula to determine the **Required consecutive samples**:

   Required consecutive samples = (playback interval * expected consecutive failures) - 1

   For example, if the playback interval of the synthetic transaction you want to monitor is 5 minutes, and you want to detect 8 consecutive playback failures, you must set **Required consecutive samples** as (5 * 8) - 1 = 39.
3. Add a condition. In the New Condition box, select **LOCATION** as **Attribute**, select **Equals** as **Operator**, and enter None as the **Value**. Add a second condition, and set **PFAILED** = `100`. Save the threshold.
4. On the navigation bar, open the **Resource Group Manager**. Create a resource group. Assign one or more staggered synthetic transactions to your resource group, then assign the threshold that you created in steps 1-3 to your resource group. Save the resource group. For more information, see “Creating a resource group for synthetic transactions” on page 1018.

To create a critical threshold that creates an event when simultaneous transaction playback instances fail at several locations, complete the following steps:

5. Create a threshold for synthetic transactions in the Threshold Manager. For more information, see “Creating a threshold for synthetic transactions” on page 1017.
6. In the Threshold Manager, select **Critical** as the **Severity** and enter 1 minute as the **Interval (HHMMSS)**. Set the **Required consecutive samples** as the same value as the playback interval of the transaction that you want to monitor.

   For example, if the playback interval of the synthetic transaction that you want to monitor is 5 minutes, set **Required consecutive samples** as 5.
7. Add a condition. In the New Condition box, select **LOCATION** as **Attribute**, select **Equals** as **Operator**, and enter None as the **Value**. Add a second condition, and set **PFAILED** = `100`. Save the threshold.
8. Create a resource group. Assign one or more synthetic transactions and the new critical threshold to your resource group. For more information, see “Creating a resource group for synthetic transactions” on page 1018.

Results

You created a critical threshold for a staggered or simultaneous synthetic transaction. When the threshold conditions are met, an event is generated. You can monitor events in the Application Performance Dashboard, in the **Events** tab.

Managing email notifications for synthetic events

Use the Resource Group Manager and IBM Alert Notification to generate email notifications when the performance of your application exceeds thresholds.

Before you begin

To configure email notifications for synthetic events, you must first enable Alert Notification for your subscription. For more information, see Alert Notification in the IBM Knowledge Center.

About this task

In the Resource Group Manager, use Alert Notification to configure email notifications. Email notifications are generated when the performance of your applications meets the conditions set by the thresholds that are associated with your resource group.

Procedure

To manage email notifications, complete the following steps:

1. Click the **System Configuration** icon and select **Resource Group Manager**. Create or edit a resource group.
2. Select a threshold in the **Threshold Assignment** table and click **Save**. Then, associate your resource group with a Synthetic Playback resource or a Synthetic Events agent resource in the **Resource Assignment** table. Click **Save**. In the Resource Group Manager, select your resource group again and click the **Edit** icon .
3. To start Alert Notification in a new tab, click **Configure email notification** in the Resource Group Manager. In Alert Notification, click **Users** on the navigation bar to create or edit recipients of email notifications. For more information, see Alert Notification in the IBM Knowledge Center.
4. In Alert Notification, click **Notification Policies** on the navigation bar. The **Notifications Editor** is automatically provisioned with a new policy. The name and filter of the policy are derived from the resource group. Click **Add Rule** to define the conditions that determine when an email notification is sent. For more information, see Alert Notification in the IBM Knowledge Center.
5. To finish configuring your email notifications, click **Save**. Your policy is listed in a table in the **Notification Policies** tab.

Guidelines to maximize agent and server performance for log file monitoring

To ensure that you get maximum performance from the OS agents and the Performance Management server, you must define the regular expressions in the format (.fmt) file and also limit the number of log file monitoring events that are reported to the Cloud APM console.

Guidelines to define regular expressions in the `.fmt` file

The `.fmt` file uses regular expressions that require a large amount of CPU processing. To improve agent and server performance, minimize the time spent checking records in the monitoring source log against regular expression in the `.fmt` file by using the following guidelines:

Minimize the use of multi-line patterns.

Multi-line patterns are expensive because the agent must determine what the records are and whether they match. When you use a multi-line pattern that is a regular expression that contains the "\n" character, or a TEC-style format that contains the '%n' token, the agent must break the monitored file into records of various sizes first. Then, the agent must check the records against the expressions in the format file. This procedure requires checking the regular expressions twice, so processing is slow. If you use the single-line pattern, it is assumed that each line of the file is a record and processing is much faster.

In some cases, it might be possible to ignore some of the lines and achieve better performance. For example, a single record from a RAS1 trace log is shown here:

```
(4D66DACB.0001-1:RAS1,400,"CTBLD")
```

```
+4D66DACB.0001 Component: ira
```

```
+4D66DACB.0001 Driver: agent_fac:15/4114877.7
```

```
+4D66DACB.0001 Timestamp: Feb 24 2011 13:18:54
```

```
+4D66DACB.0001 Target: sos510amdx6-d
```

In the example, if you are only interested in processing this line:

```
+4D66DACB.0001 Driver: agent_fac:15/4114877.7
```

you can write the following single-line pattern:

```
^\+.*Driver: agent_fac:([0-9\.\/]+)$
```

This single-line pattern processes the important value that you need without requiring the multi-line format. The other four lines of the logical record are treated as single-line records that do not match anything and are discarded.

Sort the expressions in the format file by frequency of occurrence in the monitoring log.

The agent checks each record that it reads from the log against the expressions in the format file until it finds a match. It starts with the final expression in the file, and searches upwards. When it finds a match, it stops searching. If the most commonly logged expression is listed last, then when that expression is logged, it is the only expression that is checked.

If you have 100 expressions in the format file, every time a log record matches the first one that is listed in the format file, the agent must check the other 99 expressions first, which slows down the processing. When a record that is read from the log does not match any of the patterns in the format file, the agent must check it against all of the patterns before it knows that it does not match. This process is slow and costly.

Include as much constant data as possible in the regular expressions.

For example, if the following error is returned in the log:

```
Disk error on device: /dev/sda1 Disk error on device: /dev/sdb2 yyy
```

you can write this expression:

```
^Disk.*: .*$
```

This expression causes a match, but it forces the regex engine to consider more possibilities on other lines that might be similar but ultimately do not match, for example, if the colon is missing.

The following expression is more useful because it is more precise and it causes the regular expression engine to stop processing errors that do not match:

```

^Disk error on device: /dev/sd[a-b][0-9]$
```

Do not use subexpressions that you do not need.

Subexpressions that are shown in parentheses in the example that follows are used to inform the regex engine that you want to use a value that is returned in the matched data. These subexpressions cause extra processing and are not necessary if you do not use the value that is returned. For example, when the following error is returned in the log:

```
write failure in writing to client 9.27.135.191. Error Broken pipe
```

if you include the following regular expression in the format file, the error message is captured at the end; but, if you do not use the returned value, performance is affected negatively:

```
REGEX
WriteFailure

^write failure in writing to client (.*)\. Error
(.*)$

ClientAddr $1
CustomSlot1

END
```

Use parentheses in expressions for grouping purposes.

You can use the ? operator to inform the regex engine not to capture the value that is returned. Therefore, you can use the ? operator to group only values that are returned. This grouping has a positive impact on performance. For example, if the following log data is returned:

```
Login succeeded on the first attempt for user Bob.
Login succeeded on the third attempt for user Joe.
```

To match both of the values that are returned, you must consider the first or third login attempt. If you do not care which specific login attempt succeeded or which specific user succeeded, you can include this expression to group the returned values:

```
REGEX
LoginSuceeded

^login succeeded on the (?:[a-z]+) attempt for user ([A-Z][a-z]*)\.$

UserName $1
CustomSlot1
END
```

If possible, do not use the OR (|) operator in expressions.

The | operator is expensive to process. The | operator causes the regex engine to complete a backup and to try to match values that did not match initially. This procedure is much more inefficient than having two separate expressions. For example, if you have the following expression:

```
REGEX DiskError
^.*disk error.*4|^.*disk failure.*4
END
```

it is much more efficient to use these two expressions:

```
REGEX DiskError
^.*disk error.*4
END

REGEX DiskError
^.*disk failure.*4
END
```

These expressions return the same results.

Important: These expressions violate the guideline to use as much constant data as is possible and demonstrate only the issues with the | operator.

Do not use ambiguous expressions.

Ambiguous expressions force the regex engine to back up and look for different ways to match an expression. For more information, see Performance Tips.

Ambiguous expressions might occur as a result of an expression that is included to break up a long record into many subexpressions. In this degenerate version of this problem, the expression has a space between the two (.*):

```
(.*) (.*)
```

In this example of the degenerate version, the regex engine looks for two strings of any expressions that are separated by a space. However, * also matches a space so the regex engine might assign the first space that it comes to initially to the first (.*). If it reaches the end of the input record without finding another space, it must back up and try again by using the space as the literal space called for in the expression.

To improve performance, use specific expressions only. You can use the Regex Pal tool to check whether the format file that you define matches the monitoring log. For more information, see Regex Pal.

Guidelines to limit the log file events that are reported.

The following guidelines limit the log file events that might cause the OS agents or the Cloud APM server to perform poorly:

Write specific formats in the `.fmt` file.
Write formats in the `.fmt` file that are specific and return relevant records. For example, you can generate an event for a specific error, such as the lines that begin with `Error:` and ignore the lines that begin with `Warning:`

```
Error: disk failure
Error: out of memory
WARNING: incorrect login
```

Do not turn on the Unmatchlog setting in the `.conf` file.
Ensure that you do not turn on the **Unmatchlog** setting in the `.conf` file because this setting logs all the unmatched files and overloads your file system.

Specify the `*DISCARD*` event class in the `.fmt` file.
Try to limit the CPU usage of the agent by specifying the predefined `*DISCARD*` event class in the `.fmt` file to discard data intentionally. When you use the `*DISCARD*` event class, events are not created for log records that match the pattern in the `.fmt` file. For example:

```
REGEX *DISCARD*
```

Turn on duplicate event detection over a longer time period.
You can turn on duplicate event detection by using the following keys in the `.conf` file:

- DupDetectionKeyAttributes
- EventSummaryInterval
- EventFloodThreshold

In this example, the duplicated lines are recognized by the *msg* and *CustomSlot1* values:

```
DupDetectionKeyAttributes=msg,CustomSlot1
EventSummaryInterval=300
EventFloodThreshold=send_first
```

If you have numerous duplicate events, apply the *send_first* or *send_none* threshold values to the events. For more information, see “Detecting and filtering duplicate events” on page 997.

Write specific threshold conditions.
Write specific threshold conditions that limit the set of rows that match the threshold. For example, the following threshold formula causes the threshold to fire only when an event of the FileSystemUsage event class has a value greater than or equal to 95 in CustomInteger1:

```
( Class == 'FileSystemUsage' AND CustomInteger1 >= 95)
```

Provide the correct set of `.conf` and `.fmt` files for the agent.
Ensure that you provide the correct set of `.conf` and `.fmt` files for the agent. For example, if you are configuring log file monitoring for the Windows OS agent, ensure that you configure the `.conf` and `.fmt` files that you created specifically for the Windows OS agent.

Query the MongoDB alarms database to determine the number of open events or event rate.

- Complete the following steps to query the MongoDB alarms database to determine the number of open events or event rate:

  1. Create an `event-query.js` file with a MongoDB query for the alarms database, for example:

     – This query counts all open and closed events with the following threshold name:

     ```
     UDB_DB_Pool_Hit_Rat_Pct_Crit_2 db.alarms.count
     ```

     ```
     ({"threshold_name" : "UDB_DB_Pool_Hit_Rat_Pct_Crit_2"})
     ```

- This query counts open and closed events in the MongoDB:

  ```
  db.alarms.count()
  ```

2. Run this command to get the results to the query in the `event-query.js` file: **/opt/ibm/mongodb/bin/mongo 127.0.0.1:27000/alarm -u user -p mongoUsrpasswd@08 <event-count.js**.

- Limit the amount of CPU that you specify for log monitoring. For more information, see "Log file monitoring environment variables" on page 630.

Availability Monitoring

With IBM Cloud Availability Monitoring, you can create, edit, view, and delete synthetic tests that mimic end-user behavior at your web applications.

The Availability Monitoring dashboard displays availability and response time information for monitored applications, URLs, and REST APIs. Use the dashboard to monitor alerts and activities that are associated with your application, URL, or REST API in different locations by using graphs, breakdown tables, and map views.

Availability Monitoring is available to users of the IBM Cloud Application Performance Management, Advanced on Cloud offering with the Availability Monitoring add-on.

About Availability Monitoring

Use Availability Monitoring to create synthetic tests that monitor the availability and performance of your web applications from different public and private locations around the clock.

Create tests with Availability Monitoring. Configure your tests to run at defined intervals and chosen locations. Download and deploy your own custom points of presence (PoPs) on local or private servers. Run tests from 15 public PoPs at the following locations:

Asia
Chennai, Hong Kong, Singapore, and Tokyo

Australia
Melbourne

Europe
Amsterdam, Frankfurt, London, and Paris.

Central America
Mexico

North America
Dallas, San Jose, Toronto, and Washington D.C.

South America
Sao Paolo

When you have created and configured your tests, you can then view availability and performance data for your applications on the Availability Monitoring dashboard.

Availability Monitoring has the following key features:

Get started in less than 5 minutes
Create single action tests easily to monitor your web application's performance and availability within minutes.

Maximize uptime and user satisfaction
Monitor the uptime and response time of your applications frequently from several geographical locations. Run synthetic tests to measure the performance of website loading and API calls. Monitor Selenium scripts that you use to mimic user flows from different locations.

Be proactive
Receive notifications to alert you to issues before they impact users. You can use the integrated Alert Notification service to create alert policies that reduce alert noise.

Identify reasons for failure accurately and quickly

Waterfall analysis helps you to identify the exact step when a failure occurred and the reason for the failure; for example, broken links, oversized images, slow lookups, or external requests. Screen captures are automatically created to help you to diagnose browser failures and historical performance issues. Download reports of monthly, weekly, and daily availability and response time averages for your tests.

To work in Availability Monitoring, you must be a member of a role that has view permission for the application that you wish to monitor. For more information, see "Roles and permissions" on page 982.

Accessing Availability Monitoring

The Availability Monitoring **Status Overview** tab for your application displays summary information about the availability and status of your tests. You can access the Availability Monitoring dashboard from the Availability Monitoring **Status Overview** tab.

About this task

Access the Availability Monitoring summary page by clicking an eligible application in the **All My Applications** pane on the Application Performance Dashboard. From the summary page, you can add tests for your application, view existing tests for your application, and view the Availability Monitoring dashboard.

Procedure

To access Availability Monitoring complete the following steps:

1. Click the **Performance** icon; then, click **Application Performance Dashboard**.
2. In the **All My Applications** pane, click an application that you want to monitor; then, click **Availability Monitoring** in the **Groups** pane.

   If no applications are listed, you must create an application. Ensure that you select **Custom application** as your **Template**. For more information, see "Managing applications" on page 1073.

   The Availability Monitoring **Status Overview** tab displays three gauges that show Average Test Availability in the last 24 hours, Current Test Status of all your tests, and Service Usage of your allocation for your current plan.

   You can configure how Availability Monitoring calculates Average Test Availability by selecting tests for inclusion in the **Availability Calculation**. Click the arrow icon ⌄ on the Average Test Availability gauge to view your test cards; then, click **Availability Calculation**. Click a card to add or remove it from the calculation. Excluded test cards are faded. When you are finished, click **I'm Done**. The Average Test Availability gauge is refreshed.

   You can view the status of all your tests by clicking the arrow icon ⌄ on the Current Test Status gauge. Test cards for your tests are displayed. Click a card to access the **Breakdown** dashboard for that particular test.
3. Click **See Monitoring Details** to access the Availability Monitoring dashboard and view data for all tests for your application. Click **View All Tests** to view and edit your tests in the **Synthetic Tests** pane. Click **Add New Test** to create a test.

   Note: The first time that you run Availability Monitoring, you must add a test before you can view any data in the Availability Monitoring dashboard.

Creating and configuring tests

Create and configure tests that report the availability and performance of your web applications.

Create and configure tests to monitor the availability and response time of your applications frequently from several geographical locations. Run synthetic tests to measure the performance of website loading and REST API calls. Create scripted behavior tests to run and monitor Selenium scripts that mimic user flows from different locations.

Creating a REST API test

Create a REST API test to test the response time and availability of your web application by using the following HTTP methods: GET, POST, PUT, and DELETE.

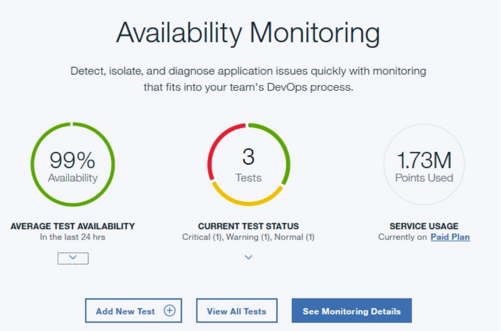
About this task

Use REST API tests to monitor the availability and performance of your web application and other URLs in response to REST calls.

Procedure

To create a REST API test, complete the following steps.

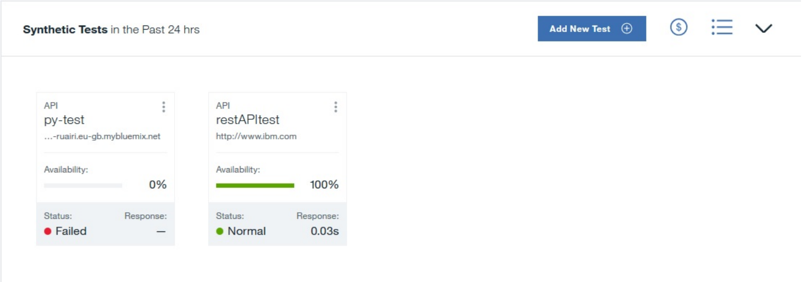
1. If you are viewing the Availability Monitoring summary page for your application, click **Add New Test**.

   Availability Monitoring

   Detect, isolate, and diagnose application issues quickly with monitoring that fits into your team's DevOps process.

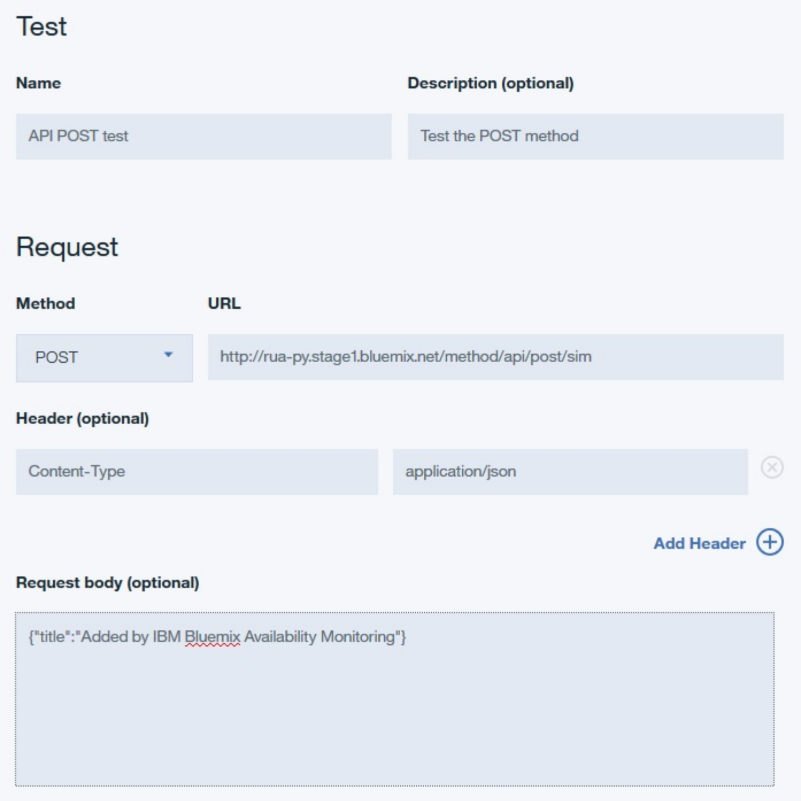
   If you are viewing the Availability Monitoring dashboard, click **Add New Test** on the **Synthetic Tests** pane.

2. Click **Single Action** on the **Monitoring Setup** page; then, click **REST API** on the **Single Action** page.
3. Enter a meaningful name for your test in the **Name** field. Add a description of the purpose of your test to the **Description** field.
4. In the **Request** section, select the type of method from the **Method** list and enter a **URL** that you want to test with this method.

   You can choose **GET**, **PUT**, **POST**, or **DELETE**. If you choose the **PUT** or **POST** method, you can enter body content to test in the **Request body (optional)** field.

   For example, the following REST API test uses the POST method to request that your web app accepts data in addition to testing the availability and performance of that web applications.



Test

Name: API POST test

Description (optional): Test the POST method

Request

Method: POST

URL: http://rua-py.stage1.bluemix.net/method/api/post/sim

Header (optional): Content-Type | application/json

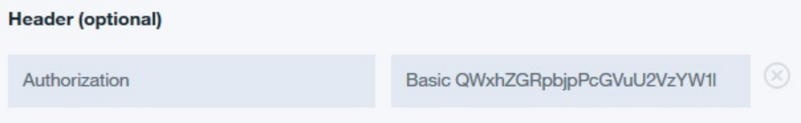
Request body (optional):

```
{"title":"Added by IBM Bluemix Availability Monitoring"}
```

5. Optional: Configure your test to include a particular header and value. Enter a header name and header value in the **Header** fields.

   If the web app that you want to test requires a user login and password, enter "`Authorization`" into the **Header name** field. Enter the word "Basic", a space character, and the base64 encoded value of your *username:password* into the **Header value** field.

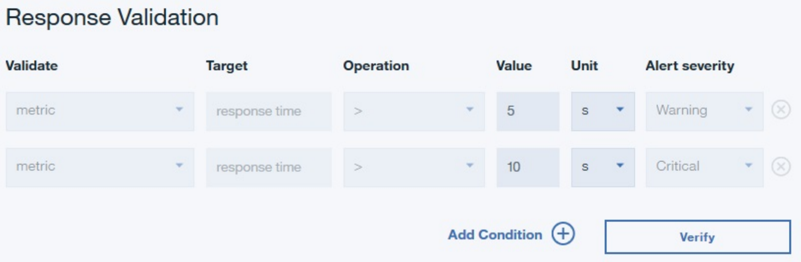
   For example, if your user name is *Aladdin* and your password is *OpenSesame*, then enter the word "Basic", a space character, and the base64 encoded value for *Aladdin:OpenSesame* into the **Header value** field.



Header (optional): Authorization | Basic QWxhZGRpbjpPcGVuU2VzYW1l

6. Configure the warning and critical alert thresholds for your test in the **Response Validation** section. Edit the **Value** and **Unit** for each row.

   Response times that exceed your warning and critical thresholds trigger alerts.



Response Validation

Validate	Target	Operation	Value	Unit	Alert severity
metric	response time	>	5	s	Warning
metric	response time	>	10	s	Critical

7. Optional: Click **Add Condition** to define and add customized response validation conditions.

   Customized response validation conditions are evaluated in aggregate to generate an alert. You can define and add up to six customized conditions for your test.

   Important:

In Availability Monitoring, each test can generate up to a total of three alerts. Your test reports the alert with the highest severity until all conditions that cause alerts are resolved. For more information, see "Alert generation in Availability Monitoring" on page 1037.

You can validate the following data:

Header response code
Select **Header response code** to test for one or for a range of HTTP response codes.

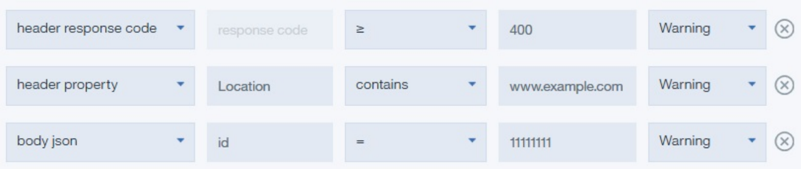
Header property
Select **Header property** to test for a particular HTTP header field property and value.

Body JSON
Select **Body JSON** to test for a particular property from a JSON body.

For each condition, enter a property to test for in the **Target** field, and a value to test for in the **Value** field. Select an operator from the **Operation** drop-down menu. Finally, choose an **Alert severity** of Warning or Critical for your condition.

Important:

Numerical values that you enter in the **Value** field are treated as numbers and not strings by default. To enter a **Value** for a response validation condition, use quotation marks "" to distinguish between a string and a number. For example, to test for the string 123, enter `"123"` in the **Value** field. To check for the number 400, enter `400` without any quotation marks.



header response code	response code	≥	400	Warning
header property	Location	contains	www.example.com	Warning
body json	id	=	11111111	Warning

8. Click **Verify** to create your REST API test and to determine whether your test request is valid.

   Availability Monitoring determines the test validity by using the selected HTTP method and any request headers that you defined for the test. No response validation takes place during test verification.

   Your validated test is displayed in the **Verified Items** table. You can add more URLs by repeating steps 3 - 8.

9. To configure your test settings, click **Next**.

   A summary of the test configuration is displayed. The following message is displayed for the default settings:

   ```
   Test will occur: Every 15 minutes from 3 public locations and no private
   locations simultaneously to determine if test exceeds the specified
   threshold.
   ```

10. In the **Settings** pane, click **Edit** to display the current settings for your test.
    You can update the following settings:

    - **Interval** defines how often the test runs.
    - **Testing frequency** determines whether your test runs from all locations simultaneously or from a different location at each interval. Select **Simultaneous** to run your test from all locations simultaneously, or select **Staggered** to run your test from a different selected location at each interval.
    - **Locations** determines the locations where your test runs

11. Select your locations from the list of **Public Locations**. To select a private location to run your test from, you must first install and configure a private PoP on the machine that you want to run your test from. For more information, see "Installing and configuring private PoP locations" on page 1033.
12. Click **Save** to finish configuring your test; then, click **Finish**.

The Availability Monitoring dashboard is displayed. After a minute, the dashboard displays information and data for your new test.

Creating a webpage test

Create a webpage test to test the availability of your web application and to monitor how long that page takes to open.

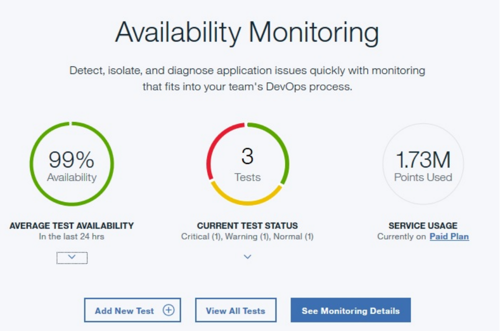
About this task

Webpage tests report the response time for loading the URL of your web application. Create a webpage test to monitor the availability and response time of your web application.

Procedure

To create a webpage test, complete the following steps.

1. If you are viewing the Availability Monitoring summary page, click **Add New Test**.



Availability Monitoring

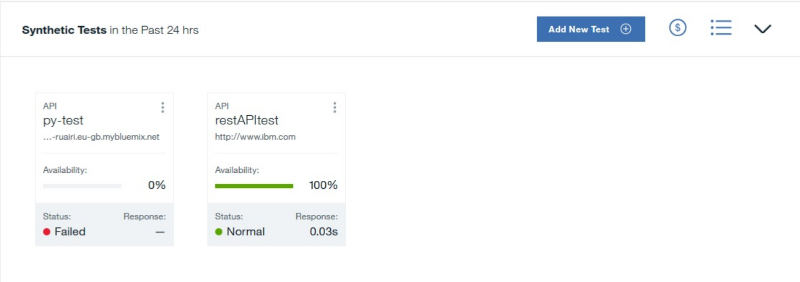
Detect, isolate, and diagnose application issues quickly with monitoring that fits into your team's DevOps process.

AVERAGE TEST AVAILABILITY
In the last 24 hrs

CURRENT TEST STATUS
Critical (1), Warning (1), Normal (1)

SERVICE USAGE
Currently on Paid Plan

If you are viewing the Availability Monitoring dashboard, click **Add New Test** on the **Synthetic Tests** pane.



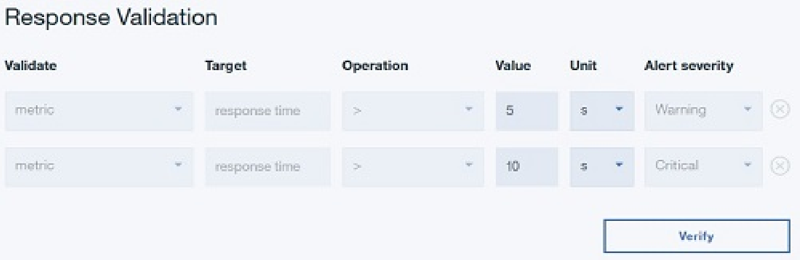
Synthetic Tests in the Past 24 hrs

API
py-test
...-ruairi.eu-gb.mybluemix.net
Availability: 0%
Status: Failed
Response: —

API
restAPItest
http://www.ibm.com
Availability: 100%
Status: Normal
Response: 0.03s

2. Click **Single Action** on the **Monitoring Setup** page; then, click **Webpage** on the **Single Action** page.
3. Enter a meaningful name for your test in the **Name** field. Add a description of the purpose of your test to the **Description** field.
4. Enter the **URL** of the web application that you want to test.
5. Configure the warning and critical alert thresholds for your test in the **Response Validation** section. Edit the **Value** and **Unit** for each row.

   Response times that exceed your warning and critical thresholds trigger alerts.



Response Validation

Validate	Target	Operation	Value	Unit	Alert severity
metric	response time	>	5	s	Warning
metric	response time	>	10	s	Critical

6. Use the **Blacklist** and **Whitelist** to specify which URLs and domains to send requests to and which URLs and domains contribute to the metrics and status of your application tests. Add URLs and domains that you want to include or block to the **Whitelist** and **Blacklist**.

   For more information, see "Blocking and filtering with the whitelist and blacklist" on page 1032.
7. Click **Verify** to create your webpage test and to determine whether your test request is valid.

   Availability Monitoring determines the test validity by sending a GET request to your test URL. No response validation takes place during test verification.

   Your validated test is displayed in the **Verified Items** table. You can add more URLs by repeating steps 3 - 7.
8. To configure your test settings, click **Next**.

   A summary of the test configuration is displayed. The following message is displayed for the default settings:

   ```
   Test will occur: Every 15 minutes from 3 public locations and no private
   locations simultaneously to determine if test exceeds the specified
   threshold.
   ```

   The estimated usage and estimated number of tests per month are displayed based on your current test configuration.
9. In the **Settings** pane, click **Edit** to display the current settings for your test.

   You can update the following settings:

   - **Interval** defines how often the test runs.
   - **Testing frequency** determines whether your test runs from all locations simultaneously or from a different location at each interval. Select **Simultaneous** to run your test from all locations simultaneously, or select **Staggered** to run your test from a different selected location at each interval.
   - **Locations** determines the locations where your test runs

   Select locations from the list of **Public Locations**. To select a private location to run your test from, you must first install and configure a private PoP on the machine that you want to run your test from. For more information, see "Installing and configuring private PoP locations" on page 1033.

   Click **Save** to finish configuring your test.
10. Click **Finish**.

    The Availability Monitoring dashboard is displayed. After a minute, the dashboard displays information and data for your new test.

Creating a script test from an uploaded script

Upload a Selenium script to create a script test that tests the availability and performance of your web application in response to simulated user behavior.

Before you begin

In order to create a script test, you must first create a Selenium script. For more information about creating Selenium scripts, see Recording synthetic scripts.

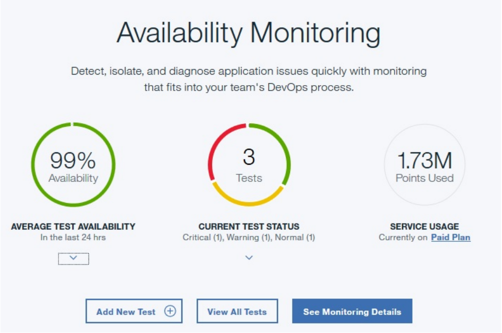
About this task

Create a script test to monitor a Selenium script that simulates your users' interactions with your web app. If you create a Selenium script that mimics a user who is logging in to your application, you can then run a script test periodically to test the performance of your app in response to the simulated user actions.

Procedure

To create a script test, complete the following steps.

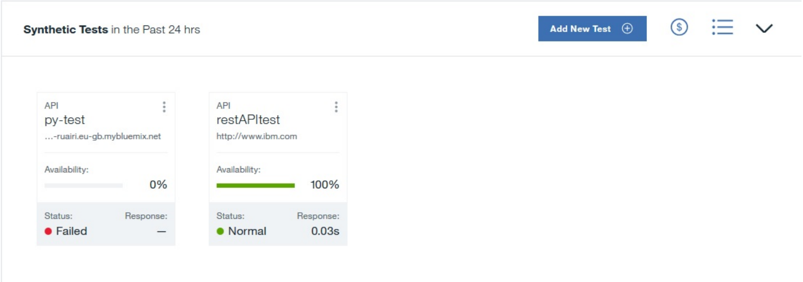
1. If you are viewing the Availability Monitoring summary page, click **Add New Test**.

   Availability Monitoring

   Detect, isolate, and diagnose application issues quickly with monitoring that fits into your team's DevOps process.

   If you are viewing the Availability Monitoring dashboard, click **Add New Test** on the **Synthetic Tests** pane.

2. Click **Scripted Behavior** on the **Monitoring Setup** page. The **Scripted Behavior Setup** page is displayed. Click **Upload File**.

   If you return to edit this test at a later point, you can download the uploaded script file. Click the **Download** icon to download your script.

3. Enter a meaningful name for your test in the **Name** field. Add a description of the purpose of your test to the **Description** field.
4. Click **Browse** to find and upload a script file.
5. Use the **Blacklist** and **Whitelist** to specify which URLs and domains to send requests to and which URLs and domains contribute to the metrics and status of your application tests. Add URLs and domains that you want to include or block to the **Whitelist** and **Blacklist**.

   For more information, see "Blocking and filtering with the whitelist and blacklist" on page 1032.

6. To configure your test settings, click **Next**.

   A summary of the test configuration is displayed. For example, the following message is displayed for the default settings:

   ```
   Test will occur: Every 15 minutes from 3 public locations and no private
   locations simultaneously to determine if test exceeds the specified
   threshold.
   ```

   The estimated usage and estimated number of tests per month are displayed based on your current test configuration.

7. In the **Settings** pane, click **Edit** to display the current settings for your test.

   You can update the following settings:

   - **Interval** defines how often the test runs.
   - **Testing frequency** determines whether your test runs from all locations simultaneously or from a different location at each interval. Select **Simultaneous** to run your test from all locations simultaneously, or select **Staggered** to run your test from a different selected location at each interval.
   - **Critical Threshold** defines the response time for critical alerts for the test.
   - **Warning Threshold** defines the response time for warning alerts for the test.

- **Locations** determines the locations where your test runs.

  Select locations from the list of **Public Locations** which are displayed by default. To select a private location to run your test from, you must first install and configure a private PoP on the machine that you want to run your test from. For more information, see "Installing and configuring private PoP locations" on page 1033.

  If required, you can enter the values for variables that are defined in your test script. For example, if your script requires a user name and password to connect to a website, you can enter the values for these variables. You can set different values for your variables in different locations in the **Script Variables** table.

  Click **Save** to finish configuring your test.

8. Click **Finish**.

   The Availability Monitoring dashboard is displayed. After a minute, the dashboard displays information and data for your new test.

Blocking and filtering with the whitelist and blacklist
Use the whitelist and blacklist to determine which resources to send requests to and which resources contribute to the metrics and status of your application tests. Whitelists and blacklists are only available for webpage and scripted behavior tests.

The **Whitelist** and **Blacklist** fields define the resources that your test can or cannot access and the resources that contribute to the metrics and status of your tests. The Whitelist and Blacklist control which dependencies and resources contribute to the response times of your tested web applications, such as third-party metrics. You can configure your Whitelist and Blacklist when you create a webpage or scripted behavior test.

You can use the **Whitelist** to define allowed domains and URLs; then, use the **Blacklist** to block specific elements of your allowed locations.

Syntax

Use commas (,) to separate items in the Blacklist and Whitelist. Use the wildcard symbol (*) to filter elements of each URL or domain.

Whitelist

Add URLs, schemes, or domains that you want to include in request and metric calculations to the Whitelist field. You can list up to 10 items in your Whitelist. Each item length cannot exceed 200 characters. All domains, schemes, and URLs that do not match the items on your Whitelist are blocked.

For example: `ibm.com, *developerworks*, *.s81c.com/*, https://www.ibm.com*, https://*`

Note: If your Whitelist URL filter includes `http://` or `https://`, you must include the wildcard symbol (*) directly after the URL, for example, `https://www.ibm.com*`.

Blacklist

Add URLs, schemes, or domains that you want to block from requests and metric calculations to the Blacklist field. You can list up to 20 items in your Blacklist. Each item length cannot exceed 200 characters.

For example: `*.profile.*.cloudfront.net/*.png, http://*`

Note: If your Blacklist URL filter includes `http://` or `https://`, you must include the wildcard symbol (*) directly after the URL, for example, `https://www.ibm.com*`.

Filtering and blocking behavior

Tests can have both a Whitelist and a Blacklist. When determining which locations are allowed or blocked, the Blacklist always overrides the Whitelist. The following table displays filtering and blocking behavior for all scenarios involving the Whitelist and Blacklist.

Table 255. Filtering and blocking behavior for the Whitelist and Blacklist

Blacklist	Whitelist	Behavior	Reason
Empty	Empty	Allow access	No filtering rules entered.
Empty	URL does not match list entry	Block access	URL is not in the whitelist.
Empty	URL matches list entry	Allow access	URL is in the whitelist. No blacklist entry to block access.
URL does not match list entry	Empty	Allow access	URL is not in the blacklist. No whitelist entry to prevent access to URLs that are not on the whitelist.
URL matches list entry	Empty	Block access	URL is in the blacklist.
URL does not match list entry	URL does not match list entry	Block access	URL is not in the whitelist.
URL does not match list entry	URL matches list entry	Allow access	URL is in the whitelist. URL is not in the blacklist.
URL matches list entry	URL does not match list entry	Block access	URL is not in the whitelist. URL is in the blacklist.
URL matches list entry	URL matches list entry	Block access	URL is in the blacklist. The blacklist entry overrides the whitelist entry.

Installing and configuring private PoP locations

Download and install a private PoP to a local machine; then, configure your private PoP for use as a location for your tests in Availability Monitoring.

Before you begin

To install a private PoP, the installation location for your private PoP must fulfill the following requirements:

- Linux is installed with a kernel version 3.1.0 or higher.
- Docker service version 1.7.1 or higher is installed and started.
- Available disk space is 4 GB or more.
- Available memory is 2 GB or more.
- CPU cores:
  - If you need only REST API playbacks in your Private PoP, have at least 2 available CPU cores.
  - If you want to run web page playbacks and script playbacks in your private PoP, have 1 CPU core for every 1 or 2 tests to run every minute.

- Check your private PoP CPU and memory usage before and after adding new tests, and after applying private PoP software updates that include updated versions of Firefox or the Selenium IDE because later versions might have higher system requirements.

Best practice for determining when to add CPU cores is to run your most demanding process on your private PoP host to get the CPU usage and memory usage: If the total CPU usage is greater than 70% and the top CPU usage process is Firefox, add CPU cores until the total CPU usage is below 50%; if free memory on your private PoP host is below 500 MB, increase memory.

If you don't have more hardware resources, but want your private PoP to run without exception, take these steps to reduce the Firefox parallel running instance count (causes your tests to run with longer intervals than what is configured in the UI because your hardware resource can't run many tests):

1. Edit the `start-pop.sh` script to add the **`MAX_TASKPOOL_SIZE`** environment variable, entering the available CPU cores in the private PoP host as the value, then run **`stop-pop.sh`** followed by **`start-pop.sh`**.
2. Set the tests to a longer interval in the UI.

You must have user access to the command line interface (CLI) of the machine where you want to install your private PoP. You must also have the necessary user permissions to add packages to Docker.

Important:

- Ensure that the system time of the machine where you want to run a private PoP is and remains synchronized with standard time. Otherwise, your test instances display incorrect timestamps on the Availability Monitoring dashboard.
- The Availability Monitoring private PoP is fully supported for the following platforms: Red Hat Enterprise Linux 7.4 and CentOS Linux 7.4.

About this task

In addition to public locations, you can deploy private Points of Presence (PoPs) when you create or edit a test in Availability Monitoring. Use private PoPs to test applications that are located behind your company's firewall, such as applications with greater privacy or security requirements. You can register a maximum of 50 private locations in Availability Monitoring. Download the precheck script and private PoP package; then, save the script and package on the machine where you want to run the private PoP.

Procedure

1. Create a test or edit an existing test.

   To create a test, click **Add New Test** on the **Synthetic Tests** pane. To edit a test, click **Actions** ⋮ ; then, click **Edit**. If you are creating a test, configure and verify your test. In the **Settings** section, click **Edit**.

   For more information, see "Creating a REST API test" on page 1026, "Creating a webpage test" on page 1029, and "Creating a script test from an uploaded script " on page 1030.
2. Click **Edit** in the **Settings** section to display the **Locations** section; then, click **Private Locations**. If you are editing a previous test, click **Private Locations** in the **Locations** section.

   If you previously installed one or more private PoPs, a list of all installed private PoPs is displayed. If no private PoPs are installed and configured, Availability Monitoring can guide you to set up a private PoP.
3. Click **Download pre-check** and save the precheck script to a machine that you want to run tests from.

   Important: You must extract and run scripts from the command-line interface (CLI) to install a private PoP. The private PoP scripts and package can be installed on a different machine, and accessed through the CLI for that machine. Do not close or refresh Availability Monitoring in your browser while you are working with private PoP scripts, or you lose any unsaved test settings.

Open a CLI for the machine that you want to locate the private PoP. From the CLI, navigate to the location where you saved the precheck script; then, run the precheck script as follows:

```
./precheck.sh
```

Ensure that you have the necessary permissions to execute shell scripts on your machine.

The precheck script displays the result of the check. If your environment fails the check, update your machine to meet the requirements that are displayed.

4. Return to Availability Monitoring, click **Download package**, and save the package. Move the package to the machine that you want to run tests from. From the CLI, navigate to the location where you saved the download package; then, run the following command to extract the package:

   ```
   tar -xvf Availability_Monitoring_PoP.tar
   ```

   Where *Availability_Monitoring_PoP.tar* is the name of the `.tar` file that contains the private PoP package that you downloaded.

5. Configure your private PoP. From the CLI, run the following script:

   ```
   ./config-pop.sh
   ```

   When prompted, enter the following information for your private PoP:

   - PoP name
   - Country location
   - City location
   - PoP latitude
   - PoP longitude
   - PoP description

6. If any of your REST API tests connect to a server that uses a self-signed certificate or is not signed by a well known CA certificate provider, put any trusted CA certificates that are in `.pem` file format into the `keyfiles` directory.

   Note:

   - Any changes to the `.pem` certificate files require the private PoP to be restarted.
   - The server being tested should send all but the root CA certificate during the TLS handshake; if this does not happen, correct the server configuration if possible. Otherwise, you can add any missing certificates to the `keyfiles` directory as described in this step. However, the PoP test (or tests) might not reflect the experience of other clients.

7. To configure your private PoP to use a proxy server when you run web page tests or scripted behavior tests, enter one of the following options:

   Important: REST API tests that run from your private PoP location with a manual or automatic proxy configuration do not use that proxy. Only web page and scripted behavior tests can use a proxy server to run from private PoP locations.

   no

   Enter no to configure your private PoP to not use a proxy when you run tests.

   manual

   Enter `manual` to configure manually a proxy IP address and port number for your private PoP proxy to use when you run tests. The script requests the proxy server IP address and port number in the following form: *`ip address`*`:`*`port number`*. You can also create a no proxy list to block domain elements, host names, or IPv4 address elements. When prompted, enter one or more domain elements or IPv4 address elements. Separate each list item with a blank space or a comma (","). The wildcard operator (*) is not supported.

   - To block a domain and any sub domains, enter a domain suffix, starting with a dot, for example: `.example.org, example.org`.

- To block a network, enter an IP address with a CIDR suffix to identify an IP address range to block, for example: `10.0.0.0/8`.

pac
: Enter `pac` to configure your private PoP to use an automatic proxy configuration URL. When prompted by the script, enter the automatic proxy configuration URL.

Your private PoP settings are saved in the `pop.properties` file.

8. Start your private PoP. From the CLI, run the following script:

   ```
   ./start-pop.sh
   ```

   When your private PoP is running, it can be found by Availability Monitoring.
9. Return to Availability Monitoring and click ↻ **Refresh Locations** to find and display your new private PoP.

   Your private PoP is listed in a table.
10. To choose your private PoP as a location for your test, select the check box for the table row that contains a private PoP. To delete a private PoP, complete the following steps:
    a) From the CLI, run the **`./stop-pop.sh`** script on the machine where your private PoP is located.

    b) Return to Availability Monitoring and click 🗑 **Delete** on the table row that contains the private PoP that you want to delete.
11. Repeat steps 3 - 10 to add further private PoPs to different machines for selection as locations in Availability Monitoring. Click **Finish** to save and start your test.

    The Availability Monitoring dashboard is displayed. After approximately 1 minute, the dashboard displays information and data for your new test.
12. Optional: To upgrade an existing private PoP, complete the following steps:
    a) Download the new private PoP package to a new folder; then, use the **`tar -xvf`** command to unpackage the new PoP in that folder.

    b) From the CLI, change directory to the folder where your old private PoP is located. Run the following script to stop the old private PoP:

    ```
    ./stop-pop.sh
    ```

    c) In the directory where your old private PoP is located, backup your existing `system.properties` and `pop.properties` files.

    Important: The `system.properties` file contains critical information that allows your private PoP to connect to the Cloud APM server. The `pop.properties` file contains the configuration data for your private PoP. If you want to retain this configuration, ensure that you preserve the `pop.properties` and `system.properties` files for your old private PoP before you upgrade your private PoP.

    d) Copy all files from your new private PoP folder except `pop.properties` and `system.properties` and replace the files where your old private PoP is located.

    e) If you need to reconfigure your upgraded private PoP, run the following script from the CLI:

    ```
    ./config-pop.sh
    ```

    f) Run **`./start-pop.sh`** from the CLI to start your upgraded private PoP.

    g) Return to Availability Monitoring and click ↻ **Refresh Locations** to find and display your upgraded private PoP.

Alert generation in Availability Monitoring

In Availability Monitoring, tests can generate up to a total of three alerts. Your test reports the alert with the highest severity until the condition causing the alert is resolved.

A separate alert is raised for three different situations:

- When the response time of your web application or URL exceeds the warning or critical thresholds set for your test. Every test measures response time by default and raises an alert based on the warning and critical thresholds for that test.
- When your test returns a HTTP response code that indicates that your web application or URL is unavailable due to a client or server error. Every test checks for the response code by default, to determine whether the test is successful or fails.
- When your test determines that one or more customized conditions are satisfied, an alarm is raised with the highest severity as defined by one or more of your customized conditions. Availability Monitoring considers all customized conditions in aggregate when determining whether an alarm is raised. This alarm remains until your test determines that all of your customized conditions no longer generate any warning or critical alerts.

When more than one alert is raised, Availability Monitoring will report the alert with the highest severity as long as any alerts are present.

For example, if you add a customized condition that raises a critical alert and another customized condition that raises a warning alert, a critical alert is generated by your test. This alert is visible on the Availability Monitoring dashboard. If the condition that causes a critical alert is no longer true, then the severity of your test alert changes to "warning". An alert remains until all conditions no longer cause an alert.

Availability Monitoring alerts are not displayed on the **Events** tab of the Cloud APM console. The alerts also do not affect the status of applications, components, and instances that are displayed in the navigator bar of the Cloud APM console.

Availability Monitoring alerts are automatically forwarded to Netcool/OMNIbus if you install the Integration Agent for Netcool/OMNIbus, or to Cloud Event Management if you configure integration with it, or to Alert Notification if it is included in your Cloud APM subscription.

Viewing app availability and performance on the Monitoring Dashboard

You can view the details of your application's availability and performance, along with alerts and any associated tests on the Availability Monitoring dashboard.

The Availability Monitoring dashboard is divided into the following panes:

- Application Summary
- Alert Frequency
- Synthetic Tests
- Response Time and Availability

Use the **Navigate to** drop-down menu to navigate quickly to any pane.

Use the guides to help you to learn about the features of Availability Monitoring. To open a guide, click the **Help** icon ?; then, click the guide that you want to view.

Video tutorial library

The Video tutorial library contains videos on how to create Availability Monitoring tests, create test scripts with Selenium IDE, and send alerts.

Welcome to Monitoring!

The Welcome to Monitoring guide highlights areas of the dashboard and explains each feature of Availability Monitoring.

You can access the **Breakdown** dashboard from the Application Summary pane, the Alert Frequency pane, the Synthetic Tests pane, or the Response Time and Availability pane. The **Breakdown** dashboard displays key statistical information for your test instances.

You can change the order of the panes to suit your needs. To move a pane, click the heading and drag the pane to a different position. To save these changes so that they persist after you log out, click **Save Layout**.

You can set the dashboard to automatically refresh every minute. Click the **Configure** icon ; then, click the **Refresh** sliding bar to select **1 min**. To refresh your page at any time, click **Refresh**.

Application Summary

The Application Summary pane displays an overview of alert status over the past 24 hours and current test status information.

The Application Summary pane displays the following information:

- **Current Status** displays the highest severity status of all your tests. The severity can be Normal, Warning, or Critical.
- **Alerts** displays the number of open alerts and divides them into warning and critical alerts.
- **Availability Report** allows you to download a `.csv` file report of the monthly, weekly, and daily availability and response time averages for the app. Click the **Report** icon to download the report.

Alert Frequency

The **Alert Frequency** panel contains a map that displays the most recent alerts. Alerts are grouped by location, and listed in the **Alerts** table.

Alert Frequency map

The **Alert Frequency** map displays at-a-glance information for all public and private points of presence (PoPs) for your tests.

Use the zoom function to enlarge any area of the map or restore it to its original size. Hover over each location to view the name of that location, and the number of warning and critical alerts at that location. You can filter the alerts that are displayed on the map by selecting **All**, **Open**, or **Closed** from the **Alerts** drop-down list.

PoP locations

PoP location icons (0) indicate the PoP locations for your tests. The color of each **PoP location** icon represents the severity of the most recent alert at each location: Normal (0), Warning (1), or Critical (1). An animated **PoP location** icon indicates that this location has the most alerts with the highest level of severity out of all locations for your test instances.

Add PoP locations to your selected test by hovering over an **Inactive PoP location** icon (+) and clicking **Test here**. The **Test Edit Mode** page is displayed for your selected test. You can select a test from the **Test** drop-down menu on the **Response Time** and **Availability** pane.

Private PoP locations are represented by **Private PoP location** icons (2)

Number of alerts

PoP location icons display the number of open, closed, or all alerts that are generated at each location. The **Critical**, **Warning**, and **Normal** icons ● 12 Critical ● 11 Warning ● 0 Normal display the number of alerts of each severity for your locations.

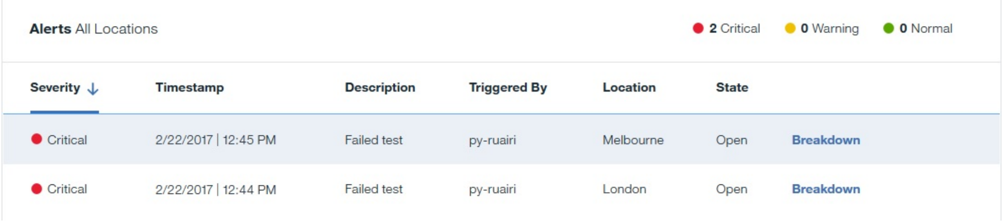
Failed tests

Locations where failed tests occur are indicated by a **PoP location** icon with a broken border 10.

Use the zoom function to enlarge any area of the map or restore it to its original size. Hover over each location to view the name of that location, and the number of warning and critical alerts at that location. You can filter the alerts that are displayed on the map by selecting **All**, **Open**, or **Closed** from the **Alerts** drop-down list.

Alerts table

The alerts for all locations are displayed in a table.



Alerts All Locations ● **2** Critical ● **0** Warning ● **0** Normal

Severity ↓	Timestamp	Description	Triggered By	Location	State	
● Critical	2/22/2017 | 12:45 PM	Failed test	py-ruairi	Melbourne	Open	Breakdown
● Critical	2/22/2017 | 12:44 PM	Failed test	py-ruairi	London	Open	Breakdown

The table displays the following information about your alerts:

- **Severity** describes the alert as critical or warning.
- **Timestamp** shows the time that the alert is created.
- **Description** summarizes the performance of your test instance.
- **Triggered By** shows the name of the test that triggered the alert.
- **Location** states where the problem occurred.
- **State** shows whether the alert is open or closed.

Viewing alert details

Each alert in the table contains a link to the **Breakdown** dashboard. Use the breakdown dashboard to help you to troubleshoot the issue that caused the alert.

Filtering alerts

To filter alerts for a particular location, click a **PoP location** icon on the map. To show alerts for all locations, click anywhere on the map that isn't a **PoP location** icon.

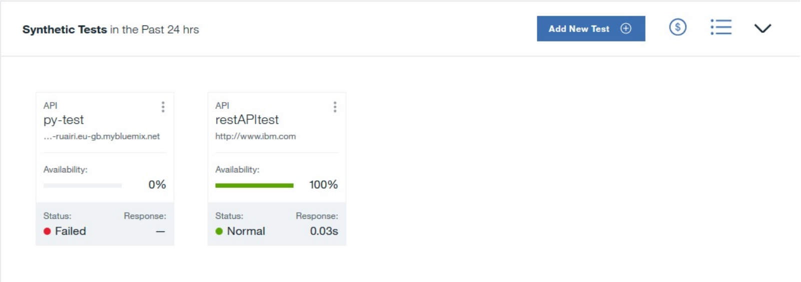
To filter the alerts in the table by severity, click the **Critical**, **Warning**, or **Normal** icon ● **12** Critical ● **11** Warning ● **0** Normal . To remove the filter and include alerts of each severity in the table, click the selected icon again.

Changing alert thresholds

Alerts are triggered by the thresholds that you specify when you create a test. In most cases, they are generated due to availability failures or slow response times. To change the threshold settings, click the **Actions** icon ⋮ on the test that generated the alert in the **Synthetic Tests** pane and click **Edit**.

Synthetic Tests

In the **Synthetic Tests** pane, you can create, edit, delete, and view *synthetic tests* that monitor the performance and availability of your applications. Tests are displayed in a list or card view in the **Synthetic Tests** pane.



Synthetic Tests in the Past 24 hrs

Add New Test

API py-test ...-ruairi.eu-gb.mybluemix.net	API restAPItest http://www.ibm.com
Availability: 0%	Availability: 100%
Status: ● Failed Response: —	Status: ● Normal Response: 0.03s

Each test card displays information about the test:

Availability
Displays the percentage availability of the test over the past 24 hours.

Status
Displays the current status of the test. The status can be Critical, Warning, Normal, Failed, Inactive, or Unknown.

Avg. Response
Displays the average response time of the test over the past 24 hours.

You can monitor three different types of test:

REST API
Reports the response time of a REST call. All HTTP request formats, such as GET, POST, PUT, and DELETE are supported.

Webpage
Reports the response time for loading the website at the URL that you enter.

Scripted behavior
Monitors Selenium scripts that you create to mimic a user's interactions with a website. For example, you can create a Selenium script that mimics a user who is logging in to your application. Run this script periodically to test the performance of your app in response to the user actions that are automated by the script. For more information about creating Selenium scripts, see "Recording synthetic scripts" on page 1007.

To add another test, click **Add New Test**.

To stop, start, delete, or edit a synthetic test, click the **Actions** icon ⋮ and click the action that you want. To view the **Breakdown** details of the test, click the test.

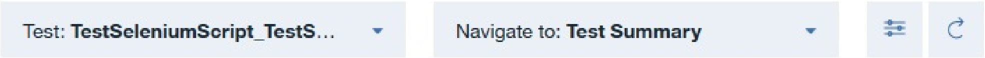
To view the specific usage of each synthetic test, click the **Cost** icon $. If you are subscribed to the paid plan, your usage is displayed in data points.

Breakdown

The **Breakdown** dashboard displays key statistical information for your tests. The dashboard also summarizes availability and response time information, historical trends, and test performance data over the previous 24 hours.

To view a detailed breakdown of a test, click the test in the **Synthetic Tests** pane. You can also open the **Breakdown** dashboard by clicking **Breakdown** in the **Alert** table in the Alert Frequency pane.

Use the **Test** drop-down menu to view breakdowns of different tests.. Use the **Navigate to** drop-down menu to navigate quickly to any pane.

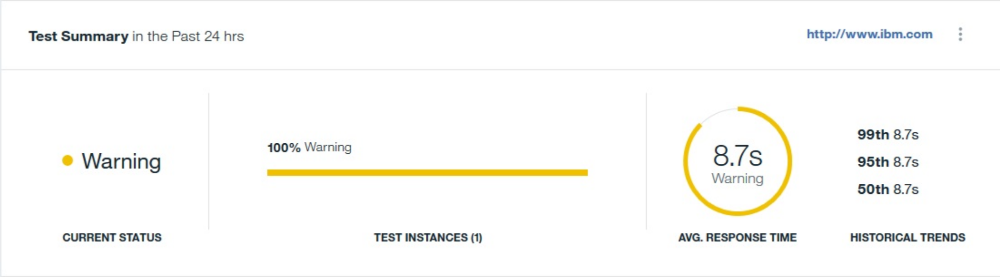


Test: **TestSeleniumScript_TestS...**

Navigate to: **Test Summary**

The Breakdown dashboard displays four panes.

Test Summary



Test Summary in the Past 24 hrs

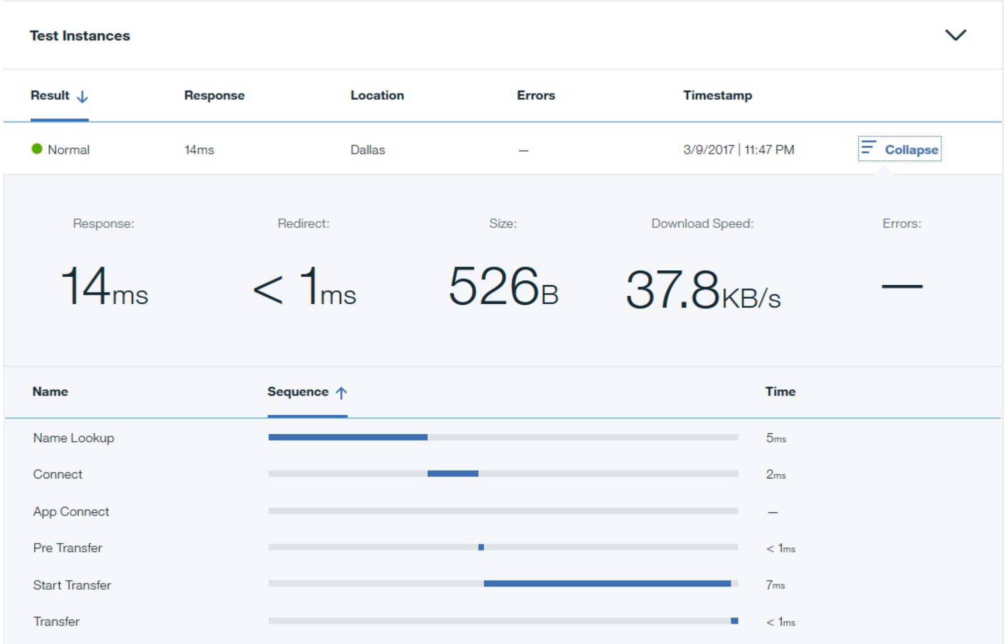
http://www.ibm.com

CURRENT STATUS	TEST INSTANCES (1)	AVG. RESPONSE TIME	HISTORICAL TRENDS
● Warning	**100%** Warning	8.7s Warning	**99th** 8.7s **95th** 8.7s **50th** 8.7s

The **Test Summary** pane displays the following test information for the past 24 hours:

- **Current Status** displays the test status.
- **Test Instances** displays a percentage breakdown of normal, warning, and critical test instances.
- **Avg. Response Time** displays the average response time of the test.
- **Historical Trends** displays the historical trends of your test performance for the 50th, 95th, and 99th percentiles in seconds or milliseconds.

Test Instances



Test Instances

Result ↓	Response	Location	Errors	Timestamp	
● Normal	14ms	Dallas	–	3/9/2017 | 11:47 PM	Collapse

Response:	Redirect:	Size:	Download Speed:	Errors:
14ms	< 1ms	526B	37.8KB/s	—

Name	Sequence ↑	Time
Name Lookup		5ms
Connect		2ms
App Connect		–
Pre Transfer		< 1ms
Start Transfer		7ms
Transfer		< 1ms

The **Test Instances** table displays detailed information about each test instance, including the status, response time, location where the test ran, number of errors, and time stamp from when the test ran. To drill down into a test instance, click **Expand**. Detailed response information is listed for each step in the test instance. You can sort any columns to help you to quickly identify the exact step where a slowdown or failure occurred. Viewing the errors, test sequence and response time helps you to identify issues easily.

The information that is displayed depends on the type of Synthetic test that is being monitored:

API

When you click **Expand** for an API test instance, a high-level summary of the following details is displayed:

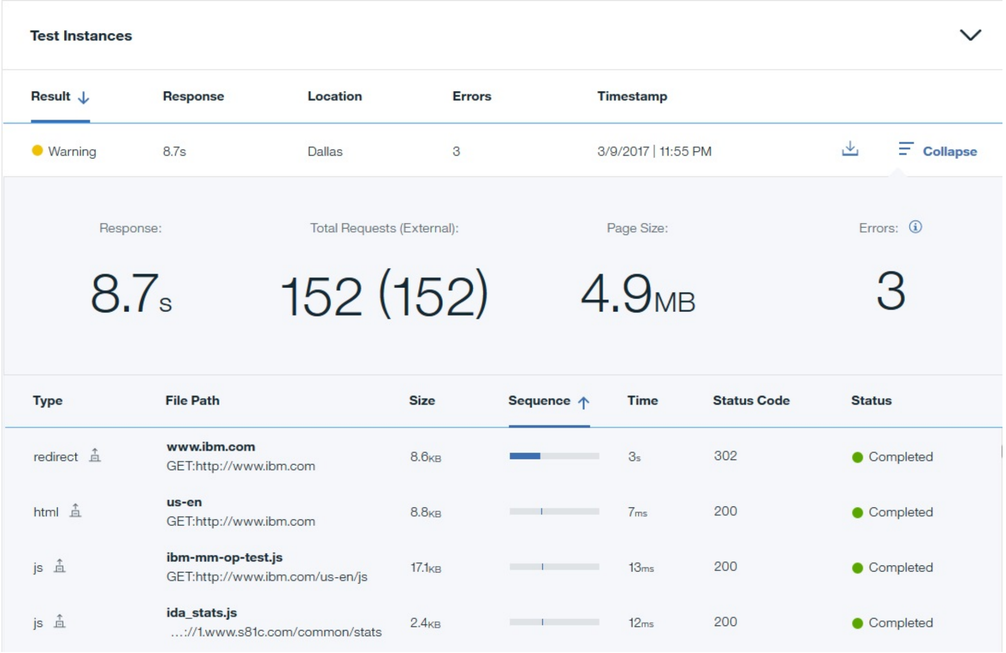
- **Response** displays the total response time for the test instance, including redirect time.
- **Redirect** displays the total redirect time for the test instance.
- **Size** displays the size of the object.

- **Download Speed** displays the speed at which each object downloads.
- **Errors** displays the number of errors that occurred during the test instance. To view the error details, click the **Information** icon.

A table displays each step in the API call, along with the name of the step, the sequence of steps, and the response time for each step. The following step names are displayed:

- **Name Lookup** represents the time that the test instance took to resolve the name of the object.
- **Connect** represents the time that the test instance took from the start of the step until a connection to the remote host or proxy was completed.
- **App Connect** represents the time that the test instance took from the start of the step until the SSL connection with the remote host was completed.
- **Pre Transfer** represents the time that the test instance took from the start of the step until just before the file transfer command begins.
- **Start Transfer** represents the time that the test instance took from the start of the step until the first byte is received.
- **Transfer** represents the time that the test instance took to transfer the file.

Webpage



Test Instances

Result	Response	Location	Errors	Timestamp		
● Warning	8.7s	Dallas	3	3/9/2017 | 11:55 PM		Collapse

Response: 8.7s

Total Requests (External): 152 (152)

Page Size: 4.9MB

Errors: 3

Type	File Path	Size	Sequence	Time	Status Code	Status
redirect	**www.ibm.com** GET:http://www.ibm.com	8.6KB		3s	302	● Completed
html	**us-en** GET:http://www.ibm.com	8.8KB		7ms	200	● Completed
js	**ibm-mm-op-test.js** GET:http://www.ibm.com/us-en/js	17.1KB		13ms	200	● Completed
js	**ida_stats.js** ...://1.www.s81c.com/common/stats	2.4KB		12ms	200	● Completed

When you click **Expand** for a webpage test instance, a high-level summary of the following details is displayed:

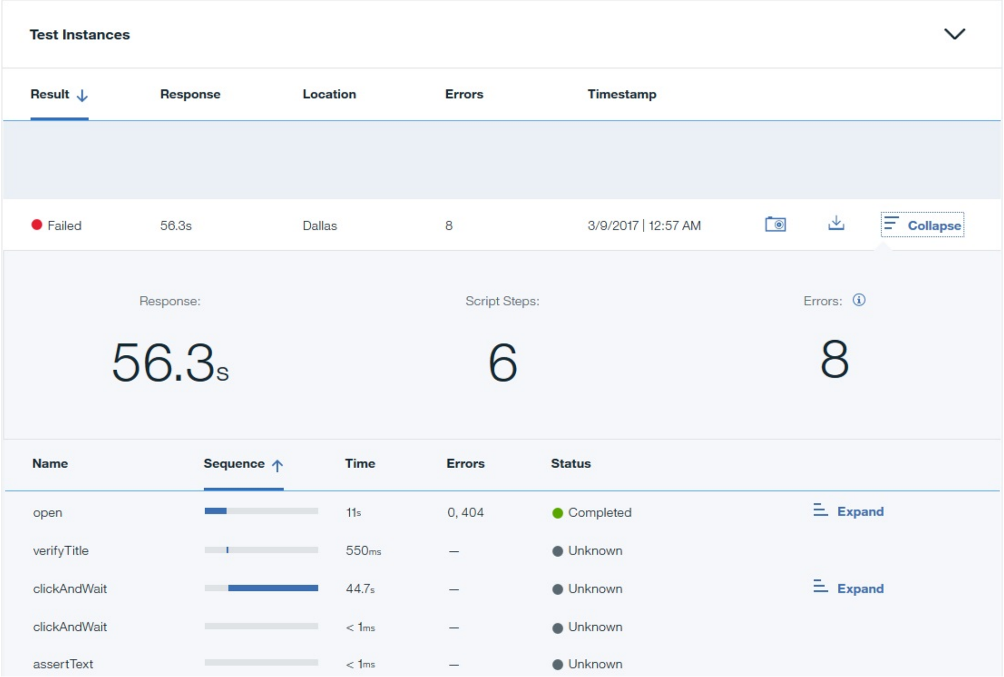
- **Response** states the response time for the test instance.
- **Total Requests (External)** displays the total number of requests for the test instance. The number of external requests is in parentheses.
- **Page Size** displays the size of the web page.
- **Errors** displays the number of errors that occurred during the test instance. To view the error details, click the **Information** icon.

A table that lists the following details for each request that is made by the test is also displayed:

- **Type** displays the type of request, for example, HTML, CSS, JavaScript, or image. External and internal requests are depicted by icons.
- **File Path** describes the location of the requested object.
- **Size** displays the size of the requested object.

- **Sequence** displays the sequence of requests that are made by the test.
- **Time** displays the time that each request takes.
- **Status Code** displays the HTTP request status code.
- **Status** describes the result of the request, for example, Completed, Unknown, or Failed.

Script



Test Instances

Result	Response	Location	Errors	Timestamp	
● Failed	56.3s	Dallas	8	3/9/2017 | 12:57 AM	Collapse

Response: 56.3s

Script Steps: 6

Errors: 8

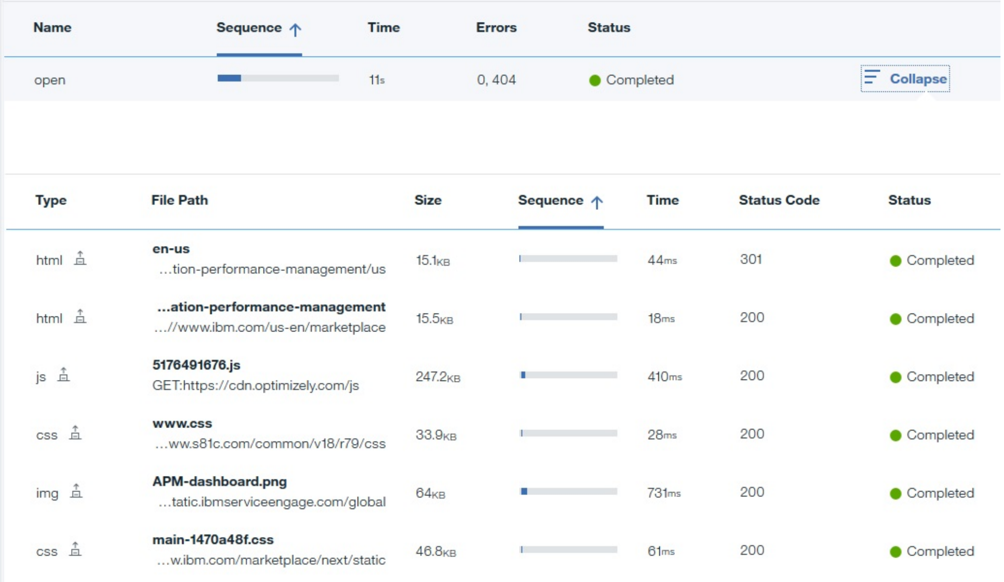
Name	Sequence	Time	Errors	Status	
open		11s	0, 404	● Completed	Expand
verifyTitle		550ms	—	● Unknown	
clickAndWait		44.7s	—	● Unknown	Expand
clickAndWait		< 1ms	—	● Unknown	
assertText		< 1ms	—	● Unknown	

When you click **Expand** for a script test instance, the response time, number of script steps, and number of errors are displayed. To view the error details, click the **Information** icon.

The following details for each script step are displayed in a table:

- **Name** displays each Selenium command that is called by your test instance, for example `Open`, `ClickAt`, or `VerifyBodyText`.
- **Sequence** displays the sequence of script steps from the beginning to the end of the test instance.
- **Time** displays the time that each script step takes.
- **Errors** displays the number of errors that occurred during each script step.
- **Status** describes the result of the script step, for example, Completed, Unknown, or Failed.

You can drill down and view details about the requests that are generated by each script step.



Name	Sequence ↑	Time	Errors	Status	
open		11s	0, 404	● Completed	Collapse

Type	File Path	Size	Sequence ↑	Time	Status Code	Status
html	**en-us** …tion-performance-management/us	15.1KB		44ms	301	● Completed
html	**…ation-performance-management** …//www.ibm.com/us-en/marketplace	15.5KB		18ms	200	● Completed
js	**5176491676.js** GET:https://cdn.optimizely.com/js	247.2KB		410ms	200	● Completed
css	**www.css** …ww.s81c.com/common/v18/r79/css	33.9KB		28ms	200	● Completed
img	**APM-dashboard.png** …tatic.ibmserviceengage.com/global	64KB		731ms	200	● Completed
css	**main-1470a48f.css** …w.ibm.com/marketplace/next/static	46.8KB		61ms	200	● Completed

Click **Expand** to view a table that contains the following details:

- **Type** displays the type of request, for example, HTML, CSS, JavaScript, or image. External and internal requests are depicted by icons.
- **File Path** describes the location of the requested object.
- **Size** displays the size of the requested object.
- **Sequence** displays the sequence of requests that are made by the test.
- **Time** displays the time that each request takes.
- **Status Code** displays the HTTP request status code.
- **Status** describes the result of the request, for example, Completed, Unknown, or Failed.

Availability Monitoring can automatically create a screen capture if the web page fails to load or a step in the script fails. For example, if one of the steps in your script opens a web page but it does not load, Availability Monitoring automatically creates a screen capture. To view a screen capture of the web page or script, click the **Screen Shot Error** icon. This feature is only available for webpage and scripted tests. It does not work with REST API tests.

You can also download a recording of the network traffic for a particular test instance as a `.har` file by clicking the **Download** icon. This feature is available for webpage and scripted behavior tests.

Response Time and Availability

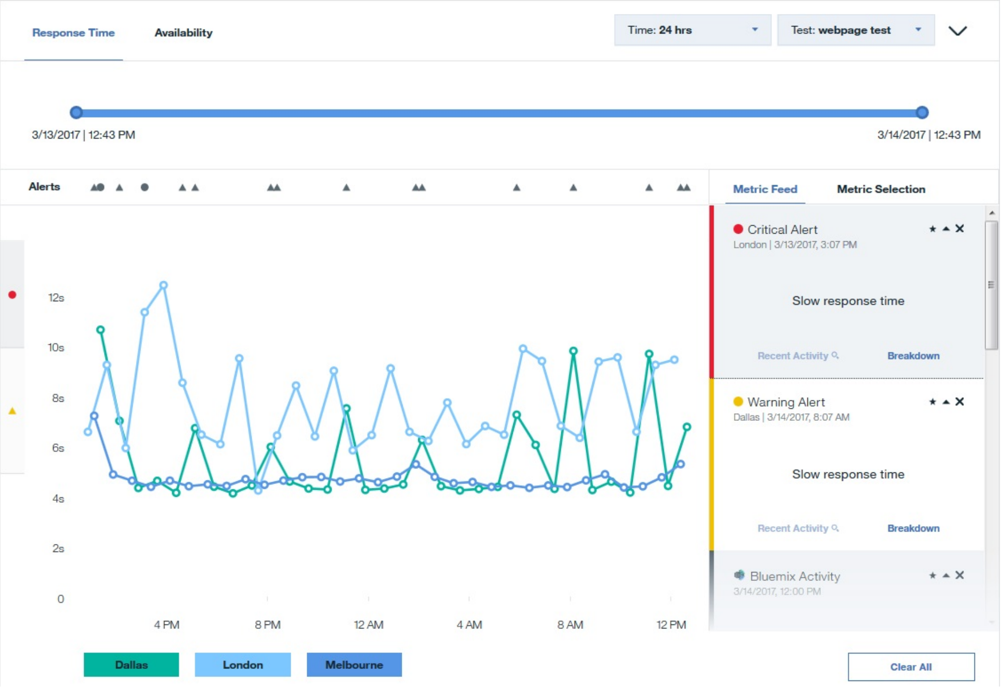
The **Response Time and Availability** pane displays a graph of the measured response times and availability for instances of your test over a defined period. For more information, see "Response Time and Availability" on page 1044.

Response Time and Availability

Use the **Response Time** and **Availability** pane to help you to visualize response time, availability trends, and alerts over time.

Response Time graph

Response time information is displayed on a line graph. To view it, click the **Response time** tab.



Critical Alert
London | 3/13/2017, 3:07 PM

Slow response time

Warning Alert
Dallas | 3/14/2017, 8:07 AM

Slow response time

Bluemix Activity
3/14/2017, 12:00 PM

Important: Response times that are measured by Availability Monitoring are slightly greater than response times that are experienced by users. Availability Monitoring simulates real user behavior, which adds to the response time measurement. The greater response time is due to the following factors:

- Availability Monitoring creates a new Firefox instance for each test to prevent previous test instances from influencing the current test. Real users might experience faster response times due to browser caching.
- Availability Monitoring installs the Firefox web driver plug-in before each test.

Individual response times for tests are represented by a **Response point** icon ○ on the line graph. Different colors denote different geographic locations where the app is running. The graph's y-axis uses alert icons to identify the warning and critical threshold ranges. The yellow warning icon ▲ represents the warning threshold range, and the red critical icon ● represents the critical threshold range. Click the yellow warning icon ▲ or the red critical icon ● to identify easily test instances that appear in the warning and critical threshold ranges. To view the details for a specific test instance, click the **Response point** icon ○ on the graph.

Filters

Choose a test from the **Test** drop-down menu. You can filter data for 3 hours, 24 hours, 7 days, 30 days, and 12 months. When you filter for a time range greater than 24 hours, the values that are displayed in the graph are averaged. To view more specific information, click the graph to drill down to the individual alerts and warnings. You can also use the slider to narrow or expand the time range.

With the Response Time graph, you can highlight and hide the data from particular PoP locations. To highlight response time data for a particular location, hover over the PoP location name; then, click the **Highlight location** icon 👁. To hide response time data for a location, hover over the PoP location name; then, click the **Hide location** icon −. To restore PoP location data to the graph, click **Add more locations** ⊕ or the **Metric Selection** tab; then, click the PoP location that you previously removed.

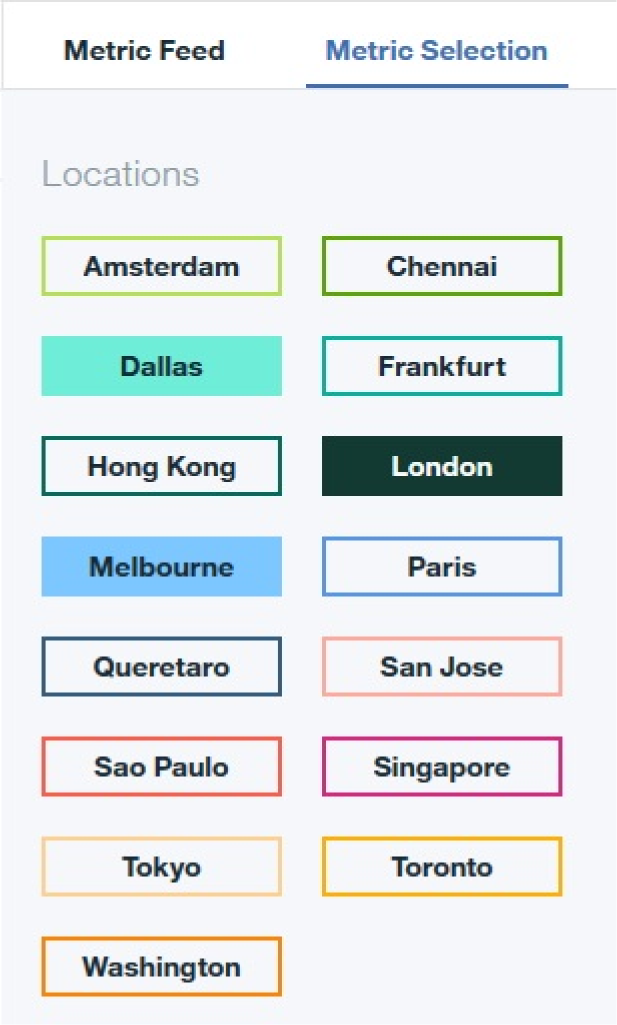
Alerts

You can easily identify warning and critical alerts in the Alerts row. Hover over an **alert icon** ● ▲ to identify the severity and timestamp for that alert. Click an **alert icon** to display details for that alert in the **Metric Feed**.

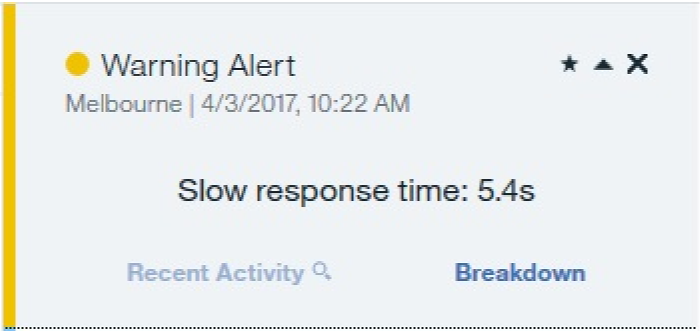
If there is more than one icon close together on the Alerts row, a **number icon** displays the number of alerts at that time. Hover over a **number icon** to display the individual alerts, and click an alert to view information in the **Metric Feed**

Metric Selection and Metric Feed

To filter for metrics by geographic region, click **Metric Selection**. Click a location to add or remove metrics that are measured at that location from the graph. Click **Add more locations** to open the Test Edit Mode page and add a PoP location to your selected test.



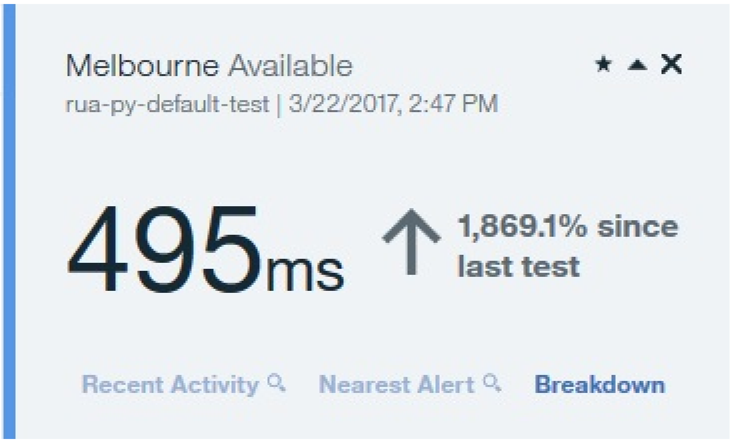
To view a list of metric details, click **Metric Feed**. The **Metric Feed** displays a list of the instances where a metric is fulfilled.


Warning Alert

Melbourne | 4/3/2017, 10:22 AM

Slow response time: 5.4s

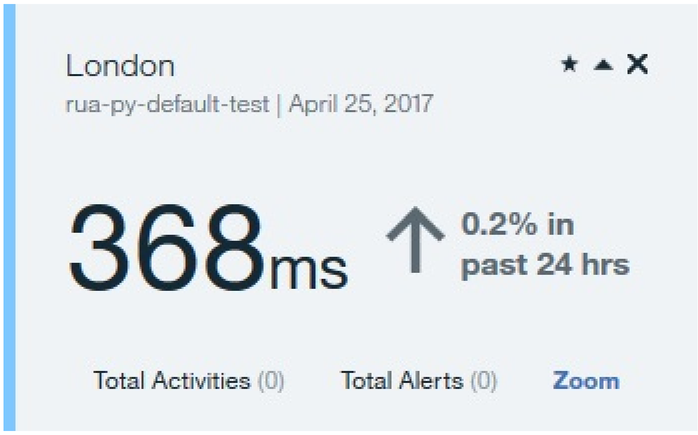
Click an **alert icon** or **response point** ○ on the graph to add the details of that metric to the **Metric Feed**.


Melbourne Available

rua-py-default-test | 3/22/2017, 2:47 PM

495ms ↑ 1,869.1% since last test

If you filter the Response Time graph for a time range greater than 24 hours and click a **response dot**, you can see aggregated details for that day in the **Metric Feed**.


London

rua-py-default-test | April 25, 2017

368ms ↑ 0.2% in past 24 hrs

Total Activities (0)

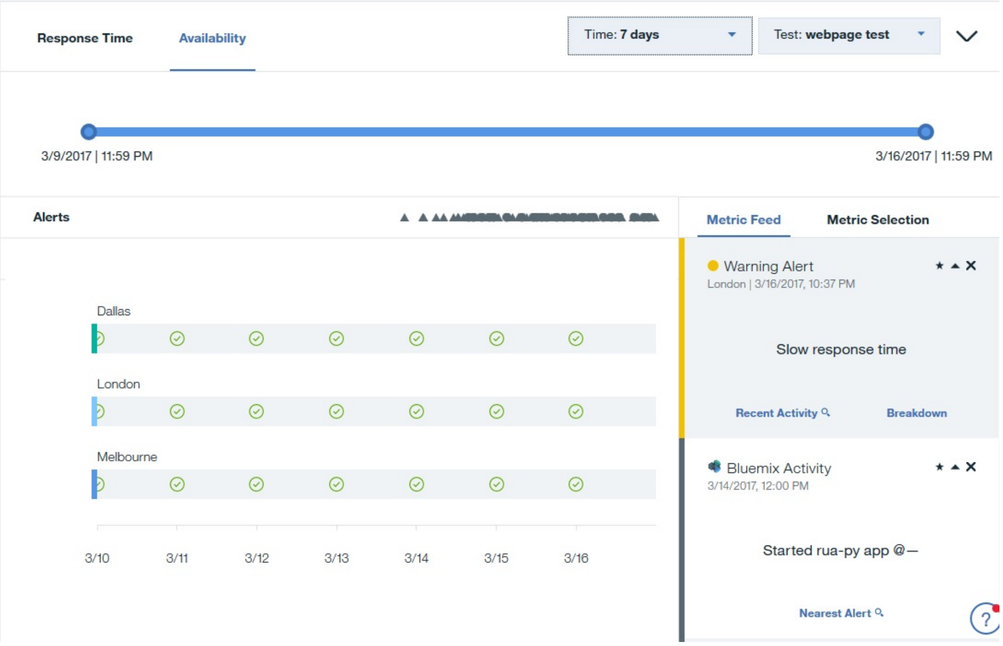
Total Alerts (0)

Click **Zoom** to view all response times and alerts that were generated by test for that day in the Response Time graph.

To view detailed information about an alert or test response time, click **Breakdown** in the **Metric Feed**. Click **Nearest Alert** to view the nearest alert to that test instance in the **Metric Feed**, if an alert occurred.

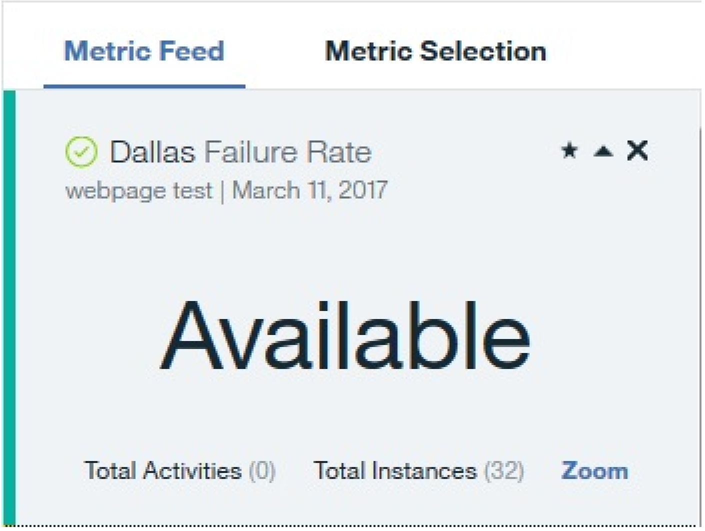
Availability

To view the availability information for your apps, click **Availability**. The Availability graph shows the daily availability of each point of presence (PoP) for the selected test.



With the Availability graph, you can highlight and hide the data from particular PoP locations. To highlight availability data for a particular location, hover over the PoP location name; then, click the **Highlight location** icon 👁. To hide availability data for a location, hover over the PoP location name; then, click the **Hide location** icon ➖. To restore PoP location data to the graph, click the **Metric Selection** tab; then, click the PoP location that you previously removed.

Hover over a graph point to display the failure rate and number of test instances for a particular day and location. Click a graph point to display this information in the **Metric Feed**.



Click **Zoom** to filter the **Availability** graph and the **Response Time** graph to display information for the selected day.

Availability Monitoring usage

You can view details about your Availability Monitoring usage on the **Monitoring** tab on your application pane, and on the Availability Monitoring main dashboard.

To see an overview of your usage from the Availability Monitoring main dashboard, click the **Configure** icon. If you are a user of the Availability Monitoring trial, your usage is displayed as a bar graphic and as a percentage, along with the number of tests in use. If you are a user of the Paid plan, your usage is displayed in data points. You can view the usage details for each individual test in the **Synthetic Tests** pane.

Your usage is measured in data points. The estimated number of data points is calculated from the following formula:

Estimated number of data points = T * L * (60/M * 24 * 30) per month

Where T = number of synthetic tests that are executed, L = number of locations, and M = interval between tests (minutes).

Simple tests such as webpage and REST API tests use 1 data point for each test. Advanced tests such as Selenium scripts and REST API scripts use 100 data points for each test.

Exploring the APIs

Use the IBM Cloud Application Performance Management APIs to create scripts for automating the onboarding of your Cloud APM environment.From the Cloud APM API-managed service offering in API Explorer on IBM developerWorks, you can access and explore the available Resource Group Management Service API, Threshold Management Service API, and Role-Based access control Service API.

Before you begin

You must have an active Cloud APM subscription to get a client ID key and run API operations.

Procedure

1. Open the API Explorer in your browser: https://developer.ibm.com/api.
2. Sign in with your IBMid.
3. In the **Search All APIs** field, enter `performance management` and click 🔍.
4. Select the IBM Cloud Application Performance Management API tile.
5. Click **Documentation** on the left side of the API Explorer window.
6. Select the specific API.
7. Select the sub section to expand the list of API operations.
8. To proceed, you need an active Cloud APM subscription. Complete one of the following steps:

   a) If you don't already have a Cloud APM subscription, Sign up for a free 30-day trial subscription.

   b) If you already have a subscription, sign in by clicking **My APIs** to test some of the API operations within API Explorer.

   When your subscription is active and you are signed in to the API Explorer page, you see a list of your API subscriptions.
9. Select an API operation for more details.
10. Select one of the languages (such as shell or curl) at the top of the API Explorer page to view a request example.
11. Retrieve your client ID key and your client secret key and store them in a safe place for external use.

    Send your client ID and client secret keys with each time API request. You must have a Cloud APM subscription to complete this action.

What to do next

For more information about how to run API operations, see the following topics:

“Accessing and using the Role-Based Access Control Service API” on page 991
“Using the Resource Group Management Service API” on page 978
“Using the Threshold Management Service API” on page 980

Advanced Configuration

Use the **Advanced Configuration** page to control communications settings and advanced features such as event forwarding.

After you click **System Configuration** > **Advanced Configuration**, the following configuration categories are displayed in the Advanced Configuration page.

UI Integration

For products that integrate with the Cloud APM console, you can add or edit the URL for launching the integrated application. The fields are populated with any URLs that were set during the integration configuration procedure.

- **Log Analysis URL** is used to launch IBM Operations Analytics - Log Analysis for searching through application logs from the Application Performance Dashboard. For more information, see “Searching log files” on page 1057.
- **Enable Subnode Events**, for agents with subnodes, controls whether subnodes are shown in the Events tab. When subnode events are enabled, the node and subnode for which an event was opened are displayed. Specifically, if you want to display log file monitoring situations in the Events tab, you must ensure that Subnode events are enabled. Default: `False`.
- **Dashboard Refresh Rate** controls the frequency of the Application Performance Dashboard automatic refresh. You can adjust the setting to any value from 1 to 60 minutes. The setting affects resource status that is displayed in the navigator and Status Overview tab. It has no affect on the Events tab entries. Default: 1 minute.

Event Manager

The Event Manager controls the forwarding of events using Simple Mail Transfer Protocol, as well as email notifications. If you enter a value for Target Email Addresses, an email will be sent for each open, close, and stop event. You can use the **Event Manager** fields to configure your events from Cloud APM to automatically open tickets in IBM Control Desk. For additional configuration tasks, see Integrating with Control Desk “Integrating with Control Desk” on page 954.

If you configure the SMTP forwarder to use SSL, you must add the SMTP Server’s signing CA certificate to the Cloud APM server keystore. Add the CA certificate to the default keystore using the JVM keytool command:

```
install_dir/java/jre/bin/keytool -importcert\
-noprompt \
-alias your_CA cert_alias \
-file path_to_your_CA_cert_file (*.cer)
 -keystore /install_dir/wlp/usr/servers/min/resources/security/key.jks \
-storepass ccmR0cKs! \
-storetype jks \
-trustcacerts
```

To view a sample email, see “Event email” on page 1052.

- **Target Email Addresses** specifies the email addresses that events are forwarded to. Separate each address with a comma (,), such as `annette@ibm.com,jim@ibm.com,owen@ibm.com`.
- **Alert Notification for ITMv6** is the option to enable Alert Notification for ITMv6 events by setting the value `True`. Default: `False`
- **Cloud Event Management Webhook** is the Webhook url that is generated in Cloud Event Management when you configure the integration between IBM Cloud Application Performance Management and Cloud Event Management. You must paste the generated Webhook url here so that events are forwarded from Cloud APM.

You can use Alert Notification instead of the Event Manager email function, which enables you to control who is notified for different events and how they are notified. For more information, see “Integrating with Alert Notification” on page 953.

If you are forwarding events to an EIF (Event Integration Facility) receiver, you can customize the EIF slots such as to add an attribute to the EIF event. For more information, see “Customizing an event to forward to an EIF receiver” on page 972. For information about forwarding your events to the IBM Netcool/OMNIbus event manager, see “Integrating with Netcool/OMNIbus” on page 945.

Tracking Analytics Service

The settings that are used for the Tracking Analytics Service. The settings apply only to the Cloud APM, Advanced offering and if you are configuring transactions tracking in your environment.

- **Connection Pool Size** is the number of concurrent Db2 connections that the Tracking Analytics Service holds in the connection pool for the "Top *N*" query. Increase this value if you are experiencing slow query times due to a large number of concurrent Cloud APM console users. Default: `10`.
- **Pseudo Nodes** enable the visualization of services that are not instrumented. Default: `True`.
- **Query Timeout in Seconds** is the number of seconds before each "Top *N*" (where *N* is a number, as in "Top 5" or "Top 10") query take before timing out. The timeout value can be increased if needed for applications that have a higher workload and have larger expected query times. Default: `120` seconds.
- **DB2 Query Re-optimization Enabled** should not normally need to be changed. The parameter affects the Db2 query optimizer. In some environments, turning off the optimizer might improve performance for some sets of transactions. Default: `False`.

Agent Subscription Facility

The Agent Subscription Facility includes the agent REST (Representative State Transfer) interface and Central Configuration Services HTTP server. The REST interface is used by agents and data collectors to send monitoring data that is persisted in the Db2 server and threshold events. The Central Configuration Services HTTP server handles requests from agents for their configuration files, such as threshold definitions. Use these parameters for configuring the communications between the Agent Subscription Facility and the Cloud APM server.

- **Missed Poll Limit (Fast Heart Beat)** is the maximum number of times that a monitoring agent with a 60-second or lower heartbeat interval fails to connect before it is marked offline. Default: `30` intervals.
- **Missed Poll Limit (Slow Heart Beat)** is the maximum number of times that a monitoring agent with a heartbeat interval that is greater than 60 seconds fails to connect before it is marked offline. Default: 3 intervals.
- **Transaction Time Out** is the amount of time, in seconds, that the server waits for a response to a request. Default: `120` seconds.
- **Remove Offline System Delay** determines the number of minutes to wait before removing the display of a managed system that is offline. In the Application Performance Dashboard, offline managed systems are indicated by the ◆ unknown status indicator. The managed system continues to display, even if you uninstall the agent, until the delay time has passed. For more information, see “Viewing and removing offline agents” on page 1078. Default: `5760` minutes (4 days).

Thresholds Enablement

Your monitoring agents each come with a set of predefined thresholds that are enabled and started with the agent. These predefined thresholds are assigned to the default system resource group for the agent.

- **Choose action to define policy for predefined best practice thresholds** controls whether the predefined thresholds for your managed resources are enabled or disabled by default. Set the field to **Disable All** if you do not want to run the predefined thresholds. The **Disable All** setting removes the assignment of the system group from all predefined thresholds. A threshold with no group assigned is distributed to no monitored systems and remains stopped until it is distributed to a resource group. If you decide later that you want to turn on the predefined thresholds, set the field to **Enable All**.

For more information about the predefined thresholds and custom thresholds, see “Background information” on page 958 and “Examples of disabled thresholds” on page 959.

Event email

Use the Event Manager fields on the Advanced Configuration page to set up event notification by email to a list of addressees.

Email for open event
When a threshold condition becomes true, an event is opened and the email message sent by the Cloud APM server contains the base attributes that apply to all agent events + attributes from the first row of the dataset that matched the threshold condition. The situation_status attribute has a value of Y for open events.

Email for close events
When the threshold condition is no longer true, a close event is generated. The email message for close events only contains the base attributes that apply to all agent events and the situation_status value is N. The agent's attributes are not included in these email messages because the threshold condition is not met.

Email for stop event
When a threshold is stopped, a stop event is generated. The email message for stop events only contains the base attributes that apply to all agent events and the situation_status value is P. The agent's attributes are not included in these email messages because the threshold condition is not met.

A threshold is stopped for an agent if you delete the threshold definition or you make a change to any of the threshold definitions that are distributed to the agent.

The following samples shows an email for an open event:

```
From:    noreply@apm.ibmserviceengage.com
To:         tester@us.ibm.com
Date:    10/25/2017 01:56 PM
Subject:    Linux_Disk_Space_Low on nc049048:LZ (Notification)
The text below lists the information received from the agent that triggered this event.
The IP and Agent values identify the agent that detected the event.
The Description and Severity values specify the name of the threshold definition and its
severity.
Below the Description are all of the attribute/value pairs present in the event, in their raw
form.
    Server IP  : 10.107.76.230 (SIDR26APAP1BLUE.test.ibm.com)
    Agent IP   : 9.42.49.48
    Agent      : nc049048:LZ
    Severity   : warning
    Description: Linux_Disk_Space_Low[Disk_Free_Percent <= 20  AND Disk_Free_Percent > 10
                 AND FS_Type != nfs  AND FS_Type != iso9660 ]
      ITM_KLZ_Disk
      ManagedSystemGroups='*LINUX_SYSTEM'
      TenantID=F43E-D704-DADC-6270-1ED8-543E-A388-6513
      adapter_host=nc049048.tivlab.raleigh.ibm.com
      apm_hostname=SIDR26APAP1BLUE.test.ibm.com
      appl_label=A:P:S
      date=01/25/2017
      disk_free=5843
      disk_free_percent=20
      disk_name=/dev/sda2
      disk_used=22676
      disk_used_percent=80
      file_system_status=2
      file_system_status_enum=Up
      fqhostname=nc049048.test.ibm.com
      fs_type=ext4
      hostname=nc049048.test.ibm.com
      identifier=Linux_Disk_Space_Lownc049048:LZ/ITM_KLZ_Disk
      inodes_free=1721587
      inodes_free_percent=88
      inodes_used=232477
      inodes_used_percent=12
      integration_type=U
      mount_options=rw
      mount_point=/
      msg='Linux_Disk_Space_Low[Disk_Free_Percent <= 20  AND Disk_Free_Percent > 10  AND FS_Type !
= nfs
     AND FS_Type != iso9660 ]'
      origin=9.42.49.48
      severity=WARNING
      situation_displayitem=/
      situation_eventdata='disk_name=/dev/
sda2;inodes_used_percent=12;mount_options=rw;fs_type=ext4;

system_name=nc049048:LZ;mount_point=/;disk_used_percent=80;disk_free=5843;file_system_status_enum=
Up;
     size=30040;disk_used=22676;inodes_used=232477;disk_free_percent=20;file_system_status=2;
     total_inodes=1954064;inodes_free=1721587;timestamp=1170125135553000;inodes_free_percent=88;~'
      situation_name=Linux_Disk_Space_Low
      situation_origin=nc049048:LZ
      situation_origin_uuid=09fb36afd6b3.22.02.09.2a.31.30.56.9d
      situation_status=Y
      situation_thrunode=nc049048:LZ
      situation_time='01/25/2017 13:55:55.000'
      situation_type=S
      size=30040
      source='ITM Agent: Private Situation'
      sub_origin=/
      sub_source=nc049048:LZ
      system_name=nc049048:LZ
      timestamp=1170125135553000
      tmz_diff=18000
      total_inodes=1954064
To unsubscribe from these emails: Log into the Cloud APM console and remove your email address
from
the list of target email addresses in the Event Manager category of the Advanced Configuration
page.
```

The following sample shows an email for a close event.

```
From:    noreply@apm.ibmserviceengage.com
To:      tester@us.ibm.com
Date:    01/25/2017 02:01 PM
Subject:    Linux_Disk_Space_Low on nc049048:LZ (Closed)
The text below lists the information received from the agent that triggered this event.
The IP and Agent values identify the agent that detected the event.
The Description and Severity values specify the name of the threshold definition and its
severity.
Below the Description are all of the attribute/value pairs present in the event in their raw
form.
    Server IP  : 10.107.76.230 (SIDR26APAP1BLUE-12f.test.ibm.com)
    Agent IP   : 9.42.49.48
    Agent      : nc049048:LZ
    Severity   : warning
    Description: Linux_Disk_Space_Low[Disk_Free_Percent <= 20  AND Disk_Free_Percent > 10
                 AND FS_Type != nfs  AND FS_Type != iso9660 ]
      ITM_KLZ_Disk
      ManagedSystemGroups='*LINUX_SYSTEM'
      TenantID=F43E-D704-DADC-6270-1ED8-543E-A388-6513
      adapter_host=nc049048.tivlab.raleigh.ibm.com
      apm_hostname=SIDR26APAP1BLUE-12f.test.ibm.com
      appl_label=A:P:S
      date=01/25/2017
      fqhostname=nc049048.test.ibm.com
      hostname=nc049048.test.ibm.com
      identifier=Linux_Disk_Space_Lownc049048:LZ/ITM_KLZ_Disk
      integration_type=U
      msg='Linux_Disk_Space_Low[Disk_Free_Percent <= 20  AND Disk_Free_Percent > 10  AND FS_Type !
= nfs
     AND FS_Type != iso9660 ]'
      origin=9.42.49.48
      severity=WARNING
      situation_displayitem=/
      situation_eventdata=~
      situation_name=Linux_Disk_Space_Low
      situation_origin=nc049048:LZ
      situation_origin_uuid=09fb36afd6b3.22.02.09.2a.31.30.56.9d
      situation_status=N
      situation_thrunode=nc049048:LZ
      situation_time='01/25/2017 14:00:55.000'
      situation_type=S
      source='ITM Agent: Private Situation'
      sub_origin=/
      sub_source=nc049048:LZ
      tmz_diff=18000
To unsubscribe from these emails: Log into the Cloud APM console and remove your email address
from
the list of target email addresses in the Event Manager category of the Advanced Configuration
page.
```

Chapter 10. Using the dashboards

Select **Performance** > **Application Performance Dashboard** to get a comprehensive status overview of your applications. You can drill down from the highest level overview to in-depth metrics in the same display.

Use the tools that are available on the dashboards to investigate critical and warning conditions in your environment, create additional metric views, and to perform actions such as searching trace logs and comparing metrics over time.

All My Applications - Application Performance Dashboard

The Application Performance Dashboard presents the summary status for your monitored domains in **All My Applications**. A *summary box* is displayed for each user defined application, such as "Inventory Management", and for the predefined applications, "My Components" or "My Transactions" if your environment includes them. From the summary boxes or the navigator, drill down to each application and its constituents to see detailed metrics.

As you select items, the path is shown and you can click one of the path links to return to that view. From any Cloud APM console page, you can click **Performance** > **Application Performance Dashboard** to open the **All My Applications** dashboard. View areas of interest either by selecting from the navigator or by clicking in a summary box to drill down to the next level.

Summary boxes

All My Applications has a summary box for each defined application. Indicators show the highest status severity for the application in the title bar and for each group in the summary box. The following predefined groups are available, depending on which monitoring products are included in the defined application:

Availability Monitoring has no subgroups

Components has a subgroup for each monitoring agent type that supports your application

Transactions can include **End User Transactions** and **Synthetic Transactions** (IBM Website Monitoring on Cloud before the August 2017 release) subgroups

In addition, each summary box includes **Events**, which shows the severity of the highest severity event that is open for the application. You can click the Events link to investigate any open events (see “Event Status” on page 1083).

Click a summary box title bar to open the Status Overview tab for the application. Or click one of the summary box icons to open the Status Overview tab for the Components group or Users or Transactions subgroup, or to open the Events tab for the application group or subgroup.

You can collapse the summary boxes and filter them by selecting or clearing check boxes:

- To show only the summary box title bars for easy scrolling through your defined applications, clear the **Show Details** check box.
- To filter the summary boxes for the severity status that you want to hide, clear the check box for a counter, such as ☐ ✓ 12 . The check box for a severity with no events is disabled. For example, in this graphic, the filters for Critical and Normal are enabled; Warning and Unknown are disabled because they have a 0 count: ☑ ✖ 2 ☑ ⚠ 0 ☑ ✓ 1 ☑ ◆ 0 .

Search

Use the Search field to find log entries from the past hour that contain the entered text. You can click to show results from a different time range. The Search text is compared to the log entries that are associated with the navigator selection and any matches are shown in a new browser tab or window. For instructions, see “Searching log files” on page 1057. The search capability is provided by IBM Operations Analytics - Log Analysis.

Actions

The **Actions** menu has options for copying the URL, opening the dashboard log, and setting a trace for troubleshooting. For more information, see “Copying the dashboard URL” on page 1095 and “Setting a trace” on page 1096.

Use the **Dashboard Log** option to review the list of agent dashboards that were updated since the last server restart.

When the **All My Applications** home dashboard or one of the applications is selected, the Actions menu includes **Launch to Reports** to help you analyze usage and performance trends if Cognos® based reports are available and your environment includes Tivoli Common Reporting. For more information, see “Reports” on page 1097.

When the **Components** group is selected, from either the navigator **Groups** section or from a summary box, the **Actions** menu includes an **Edit** option for editing the Component status overview dashboard. The **Edit** option is only available if the logged in user has the modify permission for Performance Management Dashboard and the create permission for Applications. For more information, see “Editing the Components dashboard group widgets” on page 1066.

? Help

Open the pop-up help to get a short description of the current dashboard, with the following links: **Learn More** opens the full dashboard topic in the local Cloud APM help system; and **Take a Dashboard Tour** starts the IBM Cloud APM Dashboard tour, which presents a brief description of the dashboard elements as it guides you through the features.

Navigator

The navigator displays a hierarchy of defined applications and their users, transactions, and components that reflects how they are organized. The navigator has a section for each level of the application hierarchy. At every level of the navigator, the dashboard metrics change to show data from the component. Select an item to switch the dashboard context to the selection. The scope of what you can see is determined by your user permissions.

Each navigator item has a ✖ critical, ⚠ warning, ✓ normal, or ◆ unknown status indicator, which indicates that the agent is unavailable. Each navigator section presents a count of events for each severity that is associated with the selected navigator item. For a mapping of status to threshold events, see “Event Status” on page 1083.

To make more room for other sections, click a title bar to collapse the section, and click again to restore the section. You can also hide the navigator entirely by clicking ◂ on the navigator border; and restore it by clicking ▸, or adjust the width by dragging the ↔ border.

The navigator has three sections:

- The **Applications** section lists all the defined applications in your domains or that are allowed for your user role.
  - After you select an application, the dashboard changes to a high-level status summary in the **Status Overview** tab, and an indicator of the highest severity status is displayed in the **Events** tab. For more information, see “Application - Application Performance Dashboard” on page 1058.
  - If your Cloud APM server is sized as small or medium, a predefined application named *My Components* is created. The application contains any resource monitoring components in your domain that have been discovered by the monitoring infrastructure and that you are authorized to view. For more information on authorization, see “Roles and permissions” on page 982. My Components cannot be edited or deleted. The event status of the My Components application and its components is based on the status of all components discovered by the monitoring infrastructure and not just the components and component instances that you are authorized to view. For example, if some Linux OS agents have Critical status but all of the Linux OS agents that you are authorized to view have Normal status, then the Linux OS status is Critical for the My Components application even though the status is Normal for all Linux OS agent instances that you see.

  - If you have administrator role privileges, the **Applications** toolbar includes ⊕ ⊖ ✎ tools for managing applications. For instructions, see “Managing applications” on page 1073.
- After you select an application, the **Groups** section shows the groups that support the application. For more information, see “Group and Instance - Application Performance Dashboard” on page 1063.
- After you select a subgroup, the **Instances** section is renamed for the subgroup title and populated with the individual managed system names. The **Status Overview** changes to show KPIs for the selected subgroup. After you select a managed system, detailed group widgets are displayed with KPIs from the managed system. For component instances, you also have an **Attribute Details** tab for viewing a table of KPIs from the data set attributes of your choosing. For more information, see “Viewing and managing custom charts and tables” on page 1068.

If you get a **Network Error** pop-up message in the navigator, the status indicators might change to ✓ normal until after the connection is restored. At that time, any open events are reprocessed and the status might appear normal until processing is completed.

Searching log files

To find the root cause of a problem that is experienced by users, such as slowness or a failure, you can search through log data that is associated with your applications. IBM Operations Analytics - Log Analysis provides the search capability. Application log data and performance data are brought together to help you find the root cause of a problem that is experienced by your applications and expedite problem resolution.

Procedure

Take the following steps to locate log entries that might correlate to a problem you are investigating, such as high CPU usage.

1. If the Application Performance Dashboard is not displayed, select it from the **Performance** menu.
2. If you want to search within an application, select one of the applications in the "All My Applications" dashboard.
   For example, click "My Components" to search the logs of all component resources.
3. Enter the log file text to find in the 🕒 🔍 search box.
   For example, enter `rolled back` to look for WebSphere Applications agent resources that were rolled back to the previous level.
4. If you want to find data from a time range other than **Last hour**, click 🕒 and select a different period.
5. Click 🔍.

Results

All log entries that contain the search text in the context of the current navigator level are displayed in a new browser tab or window. The browser window is named for the context, such as the "Credit Card Processing" application.

What to do next

Review the search results. You can select another application to change the search results for the context. Use the search field to further refine the results. For example, if the search field shows `db2 AND (datasourceHostName:Pear* OR datasourceHostname:Persimmon* OR datasourceHostname:Pomegranate*)`, you can delete data sources to narrow the results: `db2 AND (datasourceHostname:Persimmon*)`.

For more information, see the IBMOperations Analytics Log Analysis topic collection on IBM Knowledge Center or go to the IBM Operations Analytics - Developers Community.

Application - Application Performance Dashboard

After you select an application from the navigator or from a summary box in the **All My Applications** dashboard, a tabbed dashboard presents different facets of your application. The **Status Overview** tab presents a high-level status summary of your application. The chart thresholds and status indicators give overall health and performance feedback. Select the **Events** tab to see which event thresholds are contributing to application health.

For a description of the navigator and banner elements, see "Navigator" on page 1056, "Search" on page 1055, "Actions" on page 1056, and " Help" on page 1056.

Status Overview

- Depending on the composition of the selected application, the **Status Overview** tab presents one or more perspectives for evaluating application status at a high level:

  Availability Over Time

  IBM Website Monitoring on Cloud before the August 2017 release: The **Availability Over Time** bar chart is displayed if the application includes the Synthetic Playback agent (in the **Transactions** navigator group and in the predefined application, **My Transactions**).

  Each plot point is a transaction sample with a color indicator for a status of 🟩 **Healthy**, 🟨 **Slow** or 🟥 **Unavailable**.

  Click anywhere on the bar time line to open a pop-up window with **Transaction List** and **Location List** tables.

  Requests and Response Time

  The **Requests and Response Time** stacked bar chart is displayed if the application includes the Response Time Monitoring Agent (**End User Transactions** in the **Transactions** navigator group).

  Use this chart to look for trending patterns in performance. Each stacked bar plots the percentage of requests that completed with good response time, slow response time, or that failed to complete. The line chart overlay plots the average response time during the 5-minute period. Use the time selector to change the time range displayed, described in "Adjusting and comparing metrics over time" on page 1067.

  Aggregate Transaction Topology

  The **Aggregate Transaction Topology** is displayed when transaction tracking is enabled and the application includes any of the following agents or data collectors:

  - DataPower agent
  - HTTP Server agent
  - IBM Integration Bus agent
  - J2SE data collector
  - JBoss agent
  - Liberty data collector
  - Microsoft .NET agent
  - Microsoft SQL Server agent
  - Node.js data collector
  - Response Time Monitoring agent
  - SAP NetWeaver Java Stack agent
  - Tomcat agent
  - WebLogic agent (Linux and Windows only)
  - WebSphere Applications agent
  - WebSphere MQ agent

You must enable transaction tracking manually for all agents except the Response Time Monitoring agent. Transaction tracking is enabled automatically for data collectors. For more information, see "Agent Configuration page" on page 182.

The **Aggregate Transaction Topology** presents the resources that are associated with the application and their relationships. The footer shows a count of the selected nodes, resources, relationships, and any filters in the topology, as well as the time when the data was last refreshed.

If an application component is added to a business application, and the component carries traffic for multiple applications, the application topology that is displayed for those business applications includes paths to nodes for all applications.

For the IBM Java application stack where JavaScript is automatically injected, the highest level node represents the browser and the most granular node is the database. For other applications, the highest level node represents the application and the most granular node is the managed system instance.

Each node has a status indicator and background highlighting to show the highest status severity at that level of aggregation. If you collapse the navigator to make more space, you can still see the same status in the **Aggregate Transaction Topology**. The node's source environment shows as **Cloud** (IBM Cloud Application Performance Management), **ITM** (IBM Tivoli Monitoring), **On Premises** (IBM Cloud Application Performance Management, Private) , **Private Cloud** (IBM Cloud Private), or **Public Cloud** (IBM Cloud). No icon is shown for **Other** (managed resource is from another environment).

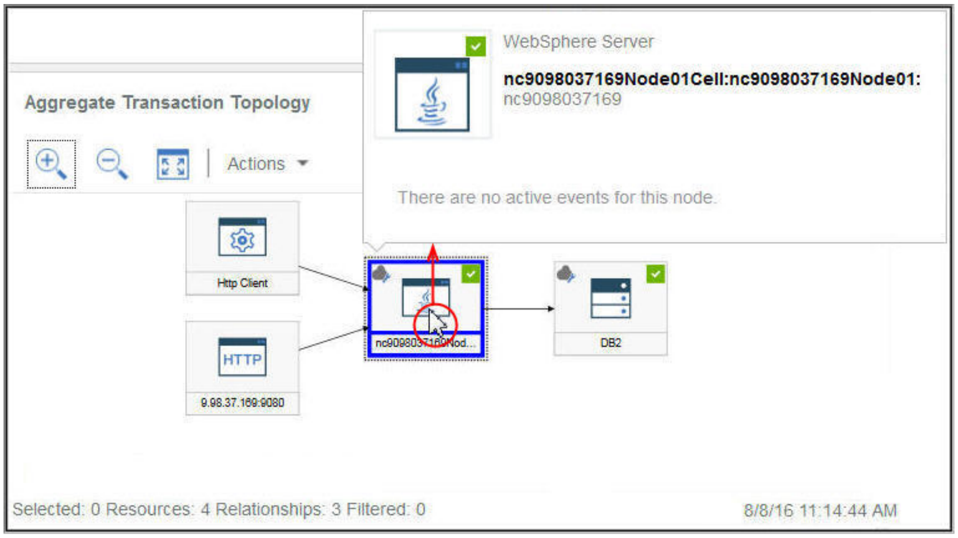
Hover the mouse over a node, open the shortcut menu, and select nodes to get more information about the status and help you identify the root cause of a problem:

- As you hover the mouse over a node, a pop-up message provides a list of the ⊗ critical and ⚠ warning events.
- Double-click a link URL in a node to open the corresponding dashboard with component or transaction details.
- Right-click a node and select one of the dashboard drill down options: **Go to Transactions Summary page** of the selected subgroup node; **Go to Component Instance page** of the instance node; or **Properties** to see the resource name, status, managed system name, and provider domain (such as "Cloud").

Use the toolbar icons to adjust the display and take actions as described in "Manipulating the Aggregate Transaction Topology widget" on page 1061.

Click the ➜ tool to toggle between this view and the **Current Components Status**, described next.

Any monitoring agents with no topology information do not show in the Aggregate Transaction Topology widget.



nc9098037169Node01Cell:nc9098037169Node01:
nc9098037169

There are no active events for this node.

Selected: 0 Resources: 4 Relationships: 3 Filtered: 0

8/8/16 11:14:44 AM

Current Components Status

The **Current Component Status** stacked bar chart shows the percentage and a count of critical, warning, normal, and unknown status for each component type in the application. Consider, for example, that 5 Linux systems are supporting the selected application. A stacked bar showing 40% critical and 60% normal indicates that 2 systems have critical status and 3 systems have normal status.

Hover the mouse over a bar segment to read the status in a pop-up window: the percentage and count of the component instances with that status. The domain or domains where the instances reside is also shown with a status count for each domain: IBM Cloud, Cloud, On Premises, ITM and Other. For example, 2 of your 5 Linux systems are in the ITM domain and 3 are in the Cloud domain. If one of the Critical systems is in the ITM domain and the other is in the Cloud domain, when you hover the mouse over the Critical 40% bar segment, the status pop-up window shows 1 system in the ITM domain and 1 in the Cloud domain.

You can click a bar to open the status summary dashboard for the component type, with a group widget for each monitored system.

Click the ➲ tool to toggle between this view and the **Aggregate Transaction Topology**, described earlier.

Availability Monitoring

When the defined application consists solely of Availability Monitoring, the summary dashboard is displayed as described in "Accessing Availability Monitoring" on page 1025.

- After you select a subgroup from the **Groups** section, the **Instances** section is renamed for the subgroup title and populated with the individual instance names. For information about the dashboards at the group, subgroup, and instance level of the navigator, and about the **Attribute Details** tab that opens after you select a managed system, see "Group and Instance - Application Performance Dashboard" on page 1063 and "Viewing and managing custom charts and tables" on page 1068.
- Some of the dashboard widgets show metrics that are based on a time range, and other widgets show the most recent metrics. If a time selector bar is displayed, you can adjust the time range for the dashboard that affects any charts or tables whose values are derived from historical data samples. For more information, see "Adjusting and comparing metrics over time" on page 1067.
While viewing charts, you can click a plot point to open a tool tip with the plot point value and other pertinent information. After viewing a line chart in the Internet Explorer Version 11 browser, you might continue to see the tool tip appear as you move the cursor around the window. If you experience this behavior, you can close the tool tip by clicking a few times in the chart.
- Data is auto-refreshed every minute in the console. This activity is essential and cannot be paused, stopped, or hidden.
- If no data is available for a chart or status summary box, an information message is displayed.

Events

- The status indicators that are displayed next to the **Events** tab title, such as ⊗**14** ⚠ **3**, show a count of the highest event severities for the selected navigator item: application, group, subgroup, or instance. Threshold severities are consolidated, as shown in the following table. For example, **Events** ⚠ means that the highest severity event is minor or warning.

Events tab	Threshold Severity
⊗Critical	Fatal and Critical
⚠Warning	Minor and Warning
✓Normal	Unknown

  When your managed environment includes IBM Operations Analytics - Predictive Insights and an anomaly is detected, an event is opened. A diamond-shaped icon overlays the status indicator, such as ⊗◆, to notify you that at least one anomaly has been detected by Operations Analytics - Predictive Insights. For example, **Events** ⚠◆, indicates that the highest status event is ⚠ Warning and that at least one anomaly event is open.
- Click the **Events** tab to see a summary of the total event count, a count of each severity type, and a percentage gauge for the severities. For more information, see "Event Status" on page 1083.

Custom Views

The pages that you create and save are associated with the selected application. For example, the Inventory Management application in the Cloud APM Guided Demo has the following monitoring agents: Linux OS, MySQL, Node.js, Hadoop, and Ruby. You can create and save a custom page at any level of the navigator from application to instance and then open it at the same level where it was created. A page that is created at a particular level can be opened only at the same level. The metrics available for the widgets can be from any of the resources in the application. Using the Inventory Management example, you can create a page with a table from the Ruby agent, a chart from Linux OS agent, and so on.

- The **Custom Views** tab is available at any level of the navigator when you select an application from **All My Applications**.
- After you open the **Custom Views** tab, the **Select a Template for your Custom Page** window is displayed or the default page is displayed if it is already set.
  - In the **Select a Template for your Custom Page** window, you can select a template to create a page.
  - On the default page, you can click + to create a new page.
- On the default page, Click ▾ in the page list and select one of the saved pages from the list.
- The options that you see in the Custom Views tab vary based on whether a page is being edited or viewed. For information about editing a page, see "Creating and managing custom pages" on page 1087. For information about viewing a page, see "Viewing custom pages" on page 1093.

Manipulating the Aggregate Transaction Topology widget

Use the **Aggregate Transaction Topology** widget to see the hierarchy of resources in the selected application. You can adjust and move around the display to see the status of each component and its relationship to other components, and open a node's corresponding dashboard.

Before you begin

After you select an application in the Application Performance Dashboard, the **Status Overview** tab is displayed with one or more charts, depending on the monitored resources that are included in the application.

The **Aggregate Transaction Topology** is displayed for the following agents that support transaction tracking:

- DataPower agent
- HTTP Server agent
- IBM Integration Bus agent
- J2SE data collector
- JBoss agent
- Liberty data collector
- Microsoft .NET agent
- Microsoft SQL Server agent
- Node.js data collector
- Response Time Monitoring agent
- SAP NetWeaver Java Stack agent
- Tomcat agent
- WebLogic agent (Linux and Windows only)
- WebSphere Applications agent
- WebSphere MQ agent

The **Aggregate Transaction Topology** shows a node object for every monitored resource that supports the topology feature.

Procedure

Take any of the following steps to manipulate the **Aggregate Transaction Topology** widget and open dashboards that are associated with the nodes:

- To open a linked dashboard, double-click the topology node. You can also right-click a node and select one of the dashboard drill-down options, **Go to Transactions Summary page** or **Go to Component Instance page**, or select **Properties** to see information about the managed system.
- To enlarge the topology display size, click **Zoom In**. You can also click **Actions** > **Zoom In** if no node is selected.
- To reduce the topology display size, click **Zoom Out**. You can also click **Actions** > **Zoom Out** if no node is selected.
- To adjust topology display size to fit within the current widget space, click **Fit Contents**. You can also click **Actions** > **Fit Contents** if no node is selected.
- To filter the topology nodes, select one of the indicators in the filter bar. You can toggle the filters on and off, and select multiple filters. Any node with a property that does not match the filter is dimmed, and the nodes that match the filter remain visible.
  - Normal, Warning, Critical, or Unknown to filter by node status.
  - To filter by environment, select **Cloud** (IBM Cloud Application Performance Management), **ITM** (IBM Tivoli Monitoring), **On Premises** (IBM Cloud Application Performance Management, Private) , **Private Cloud** (IBM Cloud Private), or **Public Cloud** (IBM Cloud).
  - Filter to add a custom filter.
- To make more room for the topology widget, click **Collapse Section** on the navigator or surrounding chart widgets, or drag a widget border.

Group and Instance - Application Performance Dashboard

Use the dashboard for the selected application group, subgroup, or instance to get a high level status of your managed systems. You can drill down to detailed dashboards with metrics for the selected instance and create custom charts and tables.

After you select an application under **All My Applications** in the Application Performance Dashboard, the **Status Overview** and **Events** tabs are displayed.

The navigator **Groups** section lists one or more of several possible groups, depending on the constituents of the defined application.

For a description of the navigator and banner elements, see "Navigator" on page 1056, "Search" on page 1055, "Actions" on page 1056, and " Help" on page 1056.

Status Overview

Groups and subgroups

- Select a group or ⊞ expand a group and select a subgroup to see a summary group widget for each managed system in the application. After you select a subgroup, the summary group widgets in the **Status Overview** tab are specific to that subgroup.
- The following predefined groups are available, depending on which monitoring products are installed:

  Availability Monitoring
  This group is displayed for custom applications. The navigation behavior for Availability Monitoring is different from the **Components** and **Transactions** groups.

  The Availability Monitoring add-on and dashboards are described in "Availability Monitoring" on page 1024.

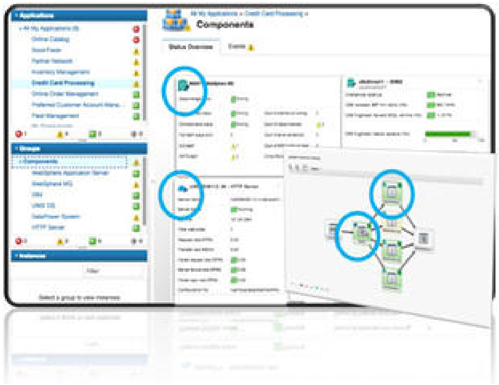
  Components
  This group is displayed for all applications, with the exception of the Response Time Monitoring Agent, the Synthetic Playback agent, and Availability Monitoring.

  The **Components** has a subgroup for each monitored software component that supports the selected application.

  Transactions
  This group includes **End User Transactions** and **Synthetic Transactions** (IBM Website Monitoring on Cloud before the August 2017 release) subgroups. For more information, see the help for Transaction Monitoring and the Synthetic Playback agent or their reference PDFs on the IBM Knowledge Center for Monitoring Agent Reference Guides.

  Note: The Transactions group is not available for the My Components application.

- After you select **Components** or a subgroup from the **Groups** section, the **Status Overview** tab changes to show a summary dashboard with a group widget for each managed resource. The source environment shows as **Cloud** (IBM Cloud Application Performance Management), **ITM** (IBM Tivoli Monitoring), **On Premises** (IBM Cloud Application Performance Management, Private) , **Private Cloud** (IBM Cloud Private), or **Public Cloud** (IBM Cloud). The **Instances** section is renamed for the subgroup title and is populated with the individual instance names.

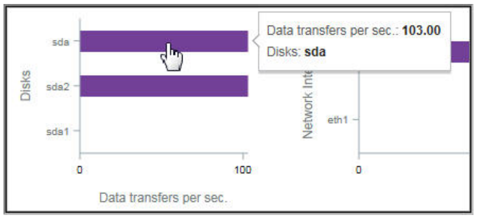


If the application has many managed system instances, many group widgets are displayed. You can scroll through the list to seem them all. You can also select a managed system type from the list of component subgroups, such as Windows OS, to confine the display to the same managed system types. You can also filter the managed system instances.

Firefox browser only: Depending on the number of agents and bandwidth, as you scroll down the Components page, you might see a pop-up message that the script to load the resource page takes a long time to complete. Select the option, "Don't ask me again" to disable the message and continue opening the widgets. Alternatively, you can enter `about:config` in the address box, search for **`dom.max_script_run_time`** and increase the time out value (in seconds). A value of 0 (zero) disables time out.

Instances

- Click inside a group widget or select the instance name from the navigator to open a detailed dashboard for the managed resource.
- If many instances are displayed in the navigator, use the search field in the Instances toolbar. As you type, any instances that do not match are removed from the display.
- To pause the Application Performance Dashboard automatic refresh, click **Pause** in the Instances toolbar; to resume automatic refresh, click **Resume**.
- The widgets and KPIs shown for any managed system might depend on the agent version. If an agent installed on the managed system is at an earlier version, it might be unable to provide as much information as the current version of the agent. A message is displayed instead of one or more KPIs in a chart or table when no data is available. The reason could be as simple as no data was reported for the time span. Or it could be related to a back-leveled agent that doesn't support the data set or an attribute included in the chart or table.
  To see a list of agent dashboards that were updated since the last Cloud APM server restart, select **Actions** > **Dashboard Log**.
- Some of the dashboard widgets show metrics that are based on a time range, and other widgets show the most recent metrics. If a time selector bar is displayed, you can adjust the time range for the dashboard that affects any charts or tables whose values are derived from historical data samples. For more information, see "Adjusting and comparing metrics over time" on page 1067.
  While viewing charts, you can click a plot point to open a tool tip with the plot point value and other pertinent information. After viewing a line chart in the Internet Explorer Version 11 browser, you might continue to see the tool tip appear as you move the cursor around the window. If you experience this behavior, you can close the tool tip by clicking a few times in the chart.
- If you're viewing a chart with bars missing, it means that the value is 0 (zero) for that data point.



- IBM Cloud Application Performance Management, Advanced users have additional diagnostics dashboards that are accessed by clicking the **Diagnose** link from a group widget in the details dashboard.

  Restriction: The managed system for which you are opening the diagnostics dashboards must reside in the IBM Cloud APM domain. If the managed system resides in the IBM Cloud or IBM Tivoli Monitoring source domain, the diagnostics dashboards are not available. See also "Cloud APM agent and Tivoli Monitoring agent coexistence" on page 930.
- If your environment includes the Synthetic Playback agent, you can launch Cloud APM reports for the agent instance from the **Actions** menu.

Events

- The status indicators that are displayed next to the **Events** tab title, such as ⊗**14** ⚠ **3**, show a count of the highest event severities for the selected navigator item: application, group, subgroup, or instance. Threshold severities are consolidated, as shown in the following table. For example, **Events** ⚠ means that the highest severity event is minor or warning.

Events tab	Threshold Severity
⊗Critical	Fatal and Critical
⚠Warning	Minor and Warning
✅Normal	Unknown

  When your managed environment includes IBM Operations Analytics - Predictive Insights and an anomaly is detected, an event is opened. A diamond-shaped icon overlays the status indicator, such as ⊗, to notify you that at least one anomaly has been detected by Operations Analytics - Predictive Insights. For example, **Events** ⚠, indicates that the highest status event is ⚠ Warning and that at least one anomaly event is open.
- Click the **Events** tab to see a summary of the total event count, a count of each severity type, and a percentage gauge for the severities. For more information, see "Event Status" on page 1083.

Custom Views

The pages that you create and save are associated with the selected application. For example, the Inventory Management application in the Cloud APM Guided Demo has the following monitoring agents: Linux OS, MySQL, Node.js, Hadoop, and Ruby. You can create and save a custom page at any level of the navigator from application to instance and then open it at the same level where it was created. A page that is created at a particular level can be opened only at the same level. The metrics available for the widgets can be from any of the resources in the application. Using the Inventory Management example, you can create a page with a table from the Ruby agent, a chart from Linux OS agent, and so on.

- The **Custom Views** tab is available at any level of the navigator when you select an application from **All My Applications**.
- After you open the **Custom Views** tab, the **Select a Template for your Custom Page** window is displayed or the default page is displayed if it is already set.
  - In the **Select a Template for your Custom Page** window, you can select a template to create a page.
  - On the default page, you can click + to create a new page.
- On the default page, Click ▾ in the page list and select one of the saved pages from the list.
- The options that you see in the Custom Views tab vary based on whether a page is being edited or viewed. For information about editing a page, see "Creating and managing custom pages" on page 1087. For information about viewing a page, see "Viewing custom pages" on page 1093.

Attribute Details

- The **Attribute Details** tab is displayed after you select a component instance from the navigator **Instances** section (renamed to the selected subgroup name) or by clicking inside a summary group widget.
- If chart or table pages have been saved for the agent, the most recently opened page is displayed with metrics from the selected component instance. Click the title ⌄ to select a different saved page from **My Pages** › or **Shared Pages** ›.
- You can edit the chart or table and click **Preview Results** to render the chart or table with the selected attributes. For more options, see "Creating a custom chart or table page" on page 1068.

Editing the Components dashboard group widgets

You can edit the threshold values of the group widgets that display in the **Components** dashboard (selected from the navigator **Groups** section). You can also control which group widgets are displayed and their position, and decide whether a widget threshold should be included in determining the component status.

About this task

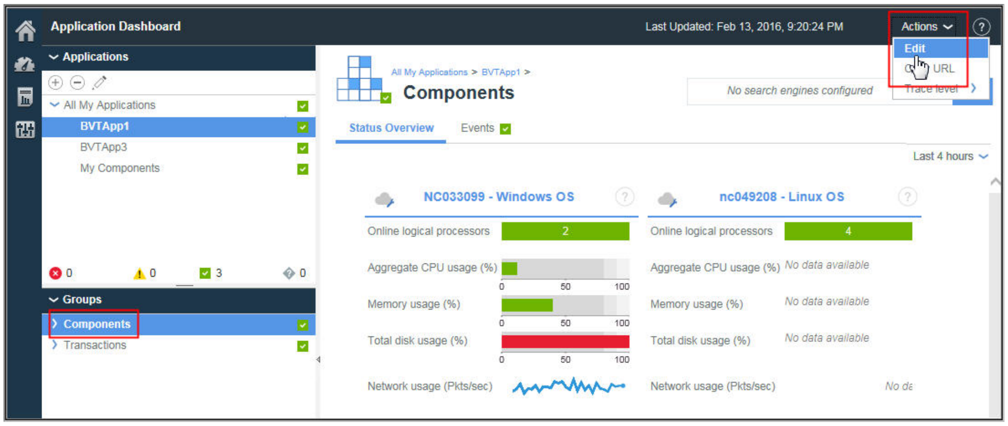
This task involves editing the Components group dashboard and its constituent summary group widget for a defined application. The Components group editor is not available for the **My Components** predefined application. For more information about defined applications, see "Managing applications" on page 1073.

Your user ID must also have the modify permission for Application Performance Dashboard and the create permission for Applications. For more information, see "Roles and permissions" on page 982.

Procedure

1. After you open the Application Performance Dashboard from the **Performance** menu, select the application whose summary group widgets you want to edit from the **All My Applications** dashboard.
2. In the navigator **Groups** section, click **Components** to open a dashboard showing group widgets for all the components in the application.
3. Click **Actions** > **Edit** to open the editor for the group widgets in the **Components** group.

   The **Edit** option shows only when the **Components** dashboard is open.



4. Make any of the following changes to the group widgets:

   - To remove a group widget from view, click ⊖.
   - To modify the summary thresholds for a widget, click ⚙ **Settings**, select the **Thresholds** tab, and change the threshold values for the critical, warning, or normal severity. After editing the thresholds for the group widget, click **Done**.
   - To add a widget, click ⊞, click applications icons until the one that you want is displayed, click inside the group widget to select it, and click **Add**.

- To resize a widget, drag the handle icon ◢. Resizing a widget does not change the size of the text or the height of the widget.
- To move a widget, drag it to a new position.

5. To save your changes and close the editor, click **Save**; or to discard your changes, click **Cancel** .

Results
The **Components** dashboard of the selected application is displayed with the new settings.

What to do next
For more information about the dashboard when a group or subgroup is selected in the navigator, see "Group and Instance - Application Performance Dashboard" on page 1063; for more information about the monitored component dashboard, click the ? button in the **Application Dashboard** banner.

Adjusting and comparing metrics over time

Some of the dashboard charts show metrics that are based on a time range and other charts show only the most recent metrics. When a time selector is displayed in the **Status Overview** tab of a managed system instance, you can adjust the time range for the charts whose values are derived from historical data samples. For attributes that have data collected for multiple days and presented in a line chart, you can compare today's values with a previous day.

Before you begin
If you are comparing with a time range from a previous day, how far back you can go depends on the number of days that the Cloud APM server has saved and the type of data displayed on the page. . For Cloud APM paid subscribers, data samples are stored for 8 days for most resource monitoring data sets. The exact number is published in the agent or data collector attribute help and reference PDF (see Chapter 2, "PDF documentation," on page 45). For Cloud APM trial subscribers, resource monitoring data samples are stored for 2 days. Transaction tracking data from the Response Time Monitoring Agent or middleware agents can only be displayed from the last 24 hours (or for the last 4 hours in some cases) and its retention period cannot be changed.

Procedure

Take these steps to adjust the time range that is displayed in the line chart for a managed resource instance or to compare the values with the same time range from a previous day.

1. If the Application Performance Dashboard is not displayed, select it from the **Performance** menu.
2. Navigate to the dashboard page for an instance that shows historical line charts and click the **Last 4 Hours** time selector.
3. Select one or more of the following options:
   - To change the time range displayed, select **Last 4 hours**, **Last 12 hours**, or **Last 1 day**.
   - To compare the time range displayed in a line chart with the metrics from a different day, select **Compare to** and select an earlier day up to the number of days shown in the pop-up calendar as available (a line is drawn through unavailable dates).
   - To have the time range applied to the dashboards of all defined applications in your monitored environment, select **All Applications**. Otherwise, leave the setting at **Only This Application** to apply the time range to only the current application (such as "My Components"). The **Compare to** selection is effective only for the current page.

Results

- If you are viewing historical data without comparison, all dashboards in the current application (or all applications) are affected by the change.
- If you are viewing a comparison, only line charts in the current page are affected. A line is drawn for each KPI to show the metrics from the chosen day. Some line charts are unavailable for comparison, as

indicated by a watermark on the chart: "No Comparison Available". This can happen with newer managed resources that have not yet collected data for the date specified. Try selecting a more recent date for the comparison.

- Any widgets for which no historical data is collected continue to display the most recent values.
- Data points are distributed along the full length of the chart for the selected time range. Time stamps are displayed on the axis label, starting with the earliest time stamp and ending with the most recent time stamp. Check the oldest data sample to confirm whether a partial range or the full range of historical data is displayed.
- Data sent to the charts and tables is normalized to GMT (Greenwich Mean Time). The **Timestamp** axis prints time stamps based on the time zone of your browser. If your time zone uses Standard Time and Daylight Saving Time, the time stamp displayed during the transition hour is displaced by one hour. Consider, for example, that you are viewing a line chart in Spain and the time changes from 2:00 AM Standard Time to 3:00 AM Daylight Saving Time. The discrepancy between the GMT of the data and the local time of the time stamp results in the time stamps having a gap of one hour from 2:00 AM to 3:00 AM. If you are viewing the same chart in New Zealand during the switch from 3:00 AM Daylight Saving Time to 2:00 AM Standard Time, the time stamps from 2:00 AM to 3:00 AM are repeated.

Viewing and managing custom charts and tables

The Application Performance Dashboard provides predefined dashboards of your managed system key performance indicators. While you are viewing the dashboard for a component instance, use the **Attribute Details** tab to view saved chart or table pages and to create and manage other pages.

For example, you observe a critical indicator in the summary dashboard and drill down to the instance where the condition is occurring. From here, you add a chart that plots the busy CPU rate to see what is happening over time. You can view details about any available attributes of the selected component instance and save the custom chart or table with the agent for display whenever you open a managed system instance.

A subset of the agent's data sets and attributes is available for use in custom charts and tables. These attributes are the most useful for displaying in dashboards. The full set of attributes is available for use in custom thresholds (see “Threshold Manager” on page 967).

To improve performance and reduce redundancy, agents restrict the number of rows that show for certain data sets in the Attribute Details. The data set descriptions in the agent help and reference PDF indicate whether the default data sample limits the number of rows that are sent to the Cloud APM server.

For visually impaired users, the ability to create historical tables provides an alternative to line charts, which assistive technologies such as screen-reader software cannot interpret. For this reason, the **Attribute Details** tab is available for Response Time Monitoring agent and Synthetic Playback agent transaction instances for creating historical tables. For more information, see “Example of creating a custom table with keyboard controls” on page 1071.

Creating a custom chart or table page

While you are viewing the Application Performance Dashboard for a component instance, you can select the **Attribute Details** tab to view saved chart or table pages and to create and manage other pages.

About this task

After you drill down from the Application Performance Dashboard home page to an instance of your managed resource, the **Attribute Details**tab is added to the **Status Overview** and **Events** tabs on the dashboard page.

These instructions are for creating custom charts and tables for component instances. You can follow the steps for Response Time Monitoring agent and Synthetic Playback agent transaction instances, with the following limitations: historical tables only (no charts); you cannot filter the **Data Set** or **Attributes** list; all attributes are selected (you cannot select or deselect individual attributes); you cannot save the page; and the time selector **Previous** option is unavailable for the Synthetic Playback agent.

Procedure

Complete the following steps to construct a chart or table from any of the data sets that are available for the selected component instance:

1. After you open the Application Performance Dashboard from the **Performance** menu, drill down to an instance of the managed resource.

   The selected system is highlighted in the navigator **Instances** section, which is named for the component type, such as **Ruby App**.
2. Click the **Attribute Details** tab.

   If no chart or table pages were saved for this monitoring agent type, the **Real time** **Table** is selected. Real time is appropriate for data sets that return multiple rows and is available only for tables.
3. If a saved chart or table page is displayed, click **New**.
4. Enter a name for the chart or table page in the title field.

   Do not use any of the following characters in the title: ! " % & ' * ? < > } { \.
5. If you prefer to see data samples over time, change the type to **Historical**.

   The **Chart** option is enabled.
6. If you selected ●**Historical** and prefer a chart rendering rather than a table, click **Chart**.
7. From the **Data Set** list, select the radio button for the attribute type that you want to see. If the list is long, use the filter box to reduce the list by entering the text that must be included in the data set name.
   For example, "config" for the Linux OS agent, filters the data sets to show only the **Linux_CPU_Config** and **Linux_OS_Config** data sets.
8. To include an attribute in the chart or table, select the check box next to the name in the **Attributes** list; to include all attributes, select the check box at the beginning of the list. Enter text in the filter box to locate specific attributes, such as "percent".
   For example, "percent" on the KLZ_VM_Stats data set, filters the list to show **Free Virtual Storage (Percent)** and 5 more "Percent"attributes.

   Charts can plot only numeric values; any text or time attributes are disabled.
9. Click **Preview Results** to generate the page with the chosen data set, one table column or chart line grouping for each attribute, and one row or plot point for each data sample.

   You also get a row or plotted line for the aggregate of all the values.
10. To hide chart lines or table rows, take one of the following steps:

Option	Description
Chart	Hide a metric (a line in the chart) by clearing the check box next to the name in the legend. Select a check box to display a metric.
Table	Reduce the number of rows that are displayed by entering the value to filter by in the filter box. You can also create an advanced filter, as described in "Defining a table filter" on page 1072.

11. To adjust the time range, use the time selector:

Option	Description
Real Time	For tables only, refreshes and shows only the latest data sampling.
2 Hours	Plots a point or adds a row for each data sample that was taken in the past two hours, at intervals.
4 Hours	Plots a point or adds a row for each data sample that was taken in the past four hours, at intervals.
12 Hours	Plots a point or adds a row for each data sample that was taken in the past 12 hours, at intervals.

Option	Description
24 Hours	Plots a point or adds a row for each data sample that was taken in the past 24 hours, at intervals.
Previous	Submenu of options to include data from the same time period **Yesterday**, **2 days ago**, or from any day up to **1 week ago**. For example, it's 2:10 PM on August 22 and you set the chart or table to show the past 4 hours. By selecting **Previous** > **1 week ago**, you see data points from 10:10 AM - 2:10 PM today and from 10:10 AM - 2:10 PM on August 15. After you select a previous day, the time selector shows an asterisk (*) such as Last 4 hours* ⌄ and the selected day's data is table is regenerated:

- It is only possible to view data from today and up to one week ago (even if you have increased the maximum historical data retention period to more than 8 days).
- Historical charts are plotted from the oldest data sample to the most recent for the time range selected, for example, last 4 hours. When a previous day is selected, you see the data from the selected time range on the previous day and the time range for today. Any days that are between the earlier date and today show a plot point with a time stamp and no data samples.
- Historical tables are plotted in descending chronological order. When a previous day is selected for the Response Time Monitoring agent, each column is replicated for the previous day with "Previous" in the column heading.
- Regardless of the time range selected, a maximum of 11,000 rows can be displayed. For example, if you chose to display 12 hours from a data set that sends 7,000 rows in 2 hours, fewer than 3 hours of historical data is returned and the oldest data samples are displayed.

12. To save or make other changes to the chart or table, select one of the following options:

Option	Description
Edit	Returns you to the editor for making any of the following changes: • Edit the title • Change to **Real time** or **Historical** data samples • Change to **Chart** or **Table** • Select a different **Data Set** or **Attributes** or both
New	Discards any unsaved changes to the current view and returns you to the selection page for creating a new chart or table.
Cancel	Cancels the editing session for the current chart or table page.
Delete	Deletes the page. **Delete** is available only after a page is saved and any current editing is canceled.
Save for Me	Saves the chart or table page for viewing by your user ID only. No other users can see the saved page.
Save to Share	Saves the chart or table page for viewing by any user ID that is logged in to the Cloud APM console

Views that you save have an opened lock 🔓 next to the title. Views that another user saved that you do not have the authority to edit have a closed lock icon.

Results

After you save the custom chart or table, it is added to the list of saved pages. The next time that you select an instance of the same data source type, such as WebSphere Applications, and select the

Attribute Details tab, the most recently opened saved page is displayed. Click the title ⌄ to select a different saved page from **My Pages** › or **Shared Pages** › .

What to do next

Repeat this procedure to create and manage other chart or table pages.

For guidance on creating a table by using keyboard controls instead of mouse clicks, see "Example of creating a custom table with keyboard controls" on page 1071.

Example of creating a custom table with keyboard controls

Visually impaired users can use the **Attribute Details** dashboard tab to create historical tables as an accessible alternative to historical line charts, which cannot be interpreted by assistive technologies such as screen-reader software.

About this task

The following example illustrates the use of keyboard controls to create a historical table for transactions that are reported by the Synthetic Playback agent. For more information about the agent, see "Managing synthetic transactions and events with Website Monitoring" on page 1006.

As you press the Tab key, the focus moves to the next to field or next section of the application window, from left to right and top to bottom. You can use these steps to generate a transactions table for the Response Time Monitoring Agent by substituting **My Transactions** with an application that includes the agent, or to generate a table for a component instance by substituting **My Transactions** with another application and selecting the **Components** group.

Procedure

Follow these steps to create a table of transactions from the Synthetic Playback agent in the **Attribute Details** tab using the keyboard shortcuts:

1. Log in to IBM Cloud Application Performance Management.

   The focus is on the navigation bar.
2. To open the Application Performance Dashboard, press down-arrow to move the focus to the **Performance** menu, press Enter to select it, press down-arrow to the **Application Performance Dashboard** option, and press Enter again.
3. To open the **Synthetics Transactions** dashboard page, press Tab repeatedly (about 7 times) until the focus moves to the navigator, press down-arrow to focus on the **My Transactions** predefined application, and press Enter.
4. To open the **Transactions Details** dashboard page, press Tab (about 10 times) until the focus moves to the navigator **Instances** section on a **Synthetics Transactions** transaction instance, and press Enter.

   The **Attribute Details** tab is displayed on the dashboard.
5. To open the **Attribute Details** tab, press Tab (about 6 times) until the focus is on the **Status Overview** tab, and press right-arrow until the focus is on the **Attribute Details** tab.

   The **Historical** ▤ **Table** and all attributes of the **Transaction Availability Over Time** data set are selected.
6. To generate the table, press Tab (about 18 times) until the focus is on the **Preview Results** button, and press Enter.

Results

The **Transaction Availability Over Time** attributes are displayed in a table, with column for each attribute and one row for each data sample over the past 4 hours.

What to do next

- To reduce the number of rows that are displayed, you can press Tab to focus on the ▽ **Filter** text box, and enter a partial or entire text or timestamp value to filter by.

- To change the time range, move the focus to the Time Selector **Last 4 hours** drop-down menu and select another option. For more information, see step "11" on page 1069 in "Creating a custom chart or table page" on page 1068.
- To generate a table with the **Transaction Response Time** data set, press Tab (about 2 times) to move the focus to the **New** tool, and press Enter. The selection panel is displayed. Select the **Transaction Response Time** data set and the **Preview Results** button.

Defining a table filter

You can limit the rows in a table that you are viewing in the dashboard **Attribute Details** tab to show only rows of a certain type, or that have specific by text or timestamp attribute values. Although numeric values are not available for filtering, such as percentages, some numeric attribute values are converted to a display value for the table and treated as text.You can apply a quick filter or open an editor to compose an advanced filter.

Procedure

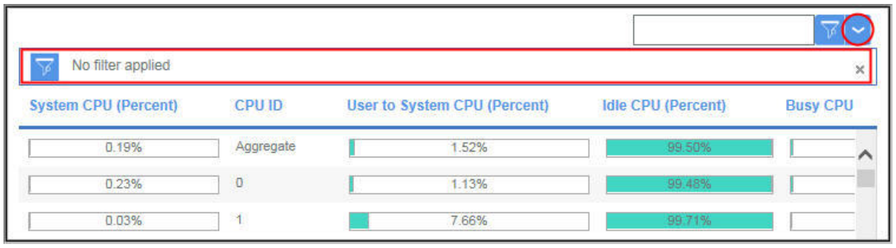
Complete these steps to filter a custom table by text or timestamp attribute values. Although numeric values are not available for filtering, such as percentages, some numeric attribute values are converted to a display value for the table and treated as text.

1. After you open the Application Performance Dashboard from the **Performance** menu, drill down to an instance of the managed resource.
2. Click the **Attribute Details** tab.

   The most recently saved page is displayed or, if no pages have been saved, the **Data Set** and **Attributes** selection lists are displayed.
3. If a saved table page is displayed, continue to step "5" on page 1072, select another saved table page from the drop-down menu, or click **Add** to create a new table.
4. If you are creating a new table or editing a saved table, select the **Data Set** and **Attributes** to use, and click **Preview Results**.
5. For a quick filter, click in the **Filter** text box and type the partial or full text to filter by.

   As you type, any rows that do not contain what you typed are removed from the table. To remove the quick filter, delete the value or click the "x".
6. For an advanced filter, click the drop-down menu and select **Build Filter** or click anywhere in the filter bar.

   The Build Filter window opens with any rules that were defined.



System CPU (Percent)	CPU ID	User to System CPU (Percent)	Idle CPU (Percent)	Busy CPU
0.19%	Aggregate	1.52%	99.50%	
0.23%	0	1.13%	99.48%	
0.03%	1	7.66%	99.71%	

7. To define a rule, complete the fields:
   a) Leave the column setting at "Any Column" or select the attribute to filter by from the list.
   b) Leave the condition at "contains" or select another operator from the list and enter the text or timestamp value to filter by in the text box:

Condition	Row is included in the table when...
contains	the filter value is found somewhere in the cell.
equals	the cell value matches the filter value exactly, including letter casing.
starts with	the cell value begins with the same characters as the filter value.

Condition	Row is included in the table when...
ends with	the cell value has the same characters at the end as the filter value.
does not equal	the cell value is not an exact match of the filter value.
does not contain	the cell value does not include the same text or number as the filter value.
does not start with	the cell value does not begin with the same characters as the filter value.
does not end with	the cell value does not end with the same characters as the filter value.
is empty	the cell shows no data.

c) After you complete the rule, click **Filter** to see the results, click ⊕ **Add Filter Rule** to add another rule, or go to the next step.

8. If the filter has multiple rules, take any of these steps:

   - **Match** is initially set to **All rules**, which means that a row is displayed only if the data in the row follows all the rules in the filter. The row is excluded if no text or timestamp values follow any one rule. If you have multiple rules and you want a row included if it follows any of the rules, change the setting to **Any rule**.
   - To edit a rule, change any of the field values.
   - To delete a rule, select it and click ⊖ **Remove Rule**.

9. When you are finished defining a rule (or rules), click **Filter** to close the dialog box and apply the filter.

Results

The groups that do not meet the filter criteria are removed from the display and the filter bar reports the number of items, for example, "480 of 1200 items shown".

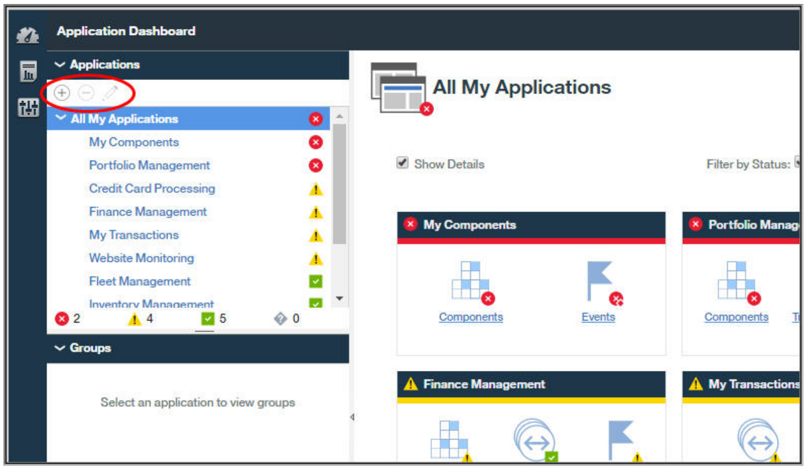
What to do next

- Hover the mouse pointer over the filter bar to open a pop-up window with the filter criteria. You can delete a rule (click ×) or click inside the window to edit the filter criteria.
- Click **Clear filter** in the filter bar or **Clear** in the **Build Filter** window to remove the filter and display all rows.

Managing applications

Use the tools that are available in the Application Performance Dashboard to organize your managed resources into applications.

The navigator **Applications** tools open the Applications editor for creating or editing applications and applying the managed resources that are available.



The *My Components* application is a predefined application that includes the managed systems that were discovered by the Cloud APM server. **My Components** cannot be edited or deleted.

The *My Transactions* application is a predefined application that includes synthetic transaction data. My Transactions cannot be edited or deleted.

For a video demonstration about adding an application, watch Application Performance Management - Define Application.

For a scenario about creating an application for monitoring the IBM Java application stack, see “Adding web applications to the Application Performance Dashboard ” on page 92 and “Associating the IBM Java application stack with the web application ” on page 93.

Restriction: You must have modify permission for Applications to use the Add application tool. You must have modify permission for Applications or the specific application to use the Remove and Edit tools. For more information, see “Working with roles, users, and permissions” on page 990.

Adding an application

Use the Applications editor to create a new application and apply the managed resources that are available, or select one from any discovered applications.

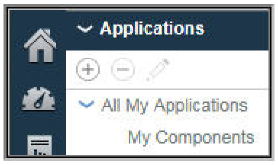
Before you begin

You must have modify permission for Applications to use the Add application tool. For more information, see “Working with roles, users, and permissions” on page 990.

Procedure

Complete the following steps in the Cloud APM console to add an application to the Application Performance Dashboard.

1. If the Application Performance Dashboard is not displayed, select it from the **Performance** menu or, if you are on another console page, click the **Home** link.
2. In the **Applications** section of the navigator, click ⊕. The **Add Application** window is displayed.

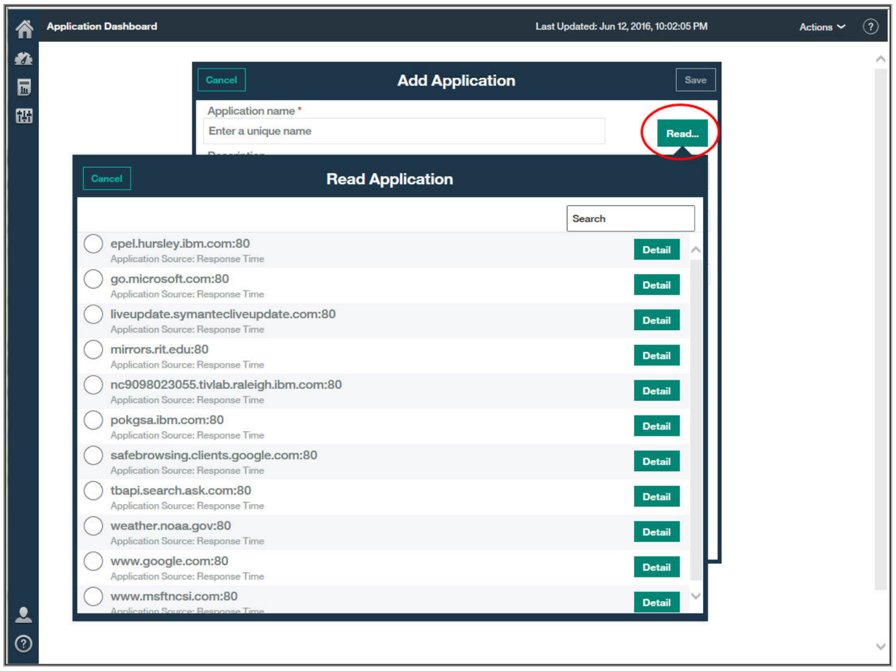
3. Enter a name for your application in the **Application name** field and, optionally, a description in the **Description** field.

   Do not use the `! " % & ' * ? < > } { \` symbols in the name or description.

   You can see some examples of application names, such as "Finance Management" and "Credit Card Processing" in the Guided Demo.

4. Click **Read** to open the **Read Application** window with a list of any discovered applications, and take one or more of the following steps:

   - Click **Detail** to see the components of an application.
   - Select the application that you want to use, and click **Save**. The **Read Application** window closes, the source repository is displayed in the **Application read from** field, and the components are listed in **Application components**.
   - Click **Cancel** to close the window without making a choice.



○ epel.hursley.ibm.com:80 Application Source: Response Time	Detail
○ go.microsoft.com:80 Application Source: Response Time	Detail
○ liveupdate.symantecliveupdate.com:80 Application Source: Response Time	Detail
○ mirrors.rit.edu:80 Application Source: Response Time	Detail
○ nc9098023055.tivlab.raleigh.ibm.com:80 Application Source: Response Time	Detail
○ pokgsa.ibm.com:80 Application Source: Response Time	Detail
○ safebrowsing.clients.google.com:80 Application Source: Response Time	Detail
○ tbapi.search.ask.com:80 Application Source: Response Time	Detail
○ weather.noaa.gov:80 Application Source: Response Time	Detail
○ www.google.com:80 Application Source: Response Time	Detail
○ www.msftncsi.com:80 Application Source: Response Time	Detail

5. In the **Template** field, keep the **Custom Application** template or select a different template with the › button, and click **Save**.

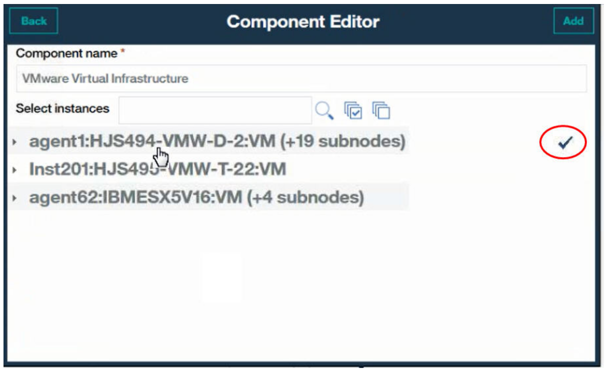
   Any associated component types and instances are shown in the **Application components** list.

6. Click ⊕ **Add components** and, in the **Select Component** window that opens, select a component from the list.

   The **Component Editor** is displayed.

7. To find and select agent node or subnode instances (or both) for the application, take one or more of the following steps:

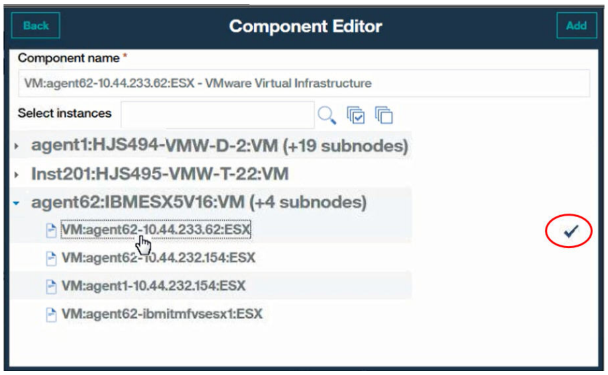
   - Click an instance to select it.
   - For agent nodes that have subnodes, select the node alone by clicking the name while the tree is collapsed, select the node and all subnodes by expanding the node tree (click ▸) and clicking the node, or select individual subnodes by expanding the node tree and clicking the instance.
   - Use the 🔍 ☑ ☐ toolbar to search for instances that contain the text in the search text box, to select all instances, or to clear all instances.
   - If you want to change the display name in the navigator, edit the component name.



Component name *

VMware Virtual Infrastructure

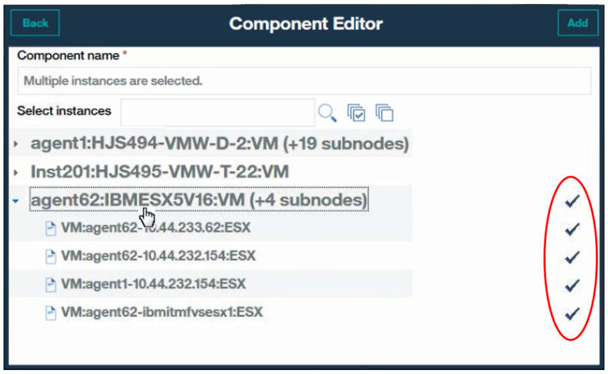
- agent1:HJS494-VMW-D-2:VM (+19 subnodes) ✓
- Inst201:HJS495-VMW-T-22:VM
- agent62:IBMESX5V16:VM (+4 subnodes)



Component name *

VM:agent62-10.44.233.62:ESX - VMware Virtual Infrastructure

- agent1:HJS494-VMW-D-2:VM (+19 subnodes)
- Inst201:HJS495-VMW-T-22:VM
- agent62:IBMESX5V16:VM (+4 subnodes)
  - VM:agent62-10.44.233.62:ESX ✓
  - VM:agent62-10.44.232.154:ESX
  - VM:agent1-10.44.232.154:ESX
  - VM:agent62-ibmitmfvsesx1:ESX



Component name *

Multiple instances are selected.

- agent1:HJS494-VMW-D-2:VM (+19 subnodes)
- Inst201:HJS495-VMW-T-22:VM
- agent62:IBMESX5V16:VM (+4 subnodes) ✓
  - VM:agent62-10.44.233.62:ESX ✓
  - VM:agent62-10.44.232.154:ESX ✓
  - VM:agent1-10.44.232.154:ESX ✓
  - VM:agent62-ibmitmfvsesx1:ESX ✓

V6-V7 If you are adding a Tivoli Monitoring agent instance and don't see it in the list of available instances, check that the Tivoli Enterprise Portal Server that is associated with the Hybrid Gateway is at a supported version (see Hybrid Gateway supported agents (APM Developer Center)).

8. Click **Add** to add the selected agent nodes and subnodes to the application, and click **Back**.

   The Application components list is updated with the new component names.
9. Select another component to add instances to and repeat “6” on page 1075, “7” on page 1075, and “8” on page 1076 or click **Close**.
10. If other instances are related to the components in the **Application components** list, a button that shows the number of related instances is displayed and you can take the following steps:

    a) Click the 1 button to see the related instances in the **Updated Details** window.
    A bar is shown for each type of update, including the instance name. For example, if one of the components was removed, it shows beneath the **Deleted** components bar.

    b) Select one or more instances and click **Save** to update the Application resources list.
11. When you are finished defining the application, close the application editor by clicking **Save** to save your changes or click **Cancel** to undo the changes.

Results

Your application updates are completed by the Cloud APM server after you save your changes. It might take a few minutes before your changes appear in the dashboard. (Try clearing the browser cache if it takes a long time for your changes to display.) The new application is displayed in the Application Performance Dashboard and the navigator **Applications** section. When the application is selected, the components are displayed in the **Groups** section.

Editing an application

Use the Applications editor to modify a defined application to add or remove managed resources as components of the application.

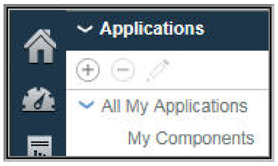
Before you begin

You must have Modify permission for Applications or the specific application to use the Edit tool. For more information, see “Roles and permissions” on page 982.

Procedure

Complete the following steps in the Cloud APM console to edit an application.

1. If the Application Performance Dashboard is not displayed, select it from the **Performance** menu or, if you are on another console page, click the **Home** link.
2. Select the application that you want to edit from the **All My Applications** list in the navigator and click ✎.

   The **Edit Application** window is displayed.
3. Optional: Edit the **Application name** or **Description**.

   Do not use the `! " % & ' * ? < > } { \` symbols in the name or description.
   If your Applications View or Modify permissions are for individual applications and not all applications, you might not be able to see the application in the dashboard or modify the application after it is renamed. This limitation is because the renamed application is treated as a new application. Your role administrator or monitoring administrator must give you View or Modify permission for the renamed application.
4. To add components and instances to the application, take the following steps.

   a) Click ⊕ and, select a component from the list in the window that opens.

      The **Component Editor** is displayed.

   b) Select agent node or subnode instances (or both) for the application:

      - Click an instance to select it.
      - For nodes that have subnodes, select the node by clicking the name while the tree is collapsed, select the node and all subnodes by expanding the node tree (click ▸) and clicking the node, or select individual subnodes by expanding the node tree and clicking the instance.
      - Use the 🔍 ☑ ❐ toolbar to search for instances that contain the text in the search text box, to select all instances, or to clear all instances.
      - If you want to change the display name in the navigator, edit the component name.

   c) Click **Add** to add the instance or instances, and click **Back**.

      The Application components list is updated with the new component names.

   d) You can select another component to add instances to, or click **Close**.

   The Application components list is updated with the new component names. A number in parentheses after the name indicates how many instances are associated with the component.
5. To edit a component name or change the instance that is associated with it, select the component from the **Application components** list and click ✎:

   a) To associate a different instance with the component, search for and select the instance that you want.

   b) To change the component name that is used as the display name in the navigator for this application, edit the **Component name** field.

   c) Click **Save**.

   The Application components list is updated with the changes that you made.
6. To remove a component or instance from the application, select it and click ⊖. Click **OK** to confirm that you want to remove it.
7. If other instances are related to the components in the **Application components** list, a button that shows the number of related instances is displayed and you can take the following steps:

   a) Click the ❶ button to see the related instances in the **Updated Details** window.

A bar is shown for each type of update, including the instance name. For example, if one of the components was removed, it shows beneath the **Deleted** components bar.

b) Select one or more instances and click **Save** to update the Application resources list.

8. After you are finished editing the application, close the application editor by clicking **Save** to save your changes, or **Cancel** to undo the changes.

Results
Your application updates are completed by the Cloud APM server after you save your changes. It might take a few minutes before your changes appear in the dashboard.
Related reference
"Roles and permissions" on page 982

Deleting an application

When you no longer need an application that you have defined for display in the Application Performance Dashboard, you can delete it. Deleting an application does not uninstall the supporting components; only the application that they are contained in. The same components are available for adding to other applications and are not removed from other applications that they belong to.

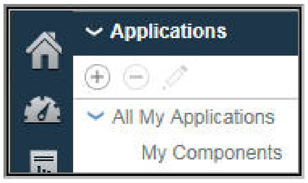
Before you begin

You must have modify permission for Applications or the specific application to use the Remove tool. For more information, see "Working with roles, users, and permissions" on page 990.

Procedure

Complete the following steps to remove an application from the Application Performance Dashboard.

1. If the Application Performance Dashboard is not displayed, select it from the **Performance** menu or, if you are on another console page, click the **Home** link.



2. In the **Applications** section of the navigator, select the application that you want to delete from the **All My Applications** list and click ⊖.

   A message asks you to confirm.
3. Click **Yes** to confirm that you want to delete the application; or **No** if you are not sure.

Results
After you click **Yes**, the application is deleted from the Application Performance Dashboard.

What to do next
Repeat this step for any other applications that you want to delete.

Viewing and removing offline agents

After an agent has been offline for four days, it is removed from the Cloud APM console. Review how offline agents are indicated and the effect on resource groups, topology views, and other features. Use the application editor to remove a managed system from the dashboard before the four days have passed. .

About this task
After agents are installed on the systems that you want to manage, they connect to the Cloud APM server and send data samples to the Application Performance Dashboard for presentation and threshold

evaluation. If the agent is offline, the ◆ status indicator is displayed in the navigator and in the dashboard. The server waits for a specific number of intervals to pass with no response from the agent before showing that the agent is unavailable. See "Examples of offline agents" on page 1080.

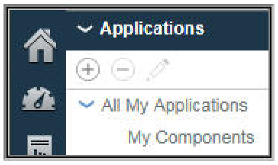
After four days, the offline agent is removed from the user interface with the following exception: If the agent is one that supports transaction tracking, the offline agent continues to show in the Aggregate Transaction Topology and Transaction Instance Topology views.

You can remove the offline agent from any defined applications, which removes it from the Cloud APM console before the four-day waiting period is complete.

Procedure

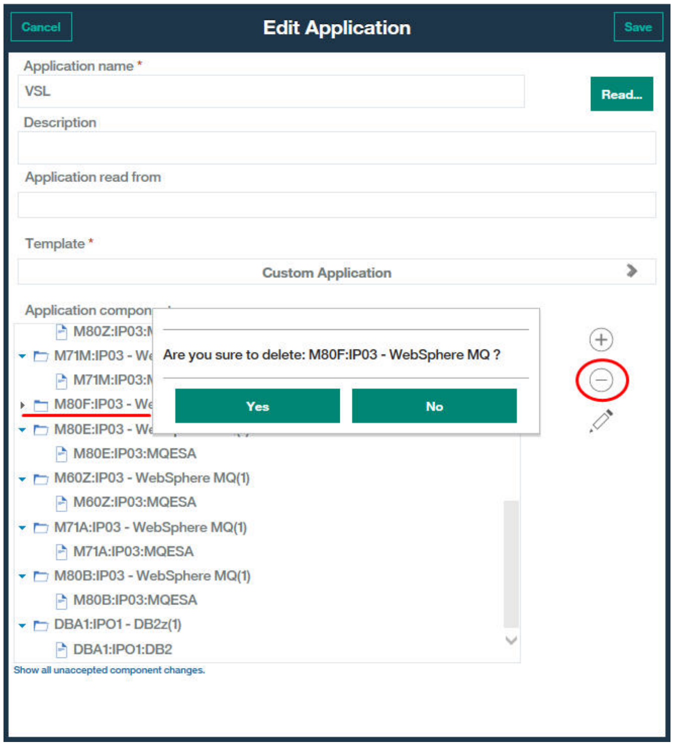
Take these steps to remove an offline agent from a defined application:

1. If the Application Performance Dashboard is not displayed, select it from the **Performance** menu or, if you are on another console page, click the **Home** link.
2. In the **Applications** section of the navigator, select the application that the offline agent is a component of and click ✎ **Edit Application**.



3. Select the agent or agent subnode in the Application Components list and click ⊖.

   For agents that have subnodes, select the agent alone by clicking the name while the tree is collapsed, select the node and all subnodes by expanding the node tree (click ▸) and clicking the node, or select individual subnodes by expanding the node tree and clicking the instance.



Application name * VSL

Description

Application read from

Template * Custom Application

Application compon

M80Z:IP03:M
M71M:IP03 - We
M71M:IP03:M
M80F:IP03 - We
M80E:IP03 - We
M80E:IP03:MQESA
M60Z:IP03 - WebSphere MQ(1)
M60Z:IP03:MQESA
M71A:IP03 - WebSphere MQ(1)
M71A:IP03:MQESA
M80B:IP03 - WebSphere MQ(1)
M80B:IP03:MQESA
DBA1:IPO1 - DB2z(1)
DBA1:IPO1:DB2

Are you sure to delete: M80F:IP03 - WebSphere MQ ?

Show all unaccepted component changes.

4. After you are finished editing the application to remove the offline agent or subnode, click **Save**.

Results

Your application updates are completed by the Cloud APM server after you save your changes. It might take a few minutes before the offline agent is removed from the Application Performance Dashboard.

Examples of offline agents

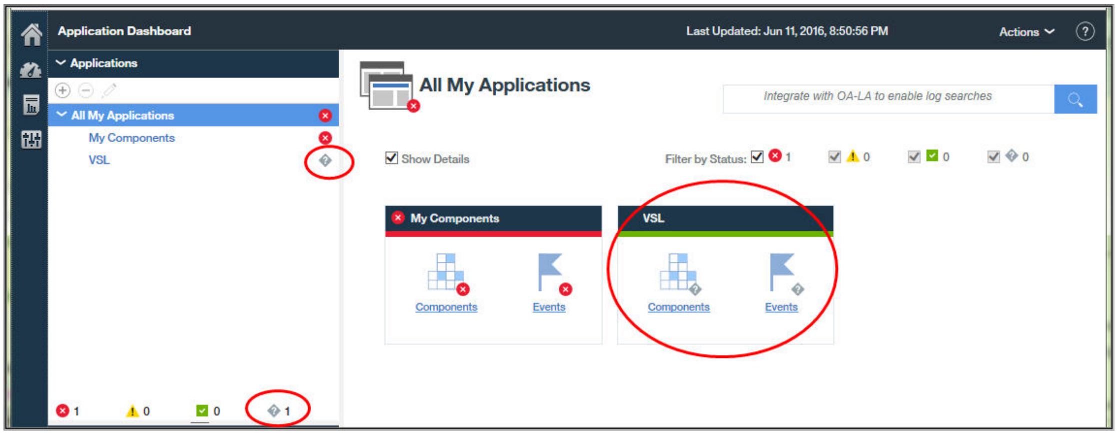
Review the examples of how offline agents are displayed in the Cloud APM console. You can remove the display for any offline agents that you no longer want to monitor. If the agent comes back online later, monitoring resumes.

When an agent is offline, no data is sent to the Cloud APM console and the Application Performance Dashboard displays a ◈ status indicator for the agent and the applications it belongs to. The agent is unavailable for adding to a defined application in the Application editor or to a custom group in the Resource Group Manager, or for creating tables and historical line charts in the Attribute Details tab.

Application Performance Dashboard - All My Applications

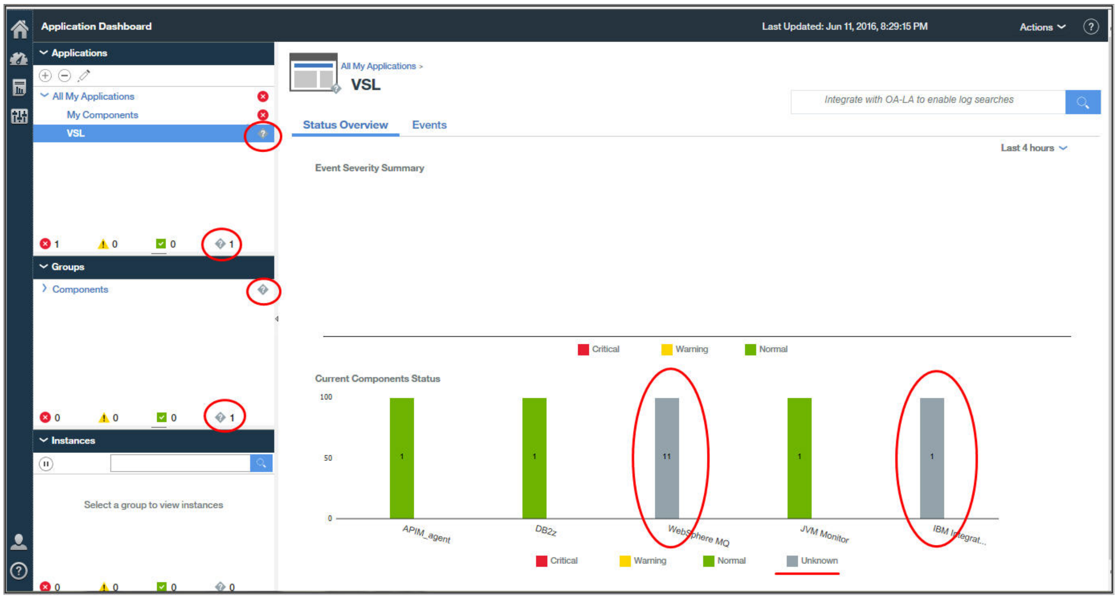
The home dashboard page, **All My Applications**, gives the first indication of offline status. The counter in the navigator Applications section shows the number of applications with unavailable resources.

The summary box shows Normal event status because no events are open for any of the application's managed resources.



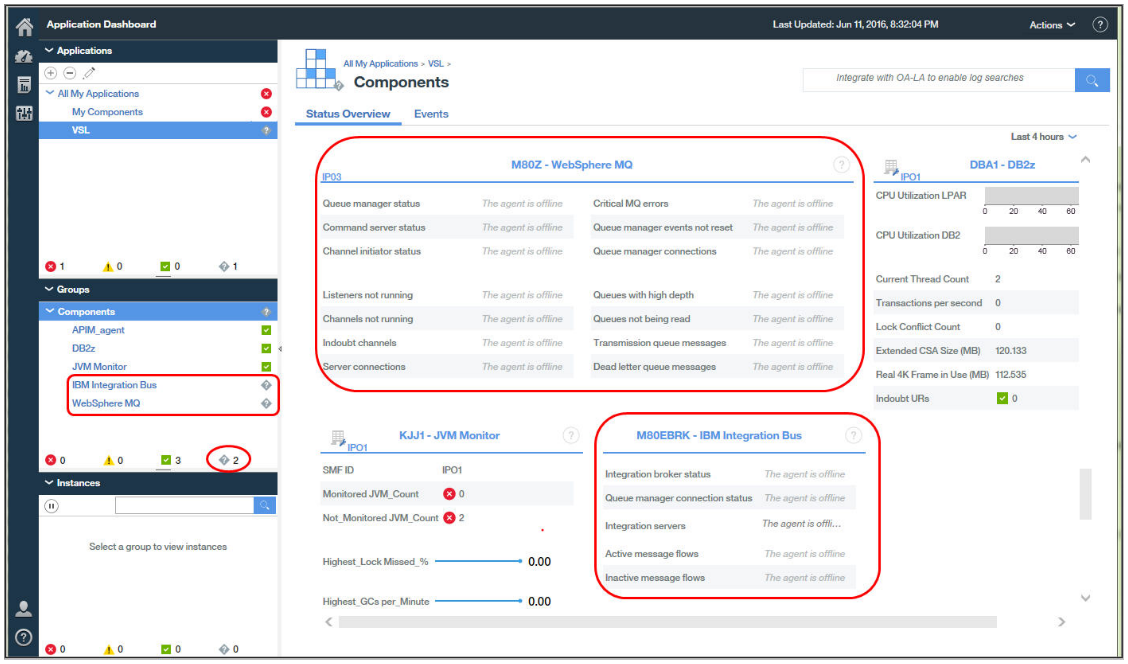
Application Performance Dashboard - Application

After the user clicks the summary box title bar or selects the application from the navigator, the **Status Overview** tab is displayed with an empty **Event Severity Summary** chart. The **Current Component Status** bar chart shows the status for offline agents as "Unknown".



Application Performance Dashboard - Group

After the user clicks inside the **Current Component Status** chart or the navigator **Components** group, the **Status Overview** tab changes to show a summary group widget for each managed resource. The group widgets for the unavailable agents display a message that the agent is offline instead of KPIs.



M80Z - WebSphere MQ

IP03

Queue manager status	The agent is offline	Critical MQ errors	The agent is offline
Command server status	The agent is offline	Queue manager events not reset	The agent is offline
Channel initiator status	The agent is offline	Queue manager connections	The agent is offline
Listeners not running	The agent is offline	Queues with high depth	The agent is offline
Channels not running	The agent is offline	Queues not being read	The agent is offline
Indoubt channels	The agent is offline	Transmission queue messages	The agent is offline
Server connections	The agent is offline	Dead letter queue messages	The agent is offline

DBA1 - DB2z

IPO1

CPU Utilization LPAR

CPU Utilization DB2

Current Thread Count	2
Transactions per second	0
Lock Conflict Count	0
Extended CSA Size (MB)	120.133
Real 4K Frame in Use (MB)	112.535
Indoubt URs	0

KJJ1 - JVM Monitor

IPO1

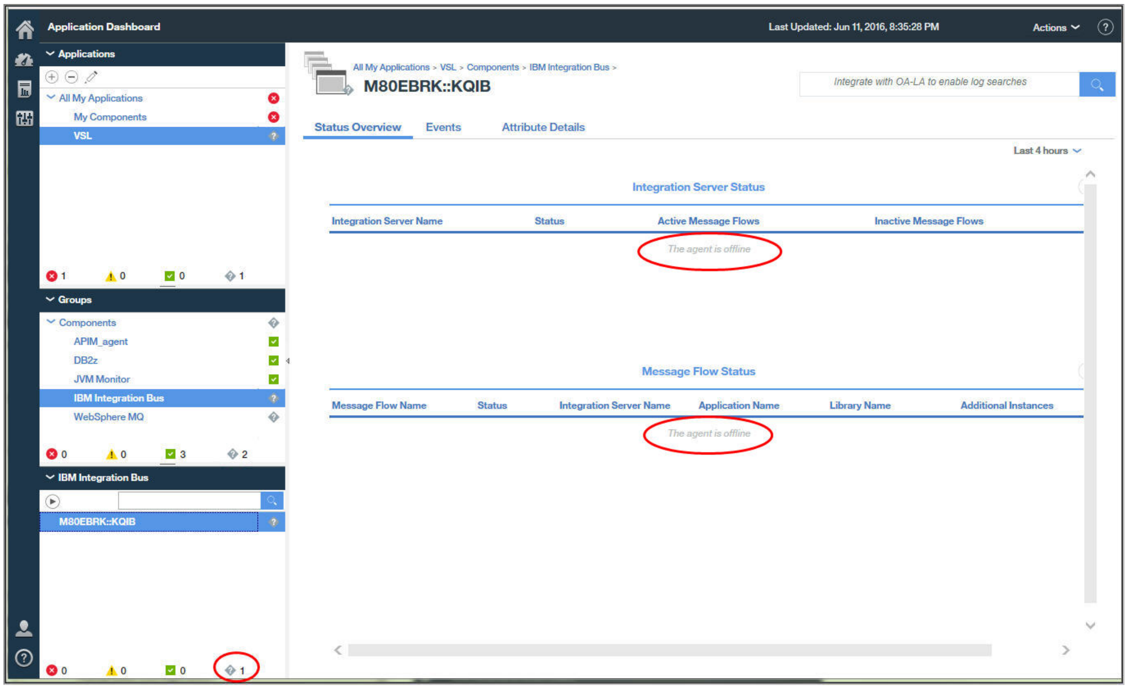
SMF ID	IPO1
Monitored JVM_Count	0
Not_Monitored JVM_Count	2
Highest_Lock Missed_%	0.00
Highest_GCs per_Minute	0.00

M80EBRK - IBM Integration Bus

Integration broker status	The agent is offline
Queue manager connection status	The agent is offline
Integration servers	The agent is offli...
Active message flows	The agent is offline
Inactive message flows	The agent is offline

Application Performance Dashboard - Instance

After the user clicks inside one of the offline agent summary group widgets, the **Status Overview** tab shows chart and table widgets for the selected agent. But, as for the summary group widget, only a message that the agent is offline is displayed instead of KPIs for the agent instance.



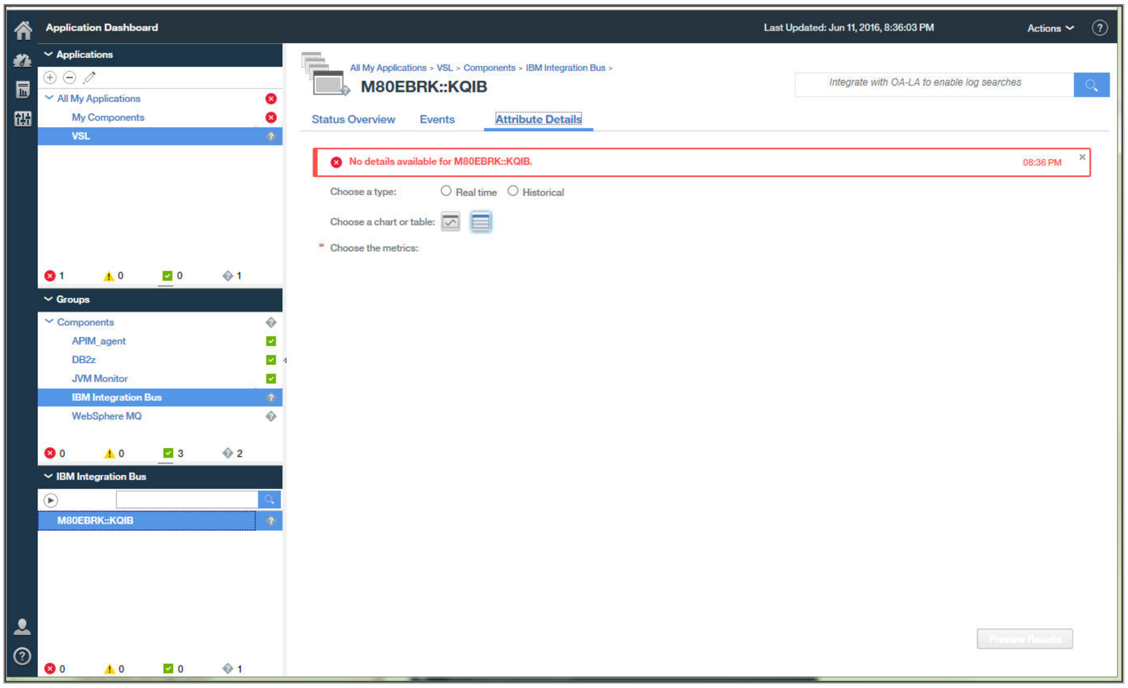
Integration Server Status

Integration Server Name	Status	Active Message Flows	Inactive Message Flows
		The agent is offline	

Message Flow Status

Message Flow Name	Status	Integration Server Name	Application Name	Library Name	Additional Instances
			The agent is offline		

After the user clicks the **Attribute Details** tab, a message says that no details are available for the agent instance. It's not possible to create a custom chart or table for the offline agent instance.



M80EBRK::KQIB

No details available for M80EBRK::KQIB. 08:36 PM

Choose a type: ○ Real time ○ Historical

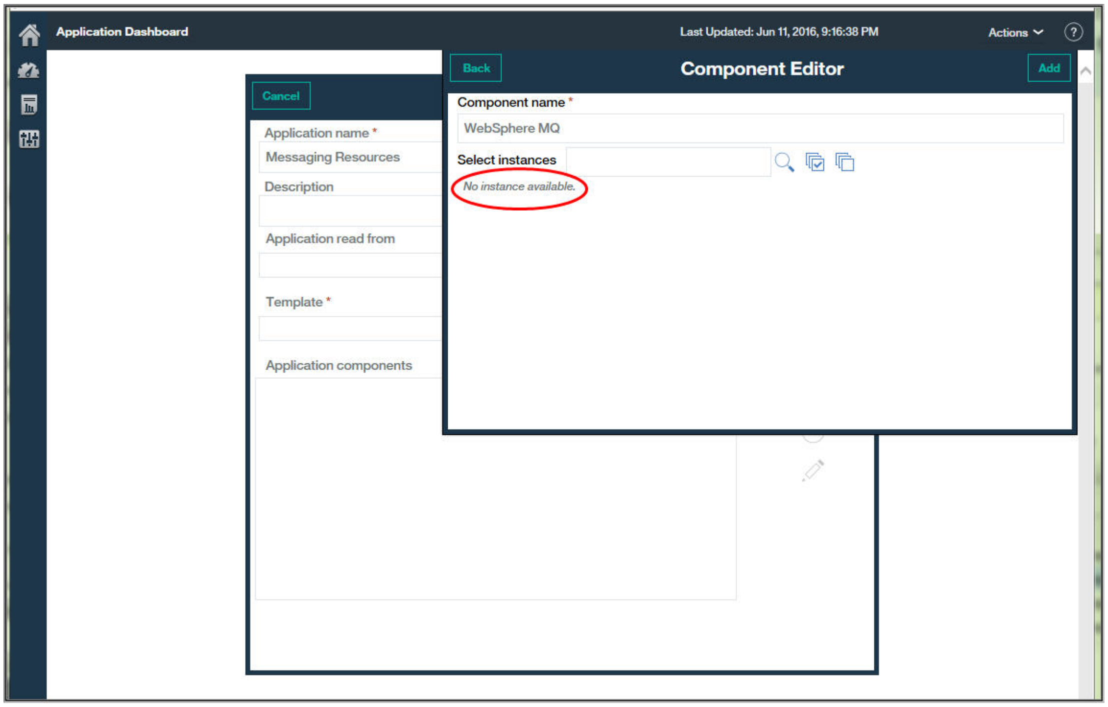
Choose a chart or table:

* Choose the metrics:

Application editor

In the Application Performance Dashboard, after the user clicks the navigator Applications ⊕ **Add Application** or ✎ **Edit Application** tool, the Application editor window pops up.

After the user clicks ⊕ **Add components** and selects an agent type, if no agents of that type are installed or are offline, a message informs you that no agent instances are available. If other agent instances are available, they appear in the list.



Component Editor

Component name *
WebSphere MQ

Select instances

No instance available.

Application name *
Messaging Resources

Description

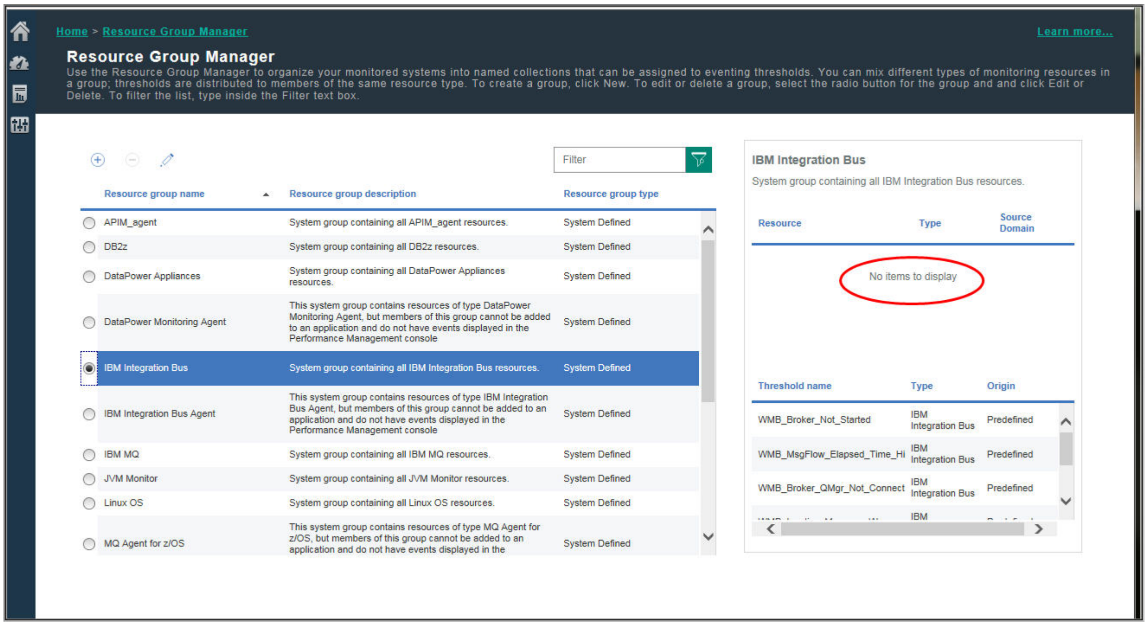
Application read from

Template *

Application components

Resource Group Manager

After the user selects **System Configuration** > **Resource Group Manager**, the page opens with a table of resource groups. As you select a group, its constituent agent instances are listed along with the assigned thresholds. If all agents instances are offline, a message says that no instances are available.



Resource Group Manager

Use the Resource Group Manager to organize your monitored systems into named collections that can be assigned to eventing thresholds. You can mix different types of monitoring resources in a group; thresholds are distributed to members of the same resource type. To create a group, click New. To edit or delete a group, select the radio button for the group and and click Edit or Delete. To filter the list, type inside the Filter text box.

	Resource group name	Resource group description	Resource group type
○	APIM_agent	System group containing all APIM_agent resources.	System Defined
○	DB2z	System group containing all DB2z resources.	System Defined
○	DataPower Appliances	System group containing all DataPower Appliances resources.	System Defined
○	DataPower Monitoring Agent	This system group contains resources of type DataPower Monitoring Agent, but members of this group cannot be added to an application and do not have events displayed in the Performance Management console	System Defined
●	IBM Integration Bus	System group containing all IBM Integration Bus resources.	System Defined
○	IBM Integration Bus Agent	This system group contains resources of type IBM Integration Bus Agent, but members of this group cannot be added to an application and do not have events displayed in the Performance Management console	System Defined
○	IBM MQ	System group containing all IBM MQ resources.	System Defined
○	JVM Monitor	System group containing all JVM Monitor resources.	System Defined
○	Linux OS	System group containing all Linux OS resources.	System Defined
○	MQ Agent for z/OS	This system group contains resources of type MQ Agent for z/OS, but members of this group cannot be added to an application and do not have events displayed in the	System Defined

IBM Integration Bus

System group containing all IBM Integration Bus resources.

Resource	Type	Source Domain
No items to display		

Threshold name	Type	Origin
WMB_Broker_Not_Started	IBM Integration Bus	Predefined
WMB_MsgFlow_Elapsed_Time_Hi	IBM Integration Bus	Predefined
WMB_Broker_QMgr_Not_Connect	IBM Integration Bus	Predefined
	IBM	

Related concepts

"Managing applications" on page 1073

Use the tools that are available in the Application Performance Dashboard to organize your managed resources into applications.

"Using the dashboards" on page 1055

Related reference

"Resource Group Manager" on page 962

Your monitored environment might have multiple managed systems that can be categorized by their purpose. Such systems often have the same threshold requirements. Use the **Resource Group Manager** to organize managed systems into groups that you can assign thresholds to. You can also create resource groups that correlate with your role-based access control (RBAC) policies.

Event Status

Use the **Event Status** to get a summary overview of open events for the selected navigator item and to respond to events with a critical or warning status by drilling down to detailed dashboards.

The status indicators are for events from the thresholds that are running on your managed systems. If you have Hybrid Gateways configured, the events can also be from situations that are running on the managed systems in your IBM Tivoli Monitoring environment. If your configuration includes IBM Operations Analytics - Predictive Insights, any detected anomalies are also displayed.

Events for some thresholds do not display in the Application Performance Dashboard. The thresholds use attributes for resources that are not published, which can occur in agents that support subnodes. (For a description of subnodes, see the Agent Builder topic,).

Critical, Warning, Normal

- The status indicators consolidate the event severities from the thresholds:

  ⊗ Critical status indicates all events with a Fatal or Critical severity

  ⚠ Warning status indicates all events with a Minor or Warning severity

  ✓ Normal status indicates all events with an Unknown severity

  ◈ Unknown status indicates that the managed system is offline. After 4 days offline, the managed system is removed from any applications and no longer is displayed in the dashboards.
  To check the status, stop, or start an agent, see "Using agent commands" on page 177

- V6-V7 When one or more Hybrid Gateways are configured, the status indicators for events from Tivoli Monitoring situations are the same as for thresholds except that Normal status indicates events with ✓ Harmless, ℹ Informational, or ◆ Unknown severity.
- When your managed environment includes IBM Operations Analytics - Predictive Insights, any detected anomalies are indicated by a diamond-shaped icon over the status indicator, such as ⚠. For more information, see "Investigating anomalies with Operations Analytics - Predictive Insights" on page 1085.

Event Severity Summary percentage gauge

- The Event Severity Summary gauge shows the Critical, Warning, and Normal event status percentages. For example, 50.00% 50.00% shows that 50% of events are from thresholds with a Minor or Warning severity and 50% are from thresholds with a Fatal or Critical severity.
- Also reported is the total number of events and how many for each status level.
- The event count includes any anomalies from Operations Analytics - Predictive Insights. For example, a total of "8 including 1 anomaly" means that there are 7 threshold events and 1 anomaly event.

Events table

- The table of open events and status is defined by the selected navigator item: application, group, subgroup, or instance.
- Events are sorted by the **Severity** column, with the highest severity shown first. Click a column heading to change the sort order.
- Each row provides the following information about the event:

  Threshold Name

  The name that was given to the threshold.

  V6-V7 The name that was given to the situation.

  Status

  The status of the event, such as **Open**.

  Severity

  The severity value of the event: ✖ Critical (applies to Fatal and Critical threshold severities), ⚠ Warning (applies to Minor and Warning threshold severities), or ✓ Normal (applies to Unknown threshold severities; for Tivoli Monitoring events, applies to Harmless, Informational, and Unknown severities).

  ◆ Unknown status indicates that the managed system is offline. After 4 days offline, the managed system is removed from any applications and no longer is displayed in the dashboards. (To check status, stop, and start an agent, see "Using agent commands" on page 177.)

  When your managed environment includes IBM Operations Analytics - Predictive Insights, analytics applied to the historical data might detect an anomaly and open an event. An event opened for a detected anomaly is indicated by an icon overlaying the status indicator, such as ✖. Click the **View anomaly analysis** link to open the Predictive Insights **Service Diagnosis** view in a new browser tab or window. Use the **Service Diagnosis** view to review the anomalous behavior in the components that support the application.

  Display Item

  Applies to multiple-row data sets only. The display item is a key attribute that was selected for the threshold to distinguish multiple events from one another that were opened for the same managed system.

  Source

  The system host name or other name that is derived from the monitoring agent that identifies the source of the event.

Timestamp

The date and time when the event occurred or the condition was observed by the originating agent, expressed in the time zone of the Cloud APM console user.

If an agent is restarted or threshold definitions are modified for an agent, then the agent's sampled events are closed and reopened if the threshold condition is still true. In these scenarios, the `Timestamp` value is updated to the time when the originating agent reopened the event.

For pure events, a new event is opened by the agent and replaces the previous event instance each time the originating agent determines that the threshold condition is true. A pure event remains open for 24 hours (or a configurable number of hours) after the last time the threshold condition evaluated to true. Only the latest instance of a pure event is displayed on the Cloud APM console.

Description

The description, if any, that was written for the threshold.

- Click a row to expand the details about the event:

Node

The managed system name of the node instance.

For agents with subnodes, the **`Enable Subnode Events`** option controls whether subnodes are shown. For more information, see "UI Integration" on page 1050.

Threshold ID

The threshold identifier.

Global Timestamp

The date and time when the event was received from the originating agent by the Cloud APM server, expressed in the time zone of the Cloud APM console user.

Type

Whether the event is pure or sampled. Pure events are unsolicited notifications. Thresholds for pure events have no sampling interval or constant metric that can be monitored for current values.

Description

The description, if any, that was written for the threshold.

Formula

The formula as it is written in the Threshold Editor. For example, `Percent Failed > 10.000 AND Transaction Definition Name != 'Ignore_Resources'`.

If the EIF Slot Customization function was used to customize the value of the **`msg`** base slot, the customized **`msg`** slot value is displayed instead of the threshold formula. For more information, see "Forward EIF Event?" on page 968 in the Threshold Manager topic and "Customizing an event to forward to an EIF receiver" on page 972.

You can select and expand other rows, or click again to collapse a row. While a row is expanded, you can drill down to the dashboards for the managed system that you can use to help determine the cause of the event.

Investigating anomalies with Operations Analytics - Predictive Insights

IBM Cloud Application Performance Management only: When your managed environment includes IBM Operations Analytics - Predictive Insights, analytics applied to the historical data can detect anomalies and open events. Use the Application Performance Dashboard to locate and view anomalies that were detected by Operations Analytics - Predictive Insights.

Before you begin

Operations Analytics - Predictive Insights must be integrated with your Cloud APM environment for you to be alerted of anomalies in the Application Performance Dashboard. For more information, see "Integrating with Operations Analytics - Predictive Insights" on page 952.

About this task

The Application Performance Dashboard shows the status summary of the applications in your domains and their component managed systems. Event status indicators in the **All My Applications** dashboard summary boxes show ✖ Critical, ⚠ Warning, and ✓ Unknown severities. If the events include anomalies that are detected by Operations Analytics - Predictive Insights, the status indicator includes an anomaly icon: ✖, ⚠, or ✓. The same indicator for critical and warning anomalies appears next to the **Events** tab title as you drill down to application, group, and instance dashboard pages: Events ⚠ 2. For a hands-on demonstration, start the IBM Cloud Application Performance Management Guided Demo, scroll down the Tasks list, and select *Identify & Diagnose Predictive Insights Anomalies*.

Procedure

Complete these steps to identify anomalies and view them in the Operations Analytics - Predictive Insights **Service Diagnosis** view:

1. Click **Performance** > **Application Performance Dashboard** to open the **All My Applications** dashboard.
2. If a summary box has an **Events** status indicator that shows the anomaly icon, click the **Events** link.

   The application dashboard opens to the **Events** tab. The **Event Severity Summary** reports the total number of events including the number of anomalies.
3. Click a table row of an anomaly event, which is indicated in the **Severity** column by ✖, ⚠, or ✓.

   The row expands to show the event details.
4. Click **View anomaly analysis** to open the Operations Analytics - Predictive Insights **Service Diagnosis** view in a new browser tab or window.

What to do next

- Use the **Service Diagnosis** view to review the anomalous behavior in the components that support the application. Click ? to open the online help for the **Service Diagnosis** view.
- Return to the application dashboard and look for any other events on the managed system that might indicate a related issue. Click the **Status Overview** tab, and drill down to the managed system instance on which the event occurred to investigate further. Use the information to determine what actions need to be taken to avoid the issues identified by Predictive Insights.
- If you expect to see anomalies but none is displayed, the Operations Analytics - Predictive Insights training time might not be sufficient to produce anomalies. Two weeks is the typical training time. It is also possible that additional configuration is required.

Custom views

Use the IBM Cloud Application Business Insights Universal View to enhance the value that the predefined Application Performance Dashboard pages already provide by customizing your own pages.

The Universal View can be used to display resource monitoring data. It cannot be used to display synthetic transaction data, transaction tracking data, response time agent data, and deep-dive diagnostic data. By using the Universal View, you can quickly build monitoring pages for an application and save them for viewing. While viewing a saved custom dashboard page, you can view the dashboard in auto refresh mode or export the dashboard to Raw data file or edit the dashboard, or delete the dashboard.

The four default roles in Cloud APM: Role Administrator, Monitoring Administrator, System Administrator, and Monitoring User have different permissions to view and modify dashboard pages. For more information, see Table 1. Roles and permissions .

The options that are available in the **Custom Views** tab depend on whether the page is being edited or viewed.

Creating and managing custom pages

Use the Custom Views tab to create or edit dashboard pages for selected application or group or instance by adding or updating widgets that are populated with the resource metrics of your choice.

About this task

The pages that you create and save are associated with the selected application. For example, the Inventory Management application in the Cloud APM Guided Demo has the following monitoring agents: Linux OS, MySQL, Node.js, Hadoop, and Ruby. You can create and save a custom page at any level of the navigator from application to instance and then open it at the same level where it was created. A page that is created at a particular level can be opened only at the same level. The metrics available for the widgets can be from any of the resources in the application. Using the Inventory Management example, you can create a page with a table from the Ruby agent, a chart from Linux OS agent, and so on.

Procedure

The pages that you create and save are associated with the selected application. Complete the following steps to create and customize a dashboard page:

1. After you open the Application Performance Dashboard from the **Performance** menu, select an application.

   The **Custom Views** tab is displayed after **Status Overview** and **Events** tabs. You can also drill down to the group, subgroup, or instance level of the navigator.
2. Click the **Custom Views** tab.

   The tab shows the **Select a Template for your Custom page** window or the default page if a default page is already set.

   - If the **Select a Template for your Custom Page** window opens, go to step 4.
   - If the default page is displayed, go to step 3.
3. Click + **Add** to create a new page.
4. Click a template from the following default template options:

   - 1x1 Template
   - 1x2 Template
   - 2x2 Template
   - 2x3 Template
   - 3x3 Template
   - 3x2 Template
   - 2x1 Template
   - 1x3 Template
   - 3x1 Template

   If you click **Back**, the page that is marked as favorite or the first page from the list opens. If no page exists, then the **Select a Template for your Custom Page** window opens.
5. Customize the template. For details, see Customizing templates.
6. Create a widget. For details, see "Defining widget properties" on page 1090.
7. Click **Set Default Timeframe for the page** and set the default data retention period for the page to 1, 2, 4, 12, or 24 hours.
8. When you are ready to save the page, complete these steps:

   a) In the **Page Name** field, enter name for the page.

      Important: Space, underscore (_), and dash (-) are allowed in the **Page name** field. However, dash followed by underscore (-_) is not allowed. For example, `System-_Overview` is not allowed.

b) Click **Save**.

The following changes may occur on the dashboard or message may be displayed:

- The `Dashboard saved` message is displayed.
- If ***** is selected in the **Set Conditions**, then the following message is displayed:

  ```
  You selected * in Resource Instance or in Set Conditions, which will
  result in a large number of data series (such as lines on a graph). The
  large number of data series can make the page readability or performance
  unusable. The advisable limit for this chart is 50 data series. Adding
  specific values helps to narrow the data within recommended limits and
  results in a better user experience.
  ```

- A red indicator is displayed on the to indicate that chart type is not selected and you need to select it.
- A red indicator is displayed on the to indicate that Select Metric is not selected and you need to select it, and then the following message is displayed:

  ```
  A metric must be saved to save a chart.
  ```

9. Select any of the following options in the page title bar:

Option | **Description**

Retrieve latest resource Metric Types

Click to refresh metric types. If there is change in the metric type or metric when any agent patch is applied, you need to refresh the metric types.

The interval between two refreshes is restricted to 15 minutes. If you click **Retrieve latest resource Metric Types** within 15 minutes of previous refresh, then following message is displayed:

```
Metadata cache was refreshed recently.
Please wait for time_remaining minute(s) to reload it.
```

When the metadata is loading a loading image is displayed.

When the metadata is loaded, the following message is displayed:

```
Metadata Cache reloaded successfully.
```

If the metadata refresh takes more than 30 seconds, then the following message is displayed:

```
Reloading of Metadata cache might take some more time.
Do you want to wait until it is done?
```

You can click **Ok** or **Cancel**.

View Dashboard

Click to view the data in the dashboard.

Important: The limit for the number of rows that are returned per data definition is 11,000 rows. By default, latest data is displayed when the limit is crossed. For high volume of data, entire data for selected time interval is not displayed. For example, if you select to view last 24 hours of data for high volume data source, then only last 6 hours of data might be displayed if the 11000 rows limit is reached.

If in a chart data series exceed 50, then the following message displayed in the widget:

```
This chart cannot be loaded because the number of
data series (such as lines on a graph) exceeds 50.
The number of data series is currently
```

Option	Description
	`current_data_series`. You can reduce the number by selecting fewer metrics or resources instances, or by refining the Conditions per Metric. For more information, see Defining widget properties: for Cloud APM - http://ibm.biz/widgetprops and for Cloud APM Private - http://ibm.biz/widgetprops-private If a chart is taking more than 30 seconds to load, then the following message is displayed in the widget: `This chart took too long to load because of a large amount of data, long network latency, or connectivity issue. Reduce the number of Resource Instances or refine the Conditions per Metric to narrow down the data. For more information, see Defining widget properties: for Cloud APM -` http://ibm.biz/widgetprops `and for Cloud APM Private -` http://ibm.biz/widgetprops-private
Save As	Click the arrow next to **Save**, and click **Save As** and specify a different name in the **Page Name** field to save the page with a different name. **Important:** If you specify a page name same as an existing page, then the existing page is overwritten.
🗑 **Delete**	Click to delete the current page.
‹ Back **Back**	Click to go back to the previous page or favorite page.

What to do next

View the custom pages as described in "Viewing custom pages" on page 1093.

Customizing templates

You can customize the template by resizing, moving, or adding widget placeholders according to your requirement.

About this task

Remember: You can customize an existing template and use it. But the customized template cannot be saved for future use to create new dashboards.

Procedure

1. In the Custom Views tab, click ✎ **Edit Template**.
2. Select a widget placeholder.

   You can resize a widget placeholder from all the sides and drag it to a different location. If the widgets overlap each other while you are resizing or dragging them, the `Invalid resize operation` or `Invalid move operation` message is displayed.
3. To add a widget placeholder to existing template, complete the following steps:
   a) Click **Set page height** in menu options and specify a higher value for row count and click anywhere outside the menu to increase the height of the page.
   b) Specify a widget placeholder according to your requirement in the blank area on the page by placing the pointer and dragging it to create a box.
4. Use the following menu options to complete different operations on the template:

Option	Description
Undo	To undo the last action.
Redo	To redo the last action.
Delete selected box	To delete a widget, select the widget and click **Delete selected box** icon.
Reset	To create a blank template. To specify a widget placeholder on the blank template, place the pointer in the Draw Templates Here area and drag it to create a box. You can create widget placeholders of different sizes in the Draw Templates Here area. The placeholders can be moved or resized but cannot overlap each other.
Set page height	To set the height of the page. You can specify the row count as 20 through 120 rows.

5. Click **Edit Template** to use the template that is created.

What to do next

Create the widget. Go to step 6 in the Creating and managing custom pages topic.

Defining widget properties

Define different properties for the widgets such as metrics and charts to view real-time data in the widgets.

Procedure

To define the properties for a widget, complete these steps:

1. In a widget, click to select a chart type to display data.

   - Line
   - Area
   - Bar
   - Grid

   Important: For line, area, bar charts, if there are more than nine legends then the color of graph repeats after ninth legend. The color of the graph is same for 1st and 10th legend, 2nd and 11th legend, and so on.

   A green indicator is displayed on the to indicate that chart type is selected.
2. Specify the following chart properties for line, area, and bar charts:

   - X Axis Label
   - Y Axis Label
   - Show Legend
   - Show Interpolation: The data that is collected to be plotted on the chart, it might include some Null values. Therefore, when the chart is plotted, the chart line is disconnected where it encounters a Null value and multiple disconnected lines appear in the chart. If you select interpolation, the line on the chart does not look disconnected where it encounters a Null value, instead it connects to the next available valid value. Therefore, you get a single connected chart line when you select interpolation.

Note: For APM V8.1.4.0 IF0005 and later versions, the line and area charts no longer display disconnected lines for null values. Therefore, Show Interpolation feature is not required anymore and hence, it is not supported.

Important: Grid has no properties.

3. Click ▤ to select the metric content.

Option	Description
Resource Type	From the **Resource Type** list, select a resource. The available resources are associated with the application. If a resource that is a part of the application is not available in the list, its resource definition was not found. Either the resource definition was not published or the managed system is not connected to the Cloud APM server.
Metric Type	From the **Metric Type** list, select a data set that you want to include in the widget.
Metric	From the **Metric** list, select an attribute to include in the view. The available attributes are from the selected data set. To select metrics, complete the following steps: a. Click the **Metric** list. A pop-up window opens where the metrics are listed alphabetically sorted in ascending order. b. Click the attributes that are listed under Metrics (select one or more) or click **Select All**. **Note:** When you click **Select All**, all the metrics in the list get selected. c. Click ➔ to add the metrics under **Selected Metrics** list. d. If you want to delete a metric under **Selected Metrics**, click 🗑. e. Click the **Resource Instance** list to close the pop-up window. **Important:** For line, bar, and area charts, metric containing numeric value needs to be selected. Metrics containing string values cannot be displayed in these charts. **Tip:** For grid, limit your selection on metrics according to the output that suits your visibility on the UI.
Resource Instance	Initially, the selection is *, which retrieves metrics from all the instances in the list. Retain the default selection or select the instance from the list. **Important:** If you select an instance, then this widget cannot be used to display data for any other agent or instance. But it is good to specify the instance to avoid processing of huge data.
Set Condition for Metric Group	If the selected **Metric Type** has multiple elements, such as CPUs or disks, then **Set Condition for Metric Group** displays other elements for you to select from the **WHERE** field. **Important:** Specify values for elements in the **WHERE** field. Avoid specifying * to reduce processing of huge data. By default WHERE condition displays last 4 hours of data. This time interval can be changed between 1 to 24 hours by the system

Option	Description
	administrator, see “Changing time interval for WHERE condition data” on page 1092.
Actions	Click 💾 **Save** to save a metric. Click ✎ **Edit** to edit a metric. Click 🗑 **Delete** to delete a metric.

4. To add another metric, click **+ Add Another Metric**.

   Important: Not applicable for grid.
5. Close the **Select Metrics** window after all the metrics are added.
   All the metrics are saved automatically after you close the **Select Metrics** window.

   The following changes may occur on the dashboard or message may be displayed:

   - If * is selected in the **Resource Instance** list in any of the metrics, then the following message is displayed:

     ```
     You selected * in Resource Instance or in Set Conditions, which will result
     in a large number of data series (such as lines on a graph). The large
     number of data series can make the page readability or performance
     unusable. The advisable limit for this chart is 50 data series. Adding
     specific values helps to narrow the data within recommended limits and
     results in a better user experience.
     ```

   - A green indicator is displayed on the ✓ to indicate that metrics are selected correctly to narrow the data within recommended limits.
   - A orange indicator is displayed on the ! to indicate that metrics are not selected correctly to narrow the data within recommended limits. Either * is selected in **Resource Instance** or in **Set Conditions**.
6. Click ✎ to enter the widget title.
   If widget title is not added, then the first metric name is assigned as the widget title automatically.

What to do next

Similarly, add charts, metrics, and titles to all the widgets and then go to step 7 in the Creating and managing custom pages topic.

Changing time interval for WHERE condition data

The time interval can be changed between 1 to 24 hours by the system administrator.

Procedure

To change the time interval, system administrator can complete the following steps:

1. Log in to the APM server where the build is deployed.
2. On the command line, run the following commands:

   ```
   export CLASSPATH=$CLASSPATH:/install_dir/gaian/lib/derbytools.jar:
   export CLASSPATH=$CLASSPATH:/install_dir/gaian/lib/derbyclient.jar:
   export CLASSPATH=$CLASSPATH:/install_dir/gaian/lib/derby.jar:
   java org.apache.derby.tools.ij
   ```

```
connect 'jdbc:derby://localhost:port/
gaiandb;user=gaiandb;password=gaian_db_password;';
```

In these commands, *install_dir* refers to the directory where APM was deployed, by default it is `/opt/ibm`. In the **`connect`** command, *port* refers to the value of port on which the database is configured and *gaian_db_password* is the Gaian database password. Contact IBM support for this password.

3. After the database is connected, run the following query to modify values for time range:

   ```
   UPDATE "OED_TOOL"."PREFERENCETABLE" SET PREFERENCES='-24H' WHERE
   FIELD='TIMEINTERVAL';
   ```

   ```
   Commit;
   ```

   ```
   Exit;
   ```

   Here the value of the time range is given in the `SET PREFERENCES='-24H'`. It can be set from 1H to 24H.

Viewing custom pages

After you create and save dashboard pages for an application, group, subgroup, or instance in the **Custom Views** tab, you can view them at any time. Some of the options that you can select include refreshing the page, selecting different time interval, editing the page to retrieve data from different resources, and exporting the dashboard as Raw Data file.

Procedure

Complete these steps to view a saved page in the **Custom Views** tab of the dashboard.

1. After you open the Application Performance Dashboard from the **Performance** menu, select an application.

   The **Custom Views** tab is displayed after **Status Overview** and **Events** tabs. You can also drill down to the group, subgroup, or instance level of the navigator.
2. Select the **Custom Views** tab.

   The tab shows the **Select a Template for your Custom page** window or the default page if a default page is already set.
3. Click ▾ in the page list and select one of the saved pages from the list.

   The pages available were saved by you or were shared by another user.

   After you select a saved page, the current and historical data samples are reported in the page.
4. Select any of the viewing options in the page title bar:

Option	Description
⟳ **Refresh**	Indicates that auto refresh is off. Click to turn auto refresh on.
⟳ **Refresh**	Indicates that auto refresh is on. Click to turn auto refresh off. **Important:** Default refresh time is 1 min.
⇪ **Export > Raw Data**	Click to export the page as DAT format. Since multiple DAT files are exported, they are downloaded to your computer in ZIP format. **Remember:** If the downloaded file does not have any extension, then add zip as the extension to it. **Important:** If the file is not downloaded to your computer, then check if software to block pop-up ad windows is enabled. You can add this site to your exception list.

Option **Description**

Extract the downloaded ZIP file. The extracted files are plain text files. The file contains of page name, duration, filters, date, time, interval, chart title, and data. The data delimiter is pipe.

You can open the DAT files by using appropriate editor or import the files to excel by specifying appropriate value separators.

⤒ **Export > PDF**

Click to export the page as PDF format.

The file contains page name, time interval, widgets, created by, and report created on.

✎ **Edit**

Click to edit the current page.

You can change the charts and metrics in the widgets, or add new widgets, change the default time frame, or edit the name for the page.

🗑 **Delete**

Click to delete the current page.

+ **Add**

Click to create a new page. To customize the page and save it, see "Creating and managing custom pages" on page 1087.

5. Select any of the following viewing options in the widget:

Option **Description**

Chart Type

Click the **Chart Type** icon and select an appropriate option from the list to change the existing chart type.

- For line and area charts, **Lines** and **Areas** options are available.
- For bar chart, **Clustered Bars** and **Clustered Columns** options are available.

Important: For grid, there are no chart type options.

For data displayed in a grid, you can filter the data as follows:

a. Click **Define filter**. The **Filter** window opens.

b. Specify values for **Column**, **Condition**, and **Value** to add a filter rule.

   Note: You can filter numeric values and text values by selecting appropriate conditions.

c. Click ⊕ **Add Filter Rule** to add another filter rule. You can add multiple filter rules.

d. In the **Match** field, select **All rules** or **Any rule** to filter the data.

   You can select **Match case** if you want to search as per the case of text that you provide in the **Value** field.

e. Click **Filter** to filter the data that is displayed in the grid.

f. Click **Clear filter** to clear the filter results.

g. Click **Cancel** to close the **Filter** window.

⊖ **Collapse**

Click to collapse the widget.

Option	Description
⊕ **Expand**	Click to expand the widget.
Maximize	Click to maximize the widget to the size of the page.
Restore	Click to restore the widget to its original size.
Legends	The widget contains check boxes for each metric. Select or clear the check boxes for each metric to view data of a particular metric or multiple metrics.

6. You can filter the data on the page by using the **Date**, **Time**, and **Interval** lists. You can also define a Custom filter for the page to display data for selected date and time intervals. To use Custom filter, from the **Interval** list, select **Custom**, and then in the **Time Period Selection** window, select the required date and time intervals.

   Note:

   - The custom filter option is available from APM V8.1.4.0 IF0005 onwards. Pages that are created by using earlier versions of Cloud APM do not display the Custom filter option.
   - Use Custom filter to filter data for a minimum time interval of 1 minute and maximum time interval of 24 hours.
   - When you are applying a custom filter, in the **Time Period Selection** window, if you click **Cancel**, then on the dashboard page, the **Interval** list does not display the interval that you applied earlier.
   - If you apply Custom filter to a page, then the data on the page is not auto refreshed.

7. To set a default page, click ▾ in the page list and click ☆ **Favorite** next to the page name that you want to set as the default page.

Dashboard utilities

Use the available options to manage the appearance and behavior of the **Application Performance Dashboard** pages.

Copying the dashboard URL

After you navigate to a place in your application hierarchy, the URL in the browser address box does not change for the new view. You can copy the URL of the Application Performance Dashboard page that you are displaying. Paste the URL into a new browser window to open the dashboard page or use the URL to access the dashboard later or to share with other.

Procedure

1. Navigate to the Application Performance Dashboard page that you want to remember.
2. Click **Actions** > **Copy URL**.
3. Right-click the **Link to the current page** hypertext link and select the option to copy the URL.

What to do next

Keep a copy of the URL or share with other users in your managed environment. After you paste the URL into your browser's address box, the target dashboard page is opened in the Cloud APM console.

If you are not logged on to the Cloud APM server, you are prompted to enter your user ID and password before the target dashboard page can be displayed. If the **Getting started** page is opened instead of the dashboard page, press F5 or click the browser's refresh toolbar button. You can turn off the **Getting started** page for future work sessions by clearing the "Show this **Getting started** page at startup" check box.

Setting a trace

Adjust the trace settings to help your administrator or IBM Support to diagnose the cause of problems with the Application Performance Dashboard.Several levels of tracing are available while you work with the navigator and the **Status Overview** tab. You can start a detailed level of tracing exactly at the point in the user interface where you are having a problem, then return tracing to a reduced level after capturing the necessary log data. For example, if a particular dashboard is behaving unexpectedly, you can raise the trace level before opening the dashboard to log the activity and then return trace logging to the normal level.

About this task

Take the following steps to set the trace level when you want to increase or reduce the amount of trace logging.

Procedure

1. If the Application Performance Dashboard is not open, select it from the **Performance** option in the navigation bar.
2. Select All My Applications or an application from the navigator or Status Overview tab.
3. Click **Actions** > **Trace level** and select one of the following levels:
   - **Verbose** to have all activity logged. Verbose trace level includes Moderate, Light, and Minimal trace logging.
   - **Moderate** to have variable changes logged, such as what parameters were passed in and what calculations were made. Moderate trace level includes Light and Minimal trace logging.
   - **Light** to log error and variable activity. You might want to set the trace to this level if you have a problem such as no data being returned but the dashboard continues to function. Light trace level includes Minimal trace logging.
   - **Minimal** is the default setting and records only unrecoverable errors. You can set the trace level back to minimal after collecting a specific activity sequence. Even if a different trace level was set before logout, the trace is always reset to the lowest level the next time you log in.
4. If you want to send performance records to a common logging file, select **Enable Log Performance Statistics**.

   The performance information from the console is written to the server where it can be combined with performance statistics from the server to provide end to end transaction response time. The required performance information includes the time that a function was started and the time that it ended.

Results

The trace is adjusted to the level chosen. The next time you log in, the trace is to **Minimal** until you change it again.

To keep communications traffic to a minimum, the log messages are transferred in batches. A final transfer is made after you log out, whether manually or after a timeout period. (If the browser fails, no final logging is sent.) The log is saved on the server computer and named `itp.log`. A new `itp.log` is created each time the server is restarted.

If you set **Enable Log Performance Statistics**, records similar to those in the following example are saved to *`install_dir`*`/usr/servers/apmui/logs/itp.log`:

```
<record>
    <date>2013-10-02T10:52:46</date>
    <millis>1380736366788</millis>
    <sequence>28008</sequence>
    <level>INFO</level>
    <class>StatusItemList</class>
    <method>tracing</method>
    <thread>96</thread>
    <message>BeginTrace:onSelectApp:272wt877d05</message>
</record>
<record>
    <date>2013-10-02T10:52:46</date>
```

```
     <millis>1380736366809</millis>
     <sequence>28009</sequence>
     <level>INFO</level>
     <class>StatusItemList</class>
     <method>tracing</method>
     <thread>96</thread>
     <message>EndTrace:onSelectApp:272wt877d05</message>
</record>
```

Locking the Cloud APM console

You can temporarily lock your work session without having to log out of the Cloud APM console. The session lock feature is not available on the Apple iPad.

Procedure

1. While you are logged in to the Cloud APM console, click ***apmadmin*** > **Lock Session** where *apmadmin* is the name that you used to log in.

   The log in screen is displayed and your session is locked.
2. To unlock your session, enter the password for your user ID.

   Your work session resumes.

Reports

Historical reports are available in Cloud APM console for data that are collected by the Response Time Monitoring Agent, the WebSphere Applications agent, and the Synthetic Playback agent.

You can run reports from the **All My Applications** dashboard. From any Cloud APM console page, click **Performance** > **Application Performance Dashboard** to open the **All My Applications** dashboard.

Response Time Monitoring Agent reports

To view the **Application Performance and Usage** or **Compare Application Performance over Two Time Periods** reports, select an application that includes Response Time Monitoring agent managed systems, and select **Actions** > **Launch To Reports**.

To view the **All My Applications** or **Compare Performance of Multiple Applications** report, select **All My Applications**, and select **Actions** > **Launch To Reports**.

WebSphere Applications agent reports

To view any WebSphere Applications agent reports, select an application that includes WebSphere Applications agent managed systems, and select **Actions** > **Launch To Reports**.

Synthetic Playback agent

To view any Synthetic Playback agent reports, select an application that includes synthetic transactions, and select **Actions** > **Launch To Reports**.

Important: If your reports do not display the correct times for transaction instances, you might need to reset the user time zone in Cognos Connection. For more information, see the technote How to convert time into local time depending on time zone setting.

For information on the supported browsers for viewing Synthetic Playback agent, Response Time Monitoring agent, and WebSphere Applications agent reports, see the Software Product Compatibility Reports for Cognos 10.2.1.7.

Response Time Monitoring Agent reports

Historical reports are available for data that is collected by the Response Time Monitoring agent. Response Time Monitoring Agent reports are not available in Cloud APM, Base. They are only available in Cloud APM, Advanced.

Note: If you configured the Response Time Monitoring agent to use the IBM HTTP Server for Response Time module then, Transaction Data Volume is not available in the reports for any applications monitored by the agent. Transaction Count data is present.

There are two types of reports available for data that is collected by the Response Time Monitoring agent: active and simple.

Active reports

Active reports are viewed in a browser in MHTML format. Internet Explorer supports MHTML by default. For other browsers, an MHTML support plug-in can be installed. Active reports are also referred to as offline interactive reports.

Simple reports

Simple reports are viewed in IBM Cognos Viewer. IBM Cognos Viewer is the default report output viewer.

The following historical predefined reports are available for data that is collected by the Response Time Monitoring agent:

Table 256. Predefined historical reports

Report	Type
All My Applications	Active
Application Performance and Usage	Active
Compare Application Performance over Two Time Periods	Simple
Compare Performance of Multiple Application	Simple

The data for the reports is stored in the DATAMART Db2 database. The reports display summarized daily, weekly, and monthly data which is retained for 26 weeks, 12 months, and 3 years respectively. Cloud APM does not provide scripts or instructions to change these retention periods.

For more information about the mapping between the Response Time Monitoring agent and Performance Management reports, see Response Time Monitoring agent attributes mapping.

All My Applications report

Use the All My Applications report to view information about user devices, data volume, response time, and error counts.

In this report, you view information for all your applications. Specify the report time period as **Last day** (default), **Last Week**, or **Last Month**. View the following information by selected time period by application:

- Stacked column chart of transaction count
- Column chart of transaction data volume
- Column chart of average transaction response time
- Stacked column chart of error counts

Application Performance and Usage report

Use this report to view performance, availability, and user device information for single applications.

In the **Select an Application** window, select an application. Click **Next**. In the **Select Key Transactions for Application**, select the transaction/s you want to filter the report by. Click **OK**. This report has three tabs - Performance. Availability, and Devices. The default time interval is week.

On the Performance tab, view the following information by selected time interval for the application you are currently viewing in the Application Performance Dashboard:

- Line chart of average response time by (key) transactions
- Line chart of average transaction response time by success, server error, and client error
- Bar and line chart of transaction data volume, the bars show transaction data volume, and the line shows transaction data volume average
- Bar and line chart of transaction count, the bars show transaction count, and the line shows polynomial values and moving average

On the Availability tab, view the following information by selected time interval for the application you are currently viewing in the Application Performance Dashboard:

- Stacked bar chart of successful versus failed transaction percentage, the failures are broken down into server error and client error
- Stacked bar chart of successful versus failed transaction count by device types, the failures are broken down into server error and client error
- Pie chart of most frequently occurring error codes

On the Devices tab, view the following information by selected time interval for the application you are currently viewing in the Application Performance Dashboard:

- Bar chart of transactions by device type
- Bar chart of transactions by device operating system
- Bar chart of transaction by device browser
- Table showing transaction performance by dimensions, you can filter this table based on device type, device operating system, device brand, and device browser

Compare Application Performance over Two Time Periods report
Use this report to examine application performance for a selected application.

In the **Select Application and Time Frequency** window, specify an application, and time frequency (weekly, daily, monthly). In the **Select Time Periods** window, choose time periods appropriate to the time interval and click **OK**.

The report displays the following charts for the selected application, for the selected time periods by the selected time interval:

- Line chart of transaction count for device type
- Line chart of transaction volume
- Line chart of average response time
- Line chart of error count

Compare Performance of Multiple Applications report
Use this report to compare the performance of multiple applications for the same time period.

In the **Select Applications and Time Frequency**, specify an application, and time frequency (weekly, daily, monthly). Click **Next** . Choose a time period appropriate to the time interval.

The report displays the following charts for the selected applications, for the selected time period by the selected time frequency:

- Line chart of transaction count
- Line chart of transaction data volume
- Line chart of average response time
- Line chart of error count

Response Time Monitoring Agent attributes mapping

Some Cloud APM reports are based on data that is collected by the Response Time Monitoring Agent. The data in these reports maps to Response Time Monitoring agent attributes.

The following table provides mapping of data items in Response Time Monitoring agent reports to the agent attributes:

Table 257. Response Time Monitoring agent attribute mapping

Report data item	Description	ODI Attribute name	ODI file Column
Application Name	The name of the monitored application reported to the Cloud APM console	Application Name	T5TXCS.APPLICATIN
Transaction Count	The total number of request and response sequences that are observed by the monitoring agent during the current aggregate interval.	Total Requests	T5TXINS.TOTREQ
Client Errors	The number of HTTP requests with a status code 400 - 499.	Client Errors	T5TXCS.NUM4XX
Server Errors	The number of HTTP requests with a status code 500 - 599.	Server Errors	T5TXCS.NUM5XX
Transaction Name	The transaction name reported to the Application Management Console.	Transaction Name	T5TXCS.TRANSACTN
Transaction Status	The response code that is associated with the transaction	Status Code	T5TXCS.STATUSCODE
Code Reply Kilobytes	The total number of kilobytes in each reply of the request during the data interval.	Reply Bytes	T5TXCS.REPLYBYT
Request Kilobytes	The total number of kilobytes in the request during the data interval.	Request kBytes	T5TXCS.REQBYTES
Total Kilobytes	The total number of kilobytes transferred for all request during the time period.	Total Bytes	T5TXCS.TOTBYTES

Table 257. Response Time Monitoring agent attribute mapping (continued)

Report data item	Description	ODI Attribute name	ODI file Column
Total Object Count	The total number of objects that are embedded in a web page for the time period	Total Object Count	T5TXCS.OBJCNT
Total Object Size	The total size of all objects that are embedded in the web page for the time period.	Total Object Size	T5TXCS.OBJSIZE
Response Time (seconds)	The total number of seconds required for the overall server transaction to complete.	Response Time	T5TXCS.RESPTIME
Render Time	The elapsed time, in seconds, to fully render the web page on the web browser using embedded JavaScript tags.	Render Time	T5TXCS.RENDERTIME
Client Time	The average elapsed time, in seconds, that the transaction spends running on the client during the current monitoring interval.	Average Client Time	T5TXCS.CLIENTTIME
Load Time	The average elapsed time, in seconds, from the time the user requests a download to the completion of the web object download.	Average Load Time	T5TXCS.LOADTIME
Browser	A description of the web browser on which the web page is displayed.	Browser Description	T5TXCS.BROWSEDESC
Server	The name or IP address of the server for the TCP Transaction.	Server Description	T5TXCS.SERVERDESC
URL Hostname	The TCP/IP hostname of the URL.	URL Hostname	T5TXCS.URLHOST
URL Method	The method that is used for performing HTTP requests (GET, POST, HEAD, PUT, OPTIONS, DELETE, TRACE, or CONNECT).	Method	T5TXCS.METHOD

Table 257. Response Time Monitoring agent attribute mapping (continued)

Report data item	Description	ODI Attribute name	ODI file Column
URL Details	The URL path to the file on the server hosting the web page.	URL Path	T5TXCS.URLPATH

For more information on the Response Time Monitoring agent, see the Transaction Monitoring Reference.

Generating Synthetic Playback agent reports

Run reports for applications that are associated with synthetic transactions.

About this task

Select an application and an associated synthetic transaction in the Application Performance Dashboard and then generate reports based on your selection. The data for the reports is stored in the DATAMART Db2 database. The reports display summarized hourly, weekly, and monthly data which is retained for 371 days, 53 weeks, and 12 months respectively. Cloud APM does not provide scripts or instructions to change these retention periods.

Five multi-page reports are available:

Transactions Overall

This two page report displays the response times and availability ratios of the selected synthetic transaction over a set date range.

Page one displays the following data:

- A line chart of the response times of the selected synthetic transactions at set intervals over a set date range
- A table of the average response times in seconds of each synthetic transaction over the set date range

Page two displays the following data:

- A line chart of the availability ratios of the selected synthetic transactions at set intervals over a set date range
- A table of the average availability ratio of each synthetic transaction over the set date range

You can access two extra reports from the **Transactions Overall** report, **Timely Analysis by Transactions** and **HTTP Metrics by Transactions**.

Timely Analysis by Transaction displays HTTP metrics of the selected synthetic transaction at set intervals over a set date range. The report includes the following items:

- A column chart of the HTTP metrics of the selected synthetic transaction at set intervals over the set date range
- A table of the HTTP metrics in milliseconds of the selected synthetic transaction over the set date range

HTTP Metrics by Transaction displays HTTP metrics of the selected synthetic transaction at set intervals over a set date range. The report includes the following items:

- A column chart of the HTTP metrics of the selected synthetic transaction at set intervals over the set date range
- A table of the HTTP metrics in milliseconds of the selected synthetic transaction over the set date range

Transaction Detail By Locations

This two page report displays the response times and availability ratios by location of the selected synthetic transactions and subtransactions over a set date range.

Page one displays the following data:

- Line charts of the response times by location of the selected synthetic transactions and subtransactions at set intervals over a set date range
- Tables of the average response times in seconds of all synthetic subtransactions over a set date range at each location

Page two displays the following data:

- Line charts of the availability ratios by location of the selected synthetic transactions and subtransactions at set intervals over a set date range
- Tables of the average availability ratios of all synthetic subtransactions over a set date range at each location

You can access four extra reports from the **Transaction Detail by Locations** report, **Timely Analysis by Locations of Transaction**, **HTTP Metrics by Locations of Transaction**, **Timely Analysis by Locations of Subtransaction**, and **HTTP Metrics by Locations of Subtransaction**.

Timely Analysis by Locations of Transaction displays HTTP metrics of a synthetic transaction by locations at set intervals over a set date range. The report includes the following items:

- A column chart of the HTTP metrics of the selected synthetic transaction by locations at set intervals over the set date range
- A table of the HTTP metrics in milliseconds of the selected synthetic transaction by locations over the set date range

HTTP Metrics by Locations of Transaction displays HTTP metrics of a synthetic transaction by locations at set intervals over a set date range. The report includes the following items:

- A column chart of the HTTP metrics of the selected synthetic transaction by locations at set intervals over the set date range
- A table of the HTTP metrics in milliseconds of the selected synthetic transaction by locations over the set date range

Timely Analysis by Locations of Subtransaction displays HTTP metrics of a subtransaction by locations at set intervals over a set date range. The report includes the following items:

- A column chart of the HTTP metrics of the selected subtransaction by locations at set intervals over the set date range
- A table of the HTTP metrics in milliseconds of the selected subtransaction by locations over the set date range

HTTP Metrics by Locations of Subtransaction displays HTTP metrics of a synthetic subtransaction by locations at set intervals over a set date range. The report includes the following items:

- A column chart of the HTTP metrics of the selected subtransaction by locations at set intervals over the set date range
- A table of the HTTP metrics in milliseconds of the selected subtransaction by locations over the set date range

Transaction Detail By Subtransactions

This two page report displays the response times and availability ratios of synthetic subtransactions at set intervals over a set date range.

Page one displays the following data:

- A line chart of the response times of the selected synthetic subtransactions at set intervals over a set date range
- A table of the average response times in seconds of each synthetic subtransaction over the set date range

Page two displays the following data:

- A line chart of the availability ratios of the selected synthetic subtransactions at set intervals over a set date range
- A table of the availability ratios of each synthetic subtransaction over the set date range

You can access two extra reports from the **Transaction Detail by Subtransactions** report, **Timely Analysis by Subtransactions** and **HTTP Metrics by Subtransactions**.

Timely Analysis by Locations of Subtransaction displays HTTP metrics of a subtransaction by locations at set intervals over a set date range. The report includes the following items:

- A column chart of the HTTP metrics of the selected subtransaction at set intervals over a set date range
- A table of the HTTP metrics in milliseconds of the selected subtransaction over the set date range

HTTP Metrics by Locations of Subtransaction displays HTTP metrics of a synthetic subtransaction by locations at set intervals over a set date range. The report includes the following items:

- A column chart of the HTTP metrics of the selected subtransaction at set intervals over a set date range
- A table of the HTTP metrics in milliseconds of the selected subtransaction over the set date range

Trend Of Transactions

This four page report displays a trend analysis of response times, availability ratios, and HTTP metrics over the previous week and over the previous five weeks.

Page one displays trend data on the response times and availability ratios of a synthetic transaction:

- A combined line chart of the average response times of a selected synthetic transaction over the previous week and over the previous 5 weeks
- A combined line chart of the availability ratio of a selected synthetic transaction that compares the availability ratio for the previous week with the baseline availability ratio over the previous 5 weeks
- A table of the average response time and availability ratio of a synthetic transaction over the previous week and over the previous 5 weeks date range
- A combined line chart of the average response times of a selected synthetic transaction over the previous week and over the previous 5 weeks, by location
- A combined line chart of the availability ratio of a selected synthetic transaction that compares the availability ratio for the previous week with the baseline availability ratio over the previous 5 weeks, by location
- A table of the average response times and availability ratios of a selected synthetic transaction over the previous week and over the previous 5 weeks, by location
- A combined line chart of the average response times of the subtransactions of a selected synthetic transaction over previous last week and over the previous 5 weeks
- A combined line chart of the average availability ratios of the subtransactions of a selected synthetic transaction that compares the availability ratio for the previous week with the baseline availability ratio over the previous 5 weeks
- A table of the average response times and availability ratios of the subtransactions of a selected synthetic transaction over the previous week and over the previous 5 weeks

Page two displays trend data on the HTTP metrics of a synthetic transaction:

- A combined line chart of the average blocking times of a selected synthetic transaction over the previous week and over the previous 5 weeks
- A combined line chart of the average DNS times of a selected synthetic transaction over the previous week and over the previous 5 weeks
- A combined line chart of the average SSL times of a selected synthetic transaction over the previous week and over the previous 5 weeks
- A combined line chart of the average connect times of a selected synthetic transaction over the previous week and over the previous 5 weeks

- A combined line chart of the average sending times of a selected synthetic transaction over the previous week and over the previous 5 weeks
- A combined line chart of the average receiving times of a selected synthetic transaction over the previous week and over the previous 5 weeks
- A combined line chart of the average rendering times of a selected synthetic transaction over the previous week and over the previous 5 weeks
- A table of average HTTP metrics of a synthetic transaction over the previous week and over the previous 5 weeks

Page three displays trend data on HTTP metrics in milliseconds of a synthetic transaction by location. The charts and table compare HTTP metric averages over the previous week with the baseline metric average over the previous 5 weeks:

- Seven combined line charts that compare different average HTTP metrics of a selected synthetic transaction for the previous week with the baseline metric over the previous 5 weeks by location
- A table of average HTTP metrics of a selected synthetic transaction over the previous week and over the previous 5 weeks by location

Page four displays trend data on HTTP metrics in milliseconds for subtransactions. The charts and table compare HTTP metric averages over the previous week with the baseline metric averages over the previous 5 weeks:

- Seven combined line charts that compare average values of different HTTP metrics of a selected synthetic transaction for the previous week with the baseline metric over the previous 5 weeks by subtransaction
- A table of average values of HTTP metrics of a selected synthetic transaction over the previous week and over the previous 5 weeks by subtransaction

Trend Of Subtransactions

This two page report displays a trend analysis of the response times, availability ratios, and HTTP metrics for subtransactions over the last week and over the previous five weeks.

Page one displays trend data on the response times and availability ratios of subtransactions for the previous Sunday, the previous week, and the previous five weeks:

- A table that compares subtransaction response times and availability ratios from the previous Sunday, the previous week, and the previous 5 weeks
- A combined line chart that compares the average response times of subtransactions for the previous week with the baseline response time for the previous 5 weeks
- A combined line chart that compares the average availability ratio of subtransactions for the previous week with the baseline availability ratio for the previous 5 weeks
- A table of average response times and availability ratios of subtransactions for the previous week.
- A table of average response times and availability ratios of subtransactions for the previous 5 weeks.

Page two displays trend data on HTTP metrics in milliseconds for subtransactions. The charts and tables display HTTP metric averages of subtransactions over the previous week with the baseline metric averages over the previous five weeks:

- Seven tables of average values of HTTP metrics of the selected transaction for the previous Sunday, the previous week, and over the previous 5 weeks by subtransactions
- Seven combined line charts that compare different average HTTP metrics in milliseconds of the selected transaction for the previous week with the baseline availability ratio over the previous 5 weeks by subtransactions
- A table of average values of HTTP metrics of subtransactions for the previous week.
- A table of average values of HTTP metrics of subtransactions for the previous 5 weeks.

Procedure

To generate reports, complete the following steps:

1. Click the **Performance** icon and select **Application Performance Dashboard**. To choose an application, expand **All My Applications** and select an application. To display all synthetic transactions that are associated with the selected application, click **Groups** > **Transactions** > **Synthetic Transactions**.
2. Select a synthetic transaction from the Transaction List table. To run a report, click **Actions** > **Launch to Reports** and select one of the following reports:

   - **Transactions Overall**
   - **Transaction Detail By Locations**
   - **Transaction Detail By Subtransactions**
   - **Trend of Transactions**
   - **Trend of Subtransactions**

   A configuration page opens in a new tab in your web browser.
3. To set the date range for your report, select a predefined date range, or enter a custom date range.
4. To set the time interval for your report, select an interval from **Time Type**. Set your report to display data from your synthetic transactions and subtransactions at **Hourly**, **Daily**, or **Weekly** intervals, over the set date range. To generate your report, click **Finish**.
5. To view reports on HTTP metrics for transactions, subtransactions, or locations, you must select a transaction, subtransaction, or location in the **Transactions Overall**, **Transaction Detail By Locations**, or **Transaction Detail By Subtransactions** reports.

   - To view **Timely Analysis by Transactions**, right-click on a transaction name in the **Transactions Overall** report and select **Go To** > **Http Metrics Analysis by Time**.
   - To view **HTTP Metrics by Transactions**, right-click on a transaction name in the **Transactions Overall** report and select **Go To** > **Http Metrics Aggregation**.
   - To view **Timely Analysis by Locations of Transaction**, right-click on a transaction name in the **Transaction Detail By Locations** report and select **Go To** > **Http Metrics Analysis by Time**.
   - To view **HTTP Metrics by Locations of Transaction**, right-click on a transaction name in the **Transaction Detail By Locations** report and select **Go To** > **Http Metrics Aggregation**.
   - To view **Timely Analysis by Locations of Subtransaction**, right-click on a subtransaction name in the **Transaction Detail By Locations** report and select **Go To** > **Http Metrics Analysis by Time**.
   - To view **HTTP Metrics by Locations of Subtransaction**, right-click on a subtransaction name in the **Transaction Detail By Locations** report and select **Go To** > **Http Metrics Aggregation**.
   - To view **Timely Analysis by Subtransactions**, right-click on a subtransaction name in the **Trend of Subtransactions** report and select **Go To** > **Http Metrics Analysis by Time**.
   - To view **HTTP Metrics by Subtransactions**, right-click on a subtransaction name in the **Trend of Subtransactions** report and select **Go To** > **Http Metrics Aggregation**.

WebSphere Applications agent reports

Predefined reports are available for data that is collected by the WebSphere Applications agent.

Scope of the WebSphere Applications agent report:

Supported subnodes

- WebSphere Application Server (KYNS)
- WebSphere Portal Server (KYNR)

Unsupported subnodes

- WebSphere Process Server (KYNP)

The data for the reports is stored in the WAREHOUS Db2 database. The reports display hourly, daily, weekly, and monthly data which is retained for 1 month, 3 months, 1 year , and 1 year respectively. Cloud APM does not provide scripts or instructions to change these retention periods. The following reports are available for data that is collected by the WebSphere Applications agent:

Application Request Performance

Description

This report analyzes how applications perform at an aggregated level across an application server. The pie-chart shows the aggregate level requests for applications. The bar chart shows the average response time for applications at an aggregate level. The two time series line charts show the average response time and total request count trend for all of the applications. To drill down into the individual requests for an application, click a pie slice or a bar.

Parameters

Date Range: select one of the predefined reporting periods or select exact start and end times from the calendar.

Required parameters: Summarization Type and Application Server Type

Tables Used

Request_Analysis_*V

DB Connection Pools

Description

This report analyzes database connection pools in an application server. The table shows the key statistics for all the connection pools at an aggregate level. When you select a specific data source, two trend charts show the trend of the key statistics.

Parameters

Date Range: select one of the predefined reporting periods or select exact start and end times from the calendar.

Required parameters: Summarization Type, Application Server Name

Tables Used

DB_Connection_Pools_*V

EJB Performance

Description

This report analyzes how EJBs deployed in the application server perform. The pie chart shows the aggregate level method count for the EJBs. The bar chart shows the average Method Response Time across EJBs at an aggregate level. The Two Time series charts show the Method Invocation Count trend and Average Method Response Time trend for all EJBs. The trend lines can be filtered per EJB by clicking a row from the list.

Parameters

Date Range: select one of the predefined reporting periods or select exact start and end times from the calendar.

Required parameters: Summarization Type, Application Server Name

Tables Used

Enterprise_Java_Beans_*V

GC Usage of Application Server

Description

This report analyzes garbage collection. Use this report to determine whether garbage collection is creating problems or if the heap is not sized properly. The first graph shows the average heap percent that is used and the average real-time percent garbage collection over time. The second graph shows the average real-time percent garbage collection runs and the average garbage collection rate.

Parameters

Date Range: select one of the predefined reporting periods or select exact start and end times from the calendar.

Required parameters: Summarization Type, Application Server Type

Tables Used

Garbage_Collection_Analysis_*V

JVM Usage for Application Server

Description

This report analyzes how the JVM of an application server performs. The stacked bar chart shows how the JVM Memory is used and is freed up. The dual line chart shows the JVM CPU consumption vis-a-vis JVM Memory usage.

Parameters

Date Range: select one of the predefined reporting periods or select exact start and end times from the calendar.

Required parameters: Summarization Type, Application Server Type

Tables Used

Application_Server_*V

Threadpools

Description

This report analyzes thread pools in an application server. The table shows the key statistics for all thread pools at an aggregate level. Once you select a thread pool from the list, the trend chart shows the trending of the key statistics for the selected Thread Pool. If no thread pool is selected, the trends show summary of all the thread pools.

Parameters

Date Range: select one of the predefined reporting periods or select exact start and end times from the calendar

Required parameters: Summarization Type, Application Server Type

Tables Used

Thread_Pools_*V

Web Application Performance

Description

This report analyzes how the applications perform in the web container of an application server (PMI Data). The pie charts show the aggregate level requests for the applications. The bar chart shows the average response time for applications at an aggregate level. The two time series line charts show the Average Response time and Total Request count trend for all the applications. Click a pie slice or a bar or a line to drill down into the individual servlet/jsps for that application.

Parameters

Date Range: select one of the predefined reporting periods or select exact start and end times from the calendar.

Required parameters: Summarization Type, Application Server Type

Tables Used

Thread_Pools_*V

Application Request Performance for Clusters

Description

This report analyzes how servers in a cluster are performing. The first chart shows the number of requests that are completed by each of the cluster members during the selected time interval. The second chart provides information about the average response time trend for each of the cluster members. There is a separate line in this chart for each server in the cluster. Click a line to drill down into the individual server data. The Application Request Performance report for this server opens

Parameters

Date Range: select one of the predefined reporting periods or select exact start and end times from the calendar.

Required parameters: Summarization Type, Cluster Name

Tables Used

Request_Analysis_*V

JVM and GC Usage for Clusters

Description

This report analyzes JVM and garbage collection usage trends by each of the cluster members. The first chart shows the average real-time percent garbage collection runs. The second chart shows average heap percent used. The last charts show CPU and JVM memory usage. All of these charts show data for each cluster member as a separate line.

Parameters

Date Range: select one of the predefined reporting periods or select exact start and end times from the calendar.

Required parameters: Summarization Type, Cluster Name

Tables Used

Garbage_Collection_Analysis_*V, Application_server_*V

Top applications with slowest response times across servers

Description

This report analyzes how the applications perform at an aggregated level across all application servers. A bar chart shows the average response time for applications at an aggregate level.

Parameters

Date Range: select one of the predefined reporting periods or select exact start and end times from the calendar.

Required parameters: Summarization Type, Number of Applications

Chapter 11. Upgrading

Upgrade your agents and data collectors to get the latest features and functionality that are available in the current release.

Upgrading your agents

Periodically, new archive files that contain upgraded monitoring agents are available for download. Archive files are available from Products and services on the IBM Marketplace website.

Before you begin

For the following agents, an agent-specific task must be completed before you complete the upgrade procedure:

- For the agents on AIX, if you are running as a non-root user, you must clear one of the libraries from memory before you start the installation procedure to upgrade the agent. Follow the instructions in "Agents on AIX: Stopping the agent and running slibclean before you upgrade" on page 1114.
- For the HMC Base agent on AIX, if you are upgrading the agent as a non-root user, you must first stop the HMC Base agent and clear dependent libraries from cache. Follow the instructions in "HMC Base agent on AIX: Stopping the agent as a non-root user and running slibclean before you upgrade" on page 1114
- For the Microsoft .NET agent, you must remove the data collector from your .NET applications before you upgrade the agent. Follow the instructions in "Microsoft .NET agent: Removing the .NET data collector before you upgrade" on page 1117.
- For the Node.js agent, you must remove the data collector plug-ins from your Node.js applications before you upgrade the agent. Follow the instructions in "Node.js agent: Removing the data collector plug-ins before you upgrade" on page 1115.
- For the Ruby agent, you must remove the data collector from your Ruby applications before you upgrade the agent. Follow the instructions in "Ruby agent: Removing the data collector plug-ins before you upgrade" on page 1118.
- For the HTTP Server agent, you must stop the HTTP server before you upgrade the agent.
- For the WebSphere MQ agent, if you enabled transaction tracking for the agent in the previous release, you must stop the agent instance before you upgrade the agent.
- For the SAP NetWeaver Java Stack agent, if you are upgrading from V8.1.3.2 to V8.1.4 then stop all the SAP NetWeaver Java Stack instances that are configured with the data collector before you upgrade the agent.
- For the Skype for Business Server agent, if you are upgrading from the older version to V8.1.4.0.2 then at the agent side, the agent name changes to Skype for Business Server. Also, after the support upgrade through SDA, you need to restart the APMUI service to reflect the new agent name (Skype for Business Server) at the MIN Server side or else you will see the old agent name (MS Lync Server) on the MIN Server dashboard.
- For Tomcat agent, if you want to upgrade TEMA core framework on Windows, you must stop both agent and server. Follow the instructions in Tomcat agent: Upgrading the TEMA Core Framework on Windows

About this task

If a new version of the agent is available, running the installation script automatically upgrades the agent. If the agent does not have a newer version available, a message is displayed explaining that the agent is already installed; your installed agent is not affected.

To install an upgraded agent, use the following procedures:

Procedure

- “Installing agents on UNIX systems” on page 122
- “Installing agents on Linux systems” on page 128
- “Installing agents on Windows systems” on page 137

Results

The agent is upgraded to the latest version. If a newer version of the monitoring agent is not available, a message is displayed explaining that the agent is already installed; your installed agent is not affected.

What to do next

After an upgrade of a Windows agent, you must restart any agent that is not both automatically configured and started by the Windows Installer. Run the following command to check the agent status:

```
./name-agent.bat status
```

Use one of the following methods to start the agent:

- Click **Start** > **All Programs** > **IBM Monitoring agents** > **IBM Performance Management**. Right-click on an agent and click **Start**.
- Run the following command:

  ```
  ./name-agent.bat start
  ```

For information about the monitoring agent commands, including the name to use, how to check agent status, and more, see “Using agent commands” on page 177. For information about which agents are started automatically and manually, see Chapter 5, “Agent and data collector deployment,” on page 113

- For the Hadoop agent, complete the following steps after you upgrade from the socket - based agent (8.1.2, Fix Pack 2, or earlier) to the REST API - based agent (8.1.3, or later):

  1. To prevent generation of unnecessary logs, remove the 17-line code from the `hadoop-metrics2.properties` files of all Hadoop nodes.
  2. Stop the Hadoop services.
  3. Delete the `Plugin.jar` file that was copied from the agent installer from all nodes in the Hadoop cluster.
  4. Start the Hadoop services.

  For information about the 17-line code and the `Plugin.jar` file, see Configuring Hadoop nodes.
- For the HMC Base agent, after you upgrade the agent from version 6.2.2.6 to 6.2.2.7, you must configure the agent again and restart the agent. For instructions, see “Configuring HMC Base monitoring” on page 260.
- For the HTTP Server agent, if you upgrade the agent from a version that is earlier than 1.0.0.4 to 1.0.0.4 or later, you must also update the `.conf` file, which is used by the HTTP Server, to replace the previous data collector configuration file with the newly generated file. You must also add the new agent instance to the console. For instructions, see “Configuring HTTP Server monitoring” on page 265.
- For the Microsoft .NET agent, after you upgrade the agent, configure the data collector. For instructions, see “Registering the data collector” on page 522.

  If you installed the agent into a new directory, you must change the Profiler service bin path by using the service controller (`sc`) command. For example,

  ```
  sc \\localhost config DotNetProfilerService binPath=
  "${install_dir}\qe\bin\DotNetProfilerService.exe
  ```

  where *install_dir* is the new installation directory.
- For the Node.js agent, after you upgrade the agent, configure the agent data collectors. For instructions, see “Configuring the Node.js agent” on page 585.

- For the OpenStack agent, to further configure the agent to use OpenStack identity API v3, reconfigure all agent instances and update the agent data collector configuration file. For instructions, see "OpenStack agent: Reconfiguring agent instances to use OpenStack identity API v3" on page 1117.
- For the Ruby agent, after you upgrade the agent, configure the data collector. For instructions, see "Configuring the diagnostics data collector" on page 711.
- For the WebSphere Applications agent, after you upgrade the agent, migrate the data collector by running the command `dc_home/bin/migrate.sh/bat` from the installation directory of the new version of the agent and restart the application server instance. For instructions, see "WebSphere Applications agent: Migrating the data collector" on page 1118.
- Linux If you want to upgrade an older version of the agent that is installed in the `/opt/ibm/ccm/agent` directory you must complete these steps on the Linux system:

  1. If you confirm that you want to migrate the agent configuration from the old installation directory `/opt/ibm/ccm/agent` to the new installation directory, for example, `/opt/ibm/apm/agent`, you must start the agent in the new installation location.

     Restriction: The older version of the agent is stopped automatically in the old installation location but it is not started automatically in the new installation location.
  2. After you verify that the agent works in the new installation directory, you must uninstall the older version of the agent from the `/opt/ibm/ccm/agent` directory. If you want to remove all agents, run the **`/opt/ibm/ccm/agent/bin/smai-agent.sh uninstall_all`** command.
- Linux If you are upgrading agents from FP6 or earlier, after you complete the upgrade of the agents to a new directory and the configuration or reconfiguration of the agents, you might want to remove the old installation directory. Complete these steps:

  1. On the VM or system where the monitoring agent (or agents) is installed, start a command line and change to the binary folder in the old installation directory, `/opt/ibm/ccm/agent/bin`.
  2. To uninstall all monitoring agents that were installed from the old installation directory, enter: `./smai-agent.sh uninstall_all`
  3. Delete the old installation directory.

Preserving agent configuration changes

Advanced users can apply override values to component customization. Applying override values ensures that values are retained during an upgrade. Test the changes in your environment first before you apply them globally.

About this task

- These instructions are for Linux and AIX agents. For a list of the agent product codes and the commands for stopping and starting the agents, see "Using agent commands" on page 177.
- The Windows agent process preserves configuration changes by design: Updated variables in the `k`*pc*`cma.ini` file, where *pc* is the product code, are kept in the Override Local Settings section. These variables are used during each configuration to update the Windows registry entries that the agents use at run time.
- The customized settings in the `.`*pc*`.environment` file and `.global.environment` file are lost after agent upgrade. To preserve your settings, make customization changes in the *pc*`.environment` and `global.environment` files. The settings in these files are not overwritten by agent upgrade.

Procedure

Take the following steps to save the configuration changes that were made to the environment file and preserve them after agent upgrade:

1. Create or update any of the following files as needed, where *`install_dir`* is the agent installation directory (such as the Linux default `/opt/ibm/apm/agent/`or AIX default `/opt/ibm/ccm/agent/`):

File name	Description
***install_dir*/config/*pc*.environment**	The *pc* in the file name is the agent product code, such as `mq` or `rz`.
***install_dir*/config/global.environment**	Update the global environment file for changes that you want to affect all agent types.

For example, `as.environment` is the persistent WebSphere Applications agent environment file. The `.as.environment` is overwritten when the agent is upgraded to a new version.

Define variables in the *key=value* format where *key* is the environment variable name and *value* is the value or setting (such as **`KDC_FAMILIES=${KDC_FAMILIES}HTTP:10001`**).

2. After you are finished updating the variable settings, save and close the environment file and restart the affected agents.

Results

The updates are applied to all agents of the same type or, if you updated the global environment file, to all agents that report to the Cloud APM server. The changes are persisted with agent version upgrades.

Agents on AIX: Stopping the agent and running `slibclean` before you upgrade

If you are upgrading an agent as a non-root user on AIX systems, you must complete this task. Before you run the agent installer, you must stop the agent and run **`slibclean`** to clear the `libkududp.`a library.

Procedure

1. Stop the agent by running one of the following commands, depending on whether the agent supports multiple instances:

   -
     ```
     ./name-agent.sh stop
     ```

   -
     ```
     ./name-agent.sh stop instance_name
     ```

   See “Using agent commands” on page 177.
2. Run the following command with root user privileges.

   ```
   slibclean
   ```

   See slibclean Command in the IBM Knowledge Center.

Results

The agent is stopped and the `libkududp.`a library is cleared.

What to do next

Run the agent installer to upgrade the agent to the release that you have downloaded. See Chapter 6, “Installing your agents,” on page 121. If the upgrade fails, reboot the server and repeat the procedure.

HMC Base agent on AIX: Stopping the agent as a non-root user and running `slibclean` before you upgrade

Before you upgrade the HMC Base agent as a non-root user on AIX, you must stop the HMC Base agent and run **`slibclean`** to clear dependent libraries from cache.

About this task

Procedure

1. Run the following command as the non-root user to stop the agent.

```
hmc_base-agent.sh stop
```

2. Run the following command with root user privileges.

   ```
   slibclean
   ```

   See slibclean Command in the IBM Knowledge Center.

Results

The HMC Base agent is stopped and the dependent libraries are cleared.

What to do next

Run the agent installer to upgrade the HMC Base agent.

Node.js agent: Removing the data collector plug-ins before you upgrade

Before you upgrade the Node.js agent, you must remove the monitoring plug-ins from your Node.js application.

About this task

Based on your Node.js agent version, you need to complete different procedure to remove the monitoring plug-ins from your Node.js application. To find out the agent version, see Agent Version Command.

Procedure

1. Remove data collector plug-ins from the beginning of the Node.js application file.
   - If you upgrade the Node.js agent from V01.00.12.00 to V01.00.13.00, complete the following procedure:
     - If you enabled resource data collection, remove the following line from the beginning of the Node.js application file:

       ```
       require('KNJ_NPM_LIB_LOCATION/node_modules/ibm-apm/knj_index.js');
       ```

       where *KNJ_NPM_LIB_LOCATION* is the directory to the `lib` folder of your npm package global installation directory. The default directory is `/usr/local/lib`.
     - If you enabled resource data collection and deep-dive diagnostics data collection, remove the following line from the beginning of the Node.js application file:

       ```
       require('KNJ_NPM_LIB_LOCATION/node_modules/ibm-apm/knj_deepdive.js');
       ```
     - If you enabled resource data collection, deep-dive diagnostics data collection, and method traces collection, remove the following line from the beginning of the Node.js application file:

       ```
       require('KNJ_NPM_LIB_LOCATION/node_modules/ibm-apm/knj_methodtrace.js');
       ```
   - If you upgrade the Node.js agent from V01.00.10.00 to V01.00.13.00, complete the following procedure:
     - If you enabled resource data collection, remove the following line from the beginning of the Node.js application file.

       ```
       require('install_dir/lx8266/nj/bin/plugin/knj_index.js');
       ```

       , where *install_dir* is the installation directory of Node.js agent.
     - If you enabled resource data collection and deep-dive diagnostics data collection, remove the following line from the beginning of the Node.js application file.

       ```
       require('install_dir/lx8266/nj/bin/plugin/knj_deepdive.js');
       ```

    - If you enabled resource data collection, deep-dive diagnostics data collection, and method traces collection, remove the following line from the beginning of the Node.js application file.

      ```
      require('install_dir/lx8266/nj/bin/plugin/knj_methodtrace.js');
      ```

2. Restart your Node.js application to disable the data collector plug-ins.
   - If the version of your current Node.js agent is V01.00.10.00, till now the data collector plug-ins are successfully removed.
   - If the version of your current Node.js agent is V01.00.12.00, continue to the following step.
3. Run the `./uninstall.sh` command from the *install_dir*`/lx8266/nj/bin` directory to remove your previous agent settings.

What to do next

Upgrade the Node.js agent. See "Upgrading your agents" on page 1111.

Response Time Monitoring agent: upgrading IBM HTTP Server Response Time module

If you were previously monitoring IBM HTTP Server using IBM HTTP Server Response Time module or HTTP Server agent, upgrade your installation.

About this task

The following table shows some installation scenarios that might be similar to the way you are monitoring IBM HTTP Server.

Response Time Monitoring agent	Using IBM HTTP Server Response Time module?	Using Packet Analyzer?	HTTP Server agent installed?
AIX and xLinux: 08.11.00 and later Windows: 08.14.02 and later	✓	—	✓
AIX and xLinux: 08.10.00	✓	—	—
AIX and xLinux: 08.10.00	✓	—	✓
7.40.07 or earlier	—	✓	—
7.40.07 or earlier	—	✓	✓

For all of these scenarios, the installation process is similar.

Procedure

1. Install the HTTP Server agent from release V8.1.1 or later on AIX or Linux; from release V8.1.4.02 or later on Windows.

   The IBM HTTP Server Response Time module is automatically installed with the agent.
2. Configure the HTTP Server agent.

   Note: If you were previously using IBM HTTP Server Response Time module, update the web server configuration file (`httpd.conf`) with the location of the new IBM HTTP Server Response Time module and remove the old load module configuration file (`mod_wrt.so`).

   Note: Response Time Monitoring agent V8.1.1 and later does not work with the load module file (`mod_wrt.so`) from previous releases. If you attempt to use an old version of this file, error log messages are created. Transactions might still be tracked, but transaction instance data will not be displayed.

For more information, see the HTTP Server agent reference PDF, which you can download from http://ibm.biz/agent-httpserver.

3. Ensure that the IBM HTTP Server and HTTP Server agent are running. If the Response Time Monitoring installer detects the HTTP Server agent, the Response Time Monitoring agent enables the IBM HTTP Server Response Time module rather than the Packet Analyzer.
4. Install the Response Time Monitoring agent to the same location `AGENT_HOME` as the HTTP Server agent.
   - Linux AIX Install V8.1.1 or later as **`root`**. `AGENT_HOME` example `/opt/ibm/apm/agent/`
   - Windows Install V8.1.4.0.2 or later with administrator permissions. `AGENT_HOME` example `C:\IBM\APM\`.
5. If you used the Packet Analyzer in earlier releases, you might need to disable the Packet Analyzer to start monitoring IBM HTTP Server with IBM HTTP Server Response Time module.
6. Restart IBM HTTP Server.

Microsoft .NET agent: Removing the .NET data collector before you upgrade

Before you upgrade the Microsoft .NET agent, you must remove the .NET data collector from your .NET applications.

Procedure

1. Unregister all modules of the data collector.

   As an administrator, enter:

   ```
   cd install_dir\qe\bin configdc unregisterdc all
   ```

   Where *install_dir* is the installation directory of the Microsoft .NET agent.
2. Restart the .NET applications.

What to do next

Upgrade the Microsoft .NET agent. See "Upgrading your agents" on page 1111.

OpenStack agent: Reconfiguring agent instances to use OpenStack identity API v3

To upgrade the OpenStack agent to use OpenStack identity API v3, after you install the latest version of the agent, you must reconfigure all agent instances and update the data collector configuration file.

About this task

This task is mandatory only when you upgrade the agent to use OpenStack identity API v3.

Procedure

1. Reconfigure all existing agent instances. For detailed instructions, see "Configuring the OpenStack agent" on page 605.
2. Find the `ksg_dc_`*`instance_name`*`.cfg` agent data collector configuration file, where *instance_name* is the name you specified for your agent instance.

   If the file does not exist, copy *`install_dir`*`/lx8266/sg/bin/ksg_dc.cfg` to the *`install_dir`*`/config` directory and change the file name to `ksg_dc_`*`instance_name`*`.cfg`.

   For example, if the instance name is `OS1`, change the name to `ksg_dc_OS1.cfg`.
3. Add the following section to the `ksg_dc_`*`instance_name`*`.cfg` file:

   ```
   #OpenStack authentication information
   [OS_authentication_info]
   OS_project_domain_name=Default
   ```

```
OS_user_domain_name=Default
OS_cert_path=
```

4. Restart the agent instance by running the following commands:

   ```
   install_dir/bin/openstack-agent.sh stop instance_name
   install_dir/bin/openstack-agent.sh start instance_name
   ```

   where *instance_name* is the name of the agent instance to be configured.

Ruby agent: Removing the data collector plug-ins before you upgrade

Before you upgrade the Ruby agent, you must remove the monitoring plug-ins from your Ruby application.

Procedure

1. Remove the old version data collector by running the following command.

   ```
   gem uninstall stacktracer
   ```

2. Navigate to the home directory of your application, open its `Gemfile`, and remove the following line:
   `gem 'stacktracer', 'version'`
   Where *version* is the version number of the Ruby agent.
3. In the home directory of your application, enter: `bundle install`

What to do next

Upgrade the Ruby agent. See "Upgrading your agents" on page 1111.

WebSphere Applications agent: Migrating the data collector

After you update the agent, you must migrate the data collector either interactively or in silent mode.

Migrating the data collector interactively

You can migrate an earlier maintenance level of the data collector interactively using the migration utility.

Before you begin

Linux AIX If you installed the WebSphere Application Server or WebSphere Portal Server using a non-root user account, before you run the configuration utilities, verify that the non-root user has read and write privileges to the following agent directories in *install_dir*/yndchome/7.3.0.14.08 where *install_dir* is the installation directory of the WebSphere Applications agent:

- `data`
- `bin`
- `runtime`
- `logs`

Provide read and write permissions using the `chmod 777` command, if required. Also, log in as the user that was used to install the application server.

About this task

You can migrate an earlier maintenance level of the data collector interactively using the migration utility. If you want to migrate many application server instances, it might be more convenient to use the migration utility in silent mode.

Important:

- You can only migrate previous maintenance levels of 7.3 version of a data collector. The version of the data collector is indicated in the data collector home directory path.

- You cannot migrate from the data collector version 7.3 to 7.3 fix pack 1. Instead, unconfigure the data collector and uninstall the agent version 7.3. Then install the agent version 7.3 fix pack 1 and configure the data collector again.

Procedure

1. Linux AIX Log in as the user that was used to install the application server.
2. Start the migration utility from the installation directory of the latest version of the agent.

   Linux AIX Run the command *dc_home*`/bin/migrate.sh`

   Windows Run the command *dc_home*`\bin\migrate.bat`
3. The utility displays the IP addresses of all network cards that are found on the local computer system.

   Enter the number that corresponds to the IP address to use.
4. The utility discovers all servers configured by older maintenance levels of the data collector and lists them. The data collectors are grouped by maintenance level.

   Select one or more application server instances from the list.

   The list might include both traditional WebSphere server instances and Liberty servers. Traditional WebSphere server instances might be under different profiles.

   Tip:

   - If several instances under one profile are monitored, you must select them all for migrating at the same time.
   - Migrate all servers under the liberty profile at the same time. Partially migrating configured servers might cause instability.

   Remember:

   - For a stand-alone environment, application server instances must be running.
   - For a Network Deployment environment, the node agent and deployment manager must be running.
   - Liberty servers are not required to be running during the migration.
5. Enter the number that corresponds to the application server instance whose data collector is to be migrated or enter an asterisk (*) to migrate the data collector of all application server instances.

   To specify a subset of servers, enter the numbers, separated by commas, that represents the servers. For example: `1,2,3`.

   The migration utility automatically integrates each data collector with the monitoring agent. The monitoring agent host and port values are retrieved from the existing configuration files.
6. Enter an alias for each of the selected servers.

   The default value is the existing server alias.
7. For the Liberty server instance, enter the JVM home directory when prompted. For example, `/opt/IBM/java`.
8. The utility determines whether WebSphere global security is enabled for each of the profiles where data collection is being migrated.

   If WebSphere Global Security is enabled for one or more profiles, specify whether to retrieve security settings from a client properties file:

   The data collector communicates with the WebSphere Administrative Services using the RMI or the SOAP protocol. If global security is enabled for a profile, you must specify the user ID and password of a user who is authorized to log in to the IBM WebSphere Application Server administrative console for the profile.

Alternatively, you can encrypt the user name and password and store them in the application server client properties files before configuring the data collector. You must use the `sas.client.props` file for an RMI connection, or the `soap.client.props` file for an SOA connection.

9. Enter 1 to allow the utility to retrieve the user name and password from the appropriate client properties file and skip to step "11" on page 1120. Otherwise, enter 2 to enter the user name and password.

   Important: It may take some time to log in to the WebSphere Application Server administrative console.

10. Enter the user name and password for each profile whether WebSphere Global Security is enabled.
11. The utility migrates data collection for each selected application server instance. It displays a status message that indicates whether the migration of each server completed successfully.
12. Restart the instances as directed by the utility. The data collector configuration takes effect when the application server instances are restarted.

Results

The data collector is migrated to the latest maintenance level installed.

What to do next

The migration utility preserves the settings that were configured in the older version of the data collector. To modify these settings, you can run either the configuration or reconfiguration utility in interactive or silent mode from the *dc_home*`\bin` directory of the new data collector. For more information, see "Configuring or reconfiguring the data collector with full configuration utilities" on page 834 .

Migrating the data collector in silent mode

You can migrate an earlier maintenance level of the data collector using the migration utility in silent mode.

Before you begin

Linux AIX If you installed the WebSphere Application Server or WebSphere Portal Server using a non-root user account, before you run the configuration utilities, verify that the non-root user has read and write privileges to the following agent directories in *install_dir*`/yndchome/7.3.0.14.08` where *install_dir* is the installation directory of the WebSphere Applications agent:

- `data`
- `bin`
- `runtime`
- `logs`

Provide read and write permissions using the `chmod 777` command, if required. Also, log in as the user that was used to install the application server.

About this task

A sample silent properties file, `sample_silent_migrate.txt`, is packaged with the migration utility. The file is available in the *install_dir*`/yndchome/7.3.0.14.08/bin` directory.

When you create your silent properties file, keep in mind these considerations:

- A line in the file that starts with a number sign (#) is treated as a comment, and is not processed. If the number sign is used elsewhere in the line, it is not considered to be the start of a comment. This means that you can use the number sign in passwords or for other uses.
- Each property is described on a separate line, in the following format: property = value.

  property

  This is the name of property. The list of valid properties that you can configure is shown in Table 258 on page 1121. Do not modify or remove properties in the sample file that are not listed in the table.

value

This is the value of the property. Default values for some properties are already provided. You can delete default values to leave property values blank, or empty. An empty value is treated as if the property is not specified, as opposed to using the default value. If you want to use default values, you can comment out the property in the file.

- Passwords are in plain text.
- Properties and their values are case-sensitive.

Table 258 on page 1121 describes the properties that are available when migrating the data collector in silent mode.

Table 258. Available properties for running the migration utility in silent mode

Property	Comment
migrate.type	Must be AD.
default.hostip	If the computer system uses multiple IP addresses, specify the IP address for the data collector to use.
itcam.migrate.home	Specifies the data collector home directory of the older maintenance version of the data collector. The directory is not deleted as part of the migration.
was.wsadmin.connection.host	Specifies the name of the host to which the wsadmin tool is connecting. In a Network Deployment environment, specify the wsadmin connection to the Deployment Manger. In a stand-alone environment, specify the wsadmin connection to the server.
was.wsadmin.username	Specifies the user ID of a user who is authorized to log on to the IBM WebSphere Application Server administrative console. This user must have the agent role on the application server.
was.wsadmin.password	Specifies the password that corresponds to the user specified in the `was.wsadmin.username` property.
was.appserver.profile.name	Specifies the name of the application server profile you want to configure. **Remember:** The property is not required for a Liberty profile.
was.appserver.home	Specifies the WebSphere Application Server home directory.
was.appserver.cell.name	Specifies the WebSphere Application Server cell name. **Remember:** The property is not required for a Liberty profile.
was.appserver.node.name	Specifies the WebSphere Application Server node name. **Remember:** The property is not required for a Liberty profile.

Table 258. Available properties for running the migration utility in silent mode (continued)

Property	Comment
was.appserver.server.name	Specifies the application server instance within the application server profile to migrate to the new version of the data collector. The silent properties file can have multiple instances of this property.

Important:

- You can only migrate previous maintenance levels of 7.3 version of a data collector. The version of the data collector is indicated in the data collector home directory path.
- You cannot migrate from the data collector version 7.3 to 7.3 fix pack 1. Instead, unconfigure the data collector and uninstall the agent version 7.3. Then install the agent version 7.3 fix pack 1 and configure the data collector again.

Procedure

1. Specify configuration options in the silent migration properties file.
2. Run the command to start the migration utility in silent mode from the installation directory of the latest version of the agent.
   - Linux AIX *dc_home*`/bin/migrate.sh -silent` *sample_silent_migration_filename*
   - Windows *dc_home*`\bin\migrate.bat -silent` *sample_silent_migration_filename*

Results

The data collector is migrated to the latest maintenance level installed.

What to do next

The migration utility preserves the settings that were configured in the older version of the data collector. To modify these settings, you can run either the configuration or reconfiguration utility in interactive or silent mode from the *dc_home*`\bin` directory of the new data collector. For more information, see "Configuring or reconfiguring the data collector with full configuration utilities" on page 834.

Tomcat agent: Upgrading the TEMA Core Framework on Windows

To upgrade TEMA core framework on windows for Tomcat agent, you must stop both the agent and server to upgrade the TEMA framework successfully.

Procedure

1. Prepare the Tomcat server setup.
2. Install and configure the Tomcat agent.
3. Log in to the IBM Cloud Application Performance Management dashboard, goto **Agent configuration** > **Tomcat**, select an instance of Tomcat agent, and click **Enable TT/DD**.
4. Restart the Tomcat Server.
5. To apply IBM APM CORE FRAMEWORK, stop both Tomcat Agent and Server.
6. Goto TEMA/`<IBM APM CORE FRAMEWORK_HOME>`. Run the command **`apmpatch.bat <Tomcat Agent Installationdir>`**.
   The framework is upgraded.
7. Check the upgraded IBM APM CORE FRAMEWORK version by executing the following instructions.
   ```
   Goto <TOMCAT_Agent_Install_Dir>\InstallITM
   Run: KinCInfo.exe -i.
   ```
8. Start both Tomcat server and agent.

Upgrading your data collectors

Periodically, new archive files that contain upgraded data collectors are available for download. Archive files are available from Products and services on the IBM Marketplace website..

Before you begin

About this task

To upgrade a data collector, complete the following steps:

Procedure

- Unconfigure the data collector from your local and/or IBM Cloud applications:
  - For the J2SE data collector, unconfiguration steps are not required.
  - For the Liberty data collector, follow the instructions in "Unconfiguring the data collector for IBM Cloud applications" on page 473 and/or "Unconfiguring the data collector for on-premises applications" on page 467.
  - For theNode.js data collector, follow the instructions in "Unconfiguring the stand-alone Node.js data collector for IBM Cloud applications" on page 594 and/or "Unconfiguring the stand-alone Node.js data collector for on-premises applications" on page 599.
  - For thePython data collector, follow the instructions in "Unconfiguring the Python data collector for IBM Cloud applications" on page 668 and/or "Unconfiguring the Python data collector for on-premises applications" on page 673.
  - For theRuby data collector, follow the instructions in "Unconfiguring the Ruby data collector for IBM Cloud applications" on page 718.
- Download the data collector package.
- Reconfigure the data collector to monitor your local and/or IBM Cloud applications:
  - For the Node.js data collector, after you upgrade the data collector, reconfigure the data collector. For instructions, see "Configuring the stand-alone Node.js data collector for IBM Cloud(formerly Bluemix) applications" on page 590 and/or "Configuring the stand-alone Node.js data collector for on-premises applications" on page 595.
  - For the Python data collector, after you upgrade the data collector, reconfigure the data collector. For instructions, see "Configuring the Python data collector for IBM Cloud applications" on page 664 and/or "Configuring the Python data collector for on-premises applications" on page 669.
  - For the Liberty data collector, after you upgrade the data collector, reconfigure the data collector. For instructions, see "Configuring the Liberty data collector in IBM Cloud environment (Liberty V18.* and older versions)" on page 468 and/or "Configuring the Liberty data collector in on-premises environments (Liberty V18.* and older versions)" on page 464.
  - For the J2SE data collector, after you upgrade the data collector, reconfigure the data collector. For instructions, see "Configuring J2SE monitoring" on page 442.
  - For the Ruby data collector, after you upgrade the data collector, reconfigure the data collector. For instructions, see "Configuring the Ruby data collector for IBM Cloud applications" on page 715.

Results

The data collector is upgraded to the latest version.

Chapter 12. Troubleshooting and support

Review the troubleshooting entries for problems that you might experience with installing, configuring, or using IBM Cloud Application Performance Management.

Troubleshooting content is available in this Knowledge Center. Previously, troubleshooting content was available in the Cloud Application Performance Management forum on IBM developerWorks. Going forward, the IBM developerWorks platform where this forum resides is not available. The troubleshooting content from the forum is now ported into this IBM Cloud Application Performance Management Troubleshooting Guide.

For IBM Cloud Application Performance Management Hybrid Gateway troubleshooting, see “Managing the Hybrid Gateway” on page 943.

Troubleshooting agents

Troubleshoot agent installation and configuration issues.

We are migrating our troubleshooting content from the Cloud Application Performance Management Forum in developerWorks to this Knowledge Center. Previously, troubleshooting content was available in the Cloud Application Performance Management Forum on developerWorks. You can continue to search for older entries in this forum. Search for entries that start with "Troubleshooting".

Internet Service Monitoring

You may find here more details about Internet Service Monitoring known issues.

Profile will not be created after create profile page is kept open for more than 10 minutes and another profile with same name will not be created

Problem

Profile will not be created after **create profile page** is kept open for more than 10 minutes and another profile with same name will not be created.

Symptom

While creating a profile if the user keeps **create profile page** idle (open without any activity) for more than 10 minutes, and then tries to create the profile, it will not get created. After that, if user tries to create the profile again with the same name, profile will not get created.

Cause

A lock file gets created at the MIN side at the time of profile creation which locks the profile creation activity for the same profile for other users. The lock file gets deleted after profile creation is done. But if the create window is idle for more than 10 minutes, the create event gets locked, and user is not able to create the profile.

Solution

- User should not keep the window idle for more than 10 minutes while creating a profile.
- Administrator can delete the lock file of the profile that was created from MIN side at `/opt/ibm/wlp/usr/servers/min/dropins/CentralConfigurationServer.war/data_source/is`.
  For example, if the profile name is ABC, a lock file `$$ABC$$1UjQ9wy1boIHTAQeoWSj1IU.lock` gets created.

Data is displayed for deactivated (Active field is false) monitor elements on APM UI on windows platform

Problem

Data is displayed for deactivated (Active field is false) monitor elements on APM UI on windows platform.

Symptom

While adding monitor elements if active field is kept as false and the monitor elements is deployed on agent machine, data for that element is getting collected initially and is shown on the dashboard. This data from the dashboard is then automatically vanishing after some time as the active field is false. This issue is occurring only on windows platform.

Cause

In the monitor code active fields value is randomly becoming true even if it is set as false in monitor xml. Hence, monitor is collecting the data.

Solution

Data will go automatically from the dashboard after some time. User doesn't need to do anything.

`custom.properties` file is getting empty when agent is offline for more than 10 minutes and no profile is deployed after the agent is upgraded to 8.1.4.0.11

Problem

When agent is upgraded to 814011 version from older versions, contents of the file `custom.properties` are not getting stored to CCS DB automatically. Hence, `custom.properties` file is getting empty when agent is offline for more than 10 minutes and no profile is deployed after the agent is upgraded to 8.1.4.0.11.

Symptom

Due to the `custom.properties` file getting empty when agent is offline for more than 10 minutes and no profile is deployed after the agent is upgraded to 8.1.4.0.11, data will not be shown in the dashboard unless user redeploy the profiles.

Cause

CCS DB is updated only when user deploys or renames profiles. CCS DB will not be updated unless user deploys or renames the profile after upgrade.

Solution

Deploy at least one profile after upgrade so that CCS DB will be updated with latest configuration from `custom.properties` file.

Discrepancy in data in the monitor XML

Problem

Discrepancy in data in the monitor XML that is present at `ISMHOME/profiles/active` of agent. It occurs when $ is used or entered in any field while configuration of profiles for all the monitors.

Solution

Refrain from using or entering $ in any field while configuration of profiles for all the monitors to avoid this issue.

303 status code default data validation condition is not visible under the DVC tab in old profiles for http and https monitor elements after agent upgrade.

Problem

303 status code default data validation condition is not visible under the DVC tab in old profiles for http and https monitor elements after agent upgrade.

Symptom

HTTP and HTTPS URL responses are not evaluated for 303 status codes.

Cause

The page is already loaded for the old profiles, hence after upgrading the agent, when the profile is opened in edit mode, the old data gets fetched. Hence 303 status condition does not get added after agent upgrade.

Solution

Add the 303 status code condition under the DVC tab manually for old profiles containing http and https monitor elements as follows:

Metric	**Operator**	**Operand**	**Status**
status	!=	303	FAILED

Microsoft Active Directory Monitoring

You may find here more details about Microsoft Active Directory monitoring known issues.

Microsoft Active Directory agent does not show updated Online help content

Problem

Online help pages are not updated with the latest content for Microsoft Active Directory agent.

Symptom

On the APM dashboard Eclipse help for Microsoft Active Directory agent, the help content is missing for Data collection interval and retention period of the following newly added attribute groups:

- Directory Services
- Kerberos Consistency Checker
- Kerberos Key Distribution Center
- Name Service Provider
- Exchange Directory Service

Cause

The problem occurs due to constraint in the build server.

Solution

User may find the help content in the respective attribute groups contextual help on the APM dashboard.

Note: The issue appears in APM V8.1.4.10 release.

Microsoft IIS monitoring

You may find here more details about Microsoft Internet Information Services known issues.

Online help pages are not updated with latest content for Microsoft IIS APM agent

Problem

Online help pages are not updated with latest content for Microsoft IIS APM agent

Symptom

Newly added attribute groups are missing in the Online help content:

- WPROCESS
- MEMIISUS
- ASP Garbage Collection
- IISSVRINFO

Cause

This problem is occurring due to build server issues.

Workaround

Not Available. However, you can see the help contents of particular attribute group on APM dashboard.

When user installs IIS agent, prerequisite checker scanner fails to install and displays an error message

Problem

When user installs IIS agent, prerequisite checker scanner fails to install and displays below error message: “Scenario: Prerequisite Scan KQ7 – KQ7 [version 08200300]:

Table 259.

Property	Result	Found	Expected
zDotNetFrameworkVersion	FAIL	[Not found]	4.6 Onwards (Release DWORD: 393295)

Overall Result: FAIL"

Solution

Refer the steps to resolve this issue

1. After downloading and extracting the installation files, bypass the prerequisite scan.
2. To bypass the prerequisite scan run the following command in CLI while IIS agent installation `export SET SKIP_PRECHECK=Y`

Microsoft .NET Monitoring

You may find here more details about Microsoft .NET monitoring known issues.

Microsoft .NET agent does not show updated Online help content

Problem

Online help pages are not updated with the latest content for Microsoft .NET agent.

Symptom

On the APM dashboard Eclipse help for Microsoft .NET agent, the Request Name attribute help content is missing under the Database Call Details attribute group.

Cause

The problem occurs due to constraint in the build server.

Solution

User may find the help content in the Database Call Details attribute group widget contextual help on the APM dashboard.

Note: The issue appears in APM V8.1.4.10 release.

System throws an exception when you browse any DB and Windows Communication Foundation related application

Problem

You get the following exception when you browse any database and Windows Communication Foundation (WCF) related application:

```
Could not load file or assembly 'KTJdotnetdc, Version=8.200.3.0, Culture=neutral,
PublicKeyToken=911d32f28c3a3213' or one of its dependencies. The system cannot find the file
specified
```

Cause

The issue occurs due to a mismatch between agent version and agent DLL code version.

Solution

To resolve the issue, follow these steps:

1. Stop the agent service from IBM Performance Management.
2. Stop the Microsoft Internet Information Services (IIS) by running the following command: **`iisreset/stop`**
3. Stop the DotNetProfilerService service by running the following command: **`net stop DotNetProfilerService`**
4. Unregister all the modules, such as HTTP, ISAPI, SOAP, Profiler, and RTModule, if they are registered or enabled.
5. Copy the `CorProfLog.dll` file from `\IBM\APM\TMAITM6_x64` and replace it in the `{APM_HOME}\qe\bin` path.
6. Register all the modules that you unregistered in step 4 and start all the services, for example, Microsoft .NET agent, IIS, and DotNetProfilerService.

Note: The issue appears in APM V8.1.4.0.11 release.

No items are displayed in Attribute Details tab of attribute group KQE_SERVICEDETAILS

Problem

There are no items displayed in the **Attribute Details** tab of attribute group **KQE_SERVICEDETAILS**.

Symptom

In the **Attribute Details** tab of attribute group **KQE_SERVICEDETAILS**, the message `No items to display` is showed.

Cause

This attribute group is implemented for **Monitoring** of **IBM Cloud Pak for Multicloud Management**.

Solution

Not applicable.

Microsoft Skype for Business Server Monitoring

You can find here more details about Microsoft Skype for Business Server monitoring known issues.

The Skype topology widget does not display data after the agent upgrade

When you upgrade the Skype for Business Server agent from v8.1.4.0.11 to v8.1.4.0.12, the `Skype Topology` group widget does not display data on the APM UI.

Problem

The Skype topology widget does not display data after the agent upgrade.

Symptom

On the Overview page, the `Skype  Topology` group widget does not display data.

Cause

After the agent upgrade, the `KQL_LyncTop_Enable` variable value is not updated from false to true.

Solution

To display the `Skype  Topology` group widget data after the agent upgrade, follow these steps:

1. Navigate to `<CANDLE_HOME>\TMAITM6_x64` or right click on **Monitoring Agent for Skype for Business Server -> Advanced -> Edit ENV File**. The `KQLENV` file opens.
2. Change the variable *KQL_LyncTop_Enable* value from false to true.
3. Restart the agent.

Note: The issue appears in APM V8.1.4.12 release.

Microsoft SharePoint Server Monitoring

You can find here more details about Microsoft SharePoint Server monitoring known issues.

Microsoft SharePoint Server agent does not show updated Online help content

Problem

Online help pages are not updated with the latest content for Microsoft SharePoint Server agent.

Symptom

On the APM dashboard Eclipse help for Microsoft SharePoint Server agent, the help content is missing for the newly added group widgets called `Last 1 Hour Trace Log Count` and `Trace Log Details`.

Cause

The problem occurs due to constraint in the build server.

Solution

User may find the help content in the respective group widgets contextual help on the APM dashboard.

Note: The issue appears in APM V8.1.4.10 release.

The Microsoft SharePoint Server agent upgrade fails with an error message

Problem

The Microsoft SharePoint Server agent upgrade fails with an error message.

Symptom

When you upgrade the Microsoft SharePoint Server agent from the previous build, the upgrade fails and gives the following error message:

```
ERROR: Failed to install Monitoring Agent for MS SharePoint Server
```

Cause

The problem occurs due to a batch file that is generated by the agent factory.

Solution

Not available. However, the agent connects to the server successfully and the APM dashboard shows the updated content. Also, this issue does not affect the SharePoint agent functionality.

Note: The issue appears in APM V8.1.4.12 release.

SharePoint Agent Configuration window does not appear while configuration if it coexists with V6 agent

Problem

If ITCAM for Microsoft Applications: Microsoft SharePoint Server Agent and Monitoring Agent for Microsoft SharePoint Server (V8) are installed on same system and if you try to reconfigure APM agent (V8) then agent configuration window does not appear and a message The `agent configuration process did not complete.` is shown.

Symptom

Agent configuration window does not open while configuring Monitoring Agent for Microsoft SharePoint Server.

Cause

While opening agent Configuration window, factory creates a temporary file `temp.ini` and writes *JAVA_HOME* variable in it as: `JAVA_HOME=Candle_Home\java\java80_x64\jre`

But if ITCAM for Microsoft Applications: Microsoft SharePoint Server Agent is already installed on the same system then factory cannot write *JAVA_HOME* variable intermittently in `temp.ini` file. Hence, configuration window fails to open.

Workaround

Configure Monitoring Agent for Microsoft SharePoint Server silently using agent configuration response file.

Microsoft SQL Server monitoring

You may find here more details about Microsoft SQL Server known issues.

Memory leak for Collector process

Problem

Memory is leaked for collector process when the perfmon counters for one or more databases is not available under the “SQLServer:Databases” perfmon object.

Symptom

The memory for collector process keeps on increasing and the memory is not released.

Cause

The SQL agent uses *perfmon* counter as source for data collection. When database is not under the "SQLServer:Databases" object in Performance Monitor(perfmon.exe), the counter value retrieval operation fails for the database. During this failure, *perfmon* counter handles are not released properly in the Collector process, resulting in a memory leak.

One of the possible causes for the why the database instance is not present under the "SQLServer:Databases" object is that the database is not in ONLINE state.

Solution

You may apply any of the following solutions to resolve the problem:

- **Configure MS SQL Agent to collect data only for databases having valid counters**:

  Refer the following guide for setting up parameter "*Database"* and disable the collection for databases which do not have database instance under SQLServer:Databases object in Performance Monitor.

  https://www.ibm.com/support/knowledgecenter/SSHLNR_8.1.4/com.ibm.pm.doc/install/sql_config_agent_parameters_database_server_properties.html
- **Troubleshoot performance counter issues for the database which do not have instance under the SQLServer:Databases object in Performance Monitor**:
  - If the database instance is not present under SQLServer:Databases object and database is not in ONLINE state, making database state ONLINE might resolve the problem.
  - Unloading and loading the counter for MS SQL Server might help. Please refer *"Resolving The Problem"* section in the following link:

    https://www.ibm.com/support/pages/mssql-agents-oq-not-collecting-data-koqcollexe-exits
- **Disable collection for Database Detail attribute group**:

  Refer following guide for setting up parameter *"Extended Parms"* to disable Database Detail Attribute Group (KOQDBD), https://www.ibm.com/support/knowledgecenter/SSHLNR_8.1.4/com.ibm.pm.doc/install/sql_config_agent_parameters_database_server_properties.html

Note: SQLServer:Databases is the name of performance object for default instance of SQL Server. For named instance the object name is `MSSQL$<instance_name>:Databases`

PostgreSQL Monitoring

You may find here more details about PostgreSQL Monitoring known issues.

Buffer Hit Percentage widget for databases with no active connections does not show help content

Problem

Information is not available for databases that do not have active connections.

Symptom

Databases with no active connections are not displayed in the Buffer Hit Percentage widget. The information should be showed in the widget help content.

Cause

Limitation due to compatibility with IBM Cloud App Management.

Solution

Not available. User shall take note of the limitation.

Memory and IP address values are not displayed on SUSE15 platform

Problem

Memory and IP address values are not displayed when agent is monitoring PostgreSQL Server locally on SUSE15 platform.

Symptom

Memory and IP address values are not displayed if agent is monitoring PostgreSQL Server on SUSE15 platform.

Cause

The command **`netstat`** fails for agent on SUSE15 platform.

Soution

User can use SUSE12 platform to monitor PostgreSQL Server locally.

No data is displayed in all of the widgets for PostgreSQL Instance resource and Database resource.

Problem

No data is displayed in all of the widgets for PostgreSQL Instance resource and Database resource.

Symptom 1

Following exception occurs in logs:

```
<CANDLE HOME>/logs/kpn_JDBC_<instance_name>_trace.log java.io.IOException: Connection to
10.46.44.18:5432 refused. Check that the hostname and port are correct and that the postmaster
is accepting TCP/IP connections.
```

Cause

Agent is not able to connect to remote PostgreSQL database.

Solution

Refer the steps given to resolve this issue:

1. Open `postgresql.conf` file located at `/var/lib/pgsql/12/data`
2. Update the **`listen_addresses`** parameter to accept the connection from remote host. For example **`listen_addresses = '*'`**
3. Restart the PostgreSQL database server
4. Restart the PostgreSQL agent

Symptom 2

Following exception occurs in logs:

```
<CANDLE HOME>/logs/kpn_JDBC_<instance_name>_trace.log java.io.IOException: FATAL: no
pg_hba.conf entry for host "<TEMA IP>", user "postgres", database "ibmdb", SSL off
```

Cause

SSL is off and agent is not able to connect to remote PostgreSQL database; OR
SSL is on and agent is not able to collect data from PostgreSQL database.

Solution

1. Stop the PostgreSQL agent.
2. Locate the postgresql.conf file at /var/lib/pgsql/<server_version>/data, where <server_version> is the PostgreSQL server version.
   Note the configuration of SSL. It can be set to ON or OFF.
3. Locate the pg_hba.conf file at /var/lib/pgsql/<server_version>/data, where <server_version> is the PostgreSQL server version.
4. Ensure the IPv6 local connection section in pg_hba.conf file is configured to accept connection from remote host. For example:

   If SSL is OFF, configure the following:

   ```
   host all all 0.0.0.0/0 md5
   ```

   If SSL is ON, configure the following:

   ```
    hostssl all all 0.0.0.0/0 md5
   ```
5. Ensure the PostgreSQL agent is configured with the latest JDBC jar.
   If the agent is not configured with the latest JDBC jar, reconfigure the PostgreSQL agent with the latest JDBC jar.
6. Restart the PostgreSQL database server if any change is made to pg_hba.conf file.
7. Start PostgreSQL agent.

SAP Monitoring

You may find here more details about SAP Monitoring known issues.

Support for new version is not getting reflected on server side when SAP Agent is upgraded from old version to new version

Problem

When SAP Agent is upgraded from old version to new version then support for new version is not getting reflected on server side.

Cause

When SAP Agent is upgraded from old version 08.20.XX.XX to new version 08.20.YY.YY then **IRA_CUSTOM_METADATA_LOCATION** parameter from <CANDLE_HOME>\TMAITM6_x64\KSAENV_XXX is not getting updated for windows platform.

Solution

When SAP Agent is upgraded from old version 08.20.XX.XX to new version 08.20.YY.YY then **IRA_CUSTOM_METADATA_LOCATION** parameter from <CANDLE_HOME>\TMAITM6_x64\KSAENV_XXX needs to be updated for windows platform.

Where XXX is SAP Agent instance that is configured.

Update the parameter **IRA_CUSTOM_METADATA_LOCATION** same as that present in <CANDLE_HOME>\TMAITM6_x64\KSAENV file.

For example: **`IRA_CUSTOM_METADATA_LOCATION=C:\IBM\SAP\tmaitm6_x64\support\sa\ksa_sda_8.2.0.0300.jar`** parameter is present in file KSAENV_XXX after agent is upgraded from 08.20.03.00 to new version 08.20.06.00

Update value of parameter as mentioned below in file KSAENV_XXX **`IRA_CUSTOM_METADATA_LOCATION=C:\IBM\SAP\tmaitm6_x64\support\sa\ksa_sda_8.2.0.0600.jar`**.

When `Name` attribute is used as a display item in threshold for `R/3_Buffer_Performance` and `R/3_Buffer_Performance_64` attribute groups, value of `Name` field may get truncated for multi-byte characters

Problem

When `Name` attribute is used as a display item in threshold for `R/3_Buffer_Performance` and `R/3_Buffer_Performance_64` attribute groups, value of `Name` field may get truncated for multi-byte characters

Cause

Length of `Name` attribute is insufficient to support multi-byte characters.

Solution

Newly added attribute `Name_U` can be used as a display item in order to avoid truncation.

WebSphere Applications monitoring

You can find more details about WebSphere Applications monitoring known issues.

Failed to install WebSphere Applications agent on Linux for IBM Z if libstdc++.so.5 is missing

Installation of WebSphere Applications agent on Linux for IBM Z fails if `libstdc++.so.5` is not installed on your system.

Symptoms

When installing the WebSphere Applications agent, you might have the following errors:

- Run the pre-check tool (SKIP_PRECHECK=0) when installing WebSphere Applications agent v07.3.0.14.10, an error message is displayed like the following example:

```
KYN - WAS Monitoring Agent [version 07301410]:
PropertyResultFoundExpected
===========================
os.lib.libstdc++_64FAILUnavailableregex{libstdc++.so.5}
```

- Skip the pre-check tool (SKIP_PRECHECK=1) when installing WebSphere Applications agent, configure the data collector for the WebSphere Applications server, and then restart the server. You can find the following error in the server log:

```
J9VMDllMain failed
JVMJ9VM135W /proc/sys/kernel/core_pattern setting "|/usr/lib/systemd/systemd-coredump %P %u
%g %s %t %c %e" specifies that core dumps are to be piped to an external program. The JVM may
be unable to locate core dumps and rename them.

JVMJ9TI001E Agent library am_ibm_16 could not be opened (libstdc++.so.5: cannot open shared
object file: No such file or directory)

JVMJ9VM015W Initialization error for library j9jvmti29(-3): JVMJ9VM009E J9VMDllMain failed
```

Cause

libstdc++.so.5 is required when installing the WebSphere Applications agent on Linux for IBM Z. On SUSE Linux Enterprise 12 and 15 for IBM Z, libstdc++.so.5 is not installed by default. The errors are caused by the missing of libstdc++.so.5 or being not correctly linked by am_ibm_16.so.

Solution

1. Install libstdc++.so.5 and check if the dependency is correctly set up in am_ibm_16.so.
2. Verify whether libstdc++.so.5 is successfully installed:

   ```
   /usr/lib64 # ls | grep libstdc
   libstdc++.so.5
   libstdc++.so.5.0.7
   ```

3. Check am_ibm_16.so if libstdc++.so.5 can be found:

   ```
   /opt/ibm/apm/agent/yndchome/7.3.0.14.10/toolkit/lib/ls3266 # ldd libam_ibm_16.so
   libstdc++.so.5 => not found
   librt.so.1 => /lib64/librt.so.1 (0x000003ff7e680000)
   libcclog_64.so => not found
   libmsg23_64.so => not found
   libcffdc_64.so => not found
   libc.so.6 => /lib64/libc.so.6 (0x000003ff7e480000)
   libpthread.so.0 => /lib64/libpthread.so.0 (0x000003ff7e400000)
   /lib/ld64.so.1 (0x000003ff7e900000)
   ```

4. If libstdc++.so.5 is not found, add the following lines into /etc/ld.so.conf file:

   ```
   /usr/lib
   /usr/lib64
   ```

5. Reboot the server.
6. Verify that libstdc++.so.5 is correctly linked.

   ```
   /opt/ibm/apm/agent/yndchome/7.3.0.14.10/toolkit/lib/ls3266 # ldd libam_ibm_16.so
   libstdc++.so.5 => /usr/lib64/libstdc++.so.5 (0x000003ff8cd80000)
   librt.so.1 => /lib64/librt.so.1 (0x000003ff8cd00000)
   libcclog_64.so => not found
   libmsg23_64.so => not found
   libcffdc_64.so => not found
   libc.so.6 => /lib64/libc.so.6 (0x000003ff8cb00000)
   libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x000003ff8ca80000)
   libpthread.so.0 => /lib64/libpthread.so.0 (0x000003ff8ca00000)
   /lib/ld64.so.1 (0x000003ff8d100000)
   ```

Now the errors will be resolved and you can successfully install and start the WebSphere Applications agent.

Agent installation and configuration failed on RHEL 8 and CentOS 8

Problem

Agent installation failed on RHEL8.

OR

Agent configuration failed on CentOS 8.

Symptom

1. Agent installation failed on RHEL 8 due to unavailability of libnsl.so.1.

   Example: Agent installation failed on RHEL 8.

   ```
   KAX - KAK=Azure Compute AgentKAL=Amazon ELBIBM Tivoli Monitoring Shared Libraries [version
   06400015]:
   Property                            Result    Found
   ```

```
Expected
========                            ======    =====
========
os.lib.libnsl_64                    FAIL      Unavailable
regex{libnsl.so.1}
```

2. Agent configuration failed on CentOS 8 due to error while loading shared libraries: libnsl.so.1.

   Example: Agent configuration failed on CentOS 8.

   ```
   /opt/ibm/tmaitm6/lx8266/bin/xc_silent: error while loading shared libraries: libnsl.so.1:
   cannot open shared object file: No such file or directory
   ```

Cause

The shared library libnsl is not installed in the system.

Solution

1. Download libnsl RPM package for RHEL 8 or CentOS 8 accordingly.
2. Install the RPM package:

   ```
   rpm -ivh <libnsl rpm>
   ```

Collecting monitoring agent logs for IBM Support

Use the problem determination collection tool, *pdcollect*, to gather required logs and other problem determination information that is requested by IBM Support for monitoring agents. The PD collector tool is installed with each monitoring agent.

Before you begin

Root or administrator permission is required for the PD collector tool to collect system information from the monitoring agents. You can review the agent logs individually in the following folders:

- Windows [64-bit] *install_dir*\TMAITM6_x64\logs
- Windows [32-bit] *install_dir*\TMAITM6\logs
- Linux AIX *install_dir*/logs

Restriction: It is only possible to run one instance of the pdcollect script.

About this task

The default location of *install_dir* is:

- Windows C:\IBM\APM
- Linux /opt/ibm/apm/agent
- AIX /opt/ibm/apm/agent

To run the PD collector tool, complete the following steps:

Procedure

1. On the command line, change to the agent directory:
   - Linux AIX *install_dir*/bin
   - Windows *install_dir*\BIN
2. Run the following command:
   - Linux AIX **./pdcollect**

- Windows **`pdcollect`**

  A file with a time stamp in the file name is generated in the `tmp` directory, such as `/tmp/pdcollect-nc049021.tar.Z`.

3. Send the output files to your IBM Support representative.

What to do next

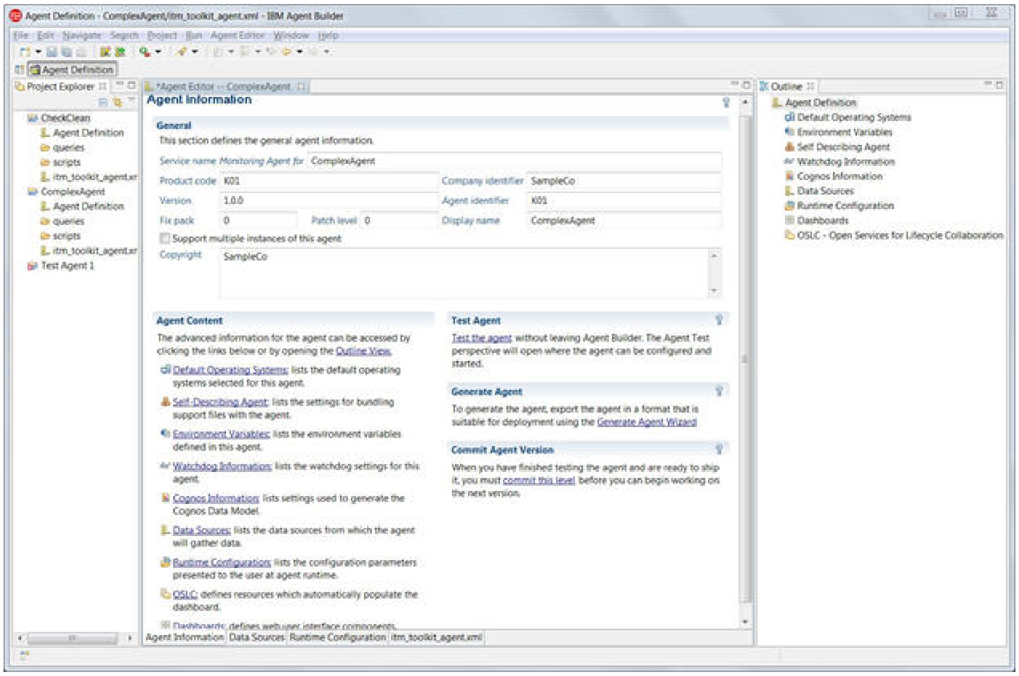
If you have installed the Ruby agent and configured it for the Diagnostics dashboards, run the kkm collector tool, *kkmCollector*, on Linux systems to gather configuration files, output files such as JSO files, and log files.

1. Change to the `*install_dir*/lx8266/km/bin` directory.
2. Run the command **`./kkmCollector`**

   A file with a time stamp in the file name is generated in the `tmp` directory, such as `/tmp/kkm_dchome.tar.gz`
3. Send the output files to your IBM Support representative.

Chapter 13. Agent Builder

The IBM Agent Builder tool provides a graphical user interface to help you create, modify, debug, and package agents for monitoring data sources in IBM Cloud Application Performance Management.



Agent Information

General

This section defines the general agent information.

Service name *Monitoring Agent for* ComplexAgent

Product code K01 | Company identifier SampleCo

Version 1.0.0 | Agent identifier K01

Fix pack 0 | Patch level 0 | Display name ComplexAgent

□ Support multiple instances of this agent

Copyright SampleCo

Agent Content

The advanced information for the agent can be accessed by clicking the links below or by opening the Outline View.

Default Operating Systems: lists the default operating systems selected for this agent.

Self-Describing Agent: lists the settings for bundling support files with the agent.

Environment Variables: lists the environment variables defined in this agent.

Watchdog Information: lists the watchdog settings for this agent.

Cognos Information: lists settings used to generate the Cognos Data Model.

Data Sources: lists the data sources from which the agent will gather data.

Runtime Configuration: lists the configuration parameters presented to the user at agent runtime.

OSLC: defines resources which automatically populate the dashboard.

Dashboards: defines web user interface components.

Test Agent

Test the agent without leaving Agent Builder. The Agent Test perspective will open where the agent can be configured and started.

Generate Agent

To generate the agent, export the agent in a format that is suitable for deployment using the Generate Agent Wizard

Commit Agent Version

When you have finished testing the agent and are ready to ship it, you must commit this level before you can begin working on the next version.

Overview of Agent Builder

You can use IBM Agent Builder to create and modify custom agents that extend the monitoring capabilities of an IBM Tivoli Monitoring, IBM Cloud Application Performance Management, IBM Cloud Pak for Multicloud Management environment. A custom agent uses either of these environments to monitor any type of in-house or customized software.

Agent Builder is based on Eclipse, an open source integrated development environment.

Agent Builder includes the following features for the Tivoli Monitoring and Cloud APM environments:

Define and modify agents
You can create and modify agents. The agents collect and analyze data about the state and performance of different resources, such as disks, memory, processor, or applications, and provide this data to the monitoring environment.

Test and prepare agents for deployment
You can test an agent within Agent Builder, collecting data on the host where Agent Builder runs (in some cases you can collect information from a different host too). You can package the agent for easy distribution and deployment.

The following additional features are available for Tivoli Monitoring:

Custom workspaces, situations and Take Action commands
You can use Agent Builder to package additional workspaces, situations and Take Action commands as application support extensions with a new or existing agent running in the Tivoli Monitoring environment

Report data models
You can use Agent Builder to generate a Cognos data model which you can use to build Tivoli Common Reporting reports. These reports can be packaged as part of your agent image.

Common Agent Builder procedures

The following table lists the main procedures that you can complete with Agent Builder.

You can use Agent Builder to create agents for the IBM Tivoli Monitoring and IBM Cloud Application Performance Management environments. You can also use it to create application support extensions for the Tivoli Monitoring environment. Application support extensions are created by creating workspaces and situations to enhance one or more existing agents.

Before you use Agent Builder, you must install it. For instructions, see "Installing and starting Agent Builder" on page 1144.

To create, test, and use an agent, complete the procedures in the following table in the order that they are listed.

Table 260. Quick-reference information for creating agents

Goal	Refer to
Create an agent by using the **Agent** wizard.	• "Creating an agent" on page 1149
Create data sources and attributes for your agent. **Important:** For a Cloud APM environment, a summary dashboard can display up to approximately five attributes; one of the attributes must denote overall agent or subnode status.	• "Editing data source and attribute properties" on page 1168
For the Tivoli Monitoring environment, create workspaces and situations for your agent. • Running at least Tivoli Monitoring Version 6.1 Fix Pack 1 • Setting the Tivoli Universal Agent solution version back to "00" • Setting the value for "AppTag"	• "Creating workspaces, Take Action commands, and situations" on page 1337 • "Importing application support files" on page 1371
For the Cloud APM environment, create resource definitions and dashboards for your agent.	• "Preparing the agent for Cloud APM" on page 1342
For the Tivoli Monitoring environment, create Cognos data models for reports for your agent.	• "Cognos data model generation" on page 1440
Test and debug your created agent, ensuring the availability of monitoring information.	• "Testing your agent in Agent Builder" on page 1346 • "Command-line options" on page 1380 • "Using the Agent Editor to modify the agent" on page 1152.
Generate an installation package and install the agent on the monitored host.	• "Installing an agent" on page 1355
Remove an agent that you created with the Agent Builder.	• "Uninstalling an agent" on page 1369

You can also use Agent Builder for packaging custom workspaces, situations, and Take Action commands as application support extensions for existing agents. These functions are available only for the Tivoli Monitoring environment:

Table 261. Quick-reference information for other functions

Goal	Refer to
Create custom workspaces, situations, and Take Action commands.	• "Creating workspaces, Take Action commands, and situations" on page 1337
Package your application support extension.	• "Creating application support extensions for existing agents" on page 1437
Build custom bundles.	• "Creating Non-agent file bundles" on page 1460

Data sources and data sets

An agent can monitor information from one or several data sources. It presents the information to the monitoring infrastructure as attributes, which are organized into data sets.

When you create an agent, you must define a *data source* for it. You can add more data sources. The data source defines how the agent gathers the monitoring information.

You can use Agent Builder to create agents that use data sources monitoring information from the following *data providers*:

- Process and service availability
- Network system availability (using ICMP ping)
- Command return codes
- Script output
- The Windows Event Log
- Windows Management Instrumentation (WMI)
- Windows Performance Monitor (Perfmon)
- Simple Network Management Protocol (SNMP)
- SNMP Events
- Hypertext Transfer Protocol (HTTP) availability and response time
- SOAP or other HTTP data source
- Java Database Connectivity (JDBC)
- Java application programming interface (API)
- Java Management Extensions (JMX)
- Common Information Model (CIM)
- Log files
- AIX binary logs
- Socket

You can also use other development tools to create custom monitoring applications that pass information to the agent through log, script output, and Java API data sources.

When you add a data source, Agent Builder adds the corresponding *data set* to the agent. The data set organizes the information that is presented to the monitoring environment. In IBM Tivoli Monitoring, a data set is known as an *attribute group*.

A data set can consist of several *attributes*, which are values that the data source provides. Each time the monitoring environment queries the agent, it fetches values from data sources and returns then as attributes in data sets.

Some data sources can return several *rows* of attribute values in the same query, for example, if the data source monitors several services at once.

Most data sources present information as one data set. SNMP and JMX data sources can, depending on the configuration, provide diverse sets of information. When you add an SNMP or JMX data source, Agent Builder creates multiple data sets to accommodate this information.

You can edit the data sets to filter the data and to create additional *derived* attributes, that is, attributes that are calculated from existing attributes using a formula. You can also join data sets, creating a new data set with information from two or more data sets. In this way, users can view combined information from different data sources.

In IBM Tivoli Monitoring, you can view all attribute content. You can also create workspaces that present information from all agent data sets in a customized view. You can use IBM Tivoli Monitoring to create situations that are triggered when any attribute reaches a certain value. A situation can issue an alert and to call a system command.

In IBM Cloud Application Performance Management, you must define a *summary* dashboard for the agent, selecting up to five attributes that are visible in the dashboard. You can also define a *detail* dashboard that displays information from any data sets as tables. You can create thresholds that are triggered when any attribute reaches a certain value; you are not required to add this attribute to the dashboard. A threshold can issue alerts.

In IBM Cloud Pak for Multicloud Management you define resources for the agent, select attributes to be displayed, and identify important metrics to chart.

Monitoring multiple servers or instances of a server

An agent can monitor multiple servers, including multiple instances of the same server. There are two ways of creating such agents: multiple instances of an agent and subnodes within an agent.

Multiple instances are a standard way to monitor application servers that can have a number of similar instances on the same host. Many standard agents in IBM Tivoli Monitoring and IBM Cloud Application Performance Management support multiple instances.

With *multiple instances*, you install an agent on a monitored hosts and then configure one or several instances, setting a name for every instance. Configure an instance of the agent for each instance of the server that you want to monitor. Each instance is a separate identical copy of the agent, and it can be started and stopped separately.

You can also define one or several types of *subnode* within an agent. Each type must correspond to a different type of resource that an agent can monitor. A subnode type contains data sources and data sets; you can also define data sources and data sets at agent level, outside any subnode. When you install the agent on a host, you can configure the required number of subnodes of each type; for every subnode type, you can set the number of subnodes independently. For IBM Cloud Application Performance Management, you can create a dashboard for the agent and a separate dashboard for each subnode.

Subnodes require different configuration steps on the monitored host. Also, to reconfigure, add or remove a subnode. you must stop and restart the entire agent; an instance can be reconfigured, added, or removed without affecting other instances. However, subnodes have a number of advantages:

- With subnodes, you can monitor a large amount of server instances while consuming less resources. As a guideline, the number of agent instances of a specific type supported on a single system is 10. But an agent can monitor up to 100 local or remote servers using subnodes.
- One agent can include subnode types for a few different kinds of servers. On the monitored system, you can configure any number of subnodes of each type. You can use this feature to conserve resources further.
- An agent with subnodes can supply system-wide data on the agent level.

You can define both multiple instances and subnodes for the same agent. In this case, each instance can include a number of subnodes. You can stop and restart each instance independently of other instances; all subnodes in an instance are stopped and restarted together.

Testing, installing, and configuring an agent

You can create an installation package for an agent and then install it on any number of monitored hosts. For some data sources, you need to set configuration values for collecting data.

After defining data sources and attributes for an agent, you can test it by running it within Agent Builder. You can test a single data set (attribute group) or the full agent.

To test the agent more extensively and to use it, you can create an installation image. This image provides scripts for installing and configuring the agent on any monitored host.

Tip: Before installing the agent, ensure that the operating system agent for your monitoring environment (IBM Tivoli Monitoring, IBM Cloud Application Performance Management, or IBM Cloud Pak for Multicloud Management) is installed on the host.

After installing the agent, you might need to configure it. If the agent supports multiple instances, you must configure the agent to create at least one instance.

Some data sources require additional configuration values; for example, for the SNMP data source, you must configure the IP address of the host that you monitor using the SNMP protocol. Use the configuration script, which is deployed by the installation package, to set these values.

Alternatively, you can set these values in Agent Builder before creating the installation image. In this case, you do not have to set them again on the monitored hosts.

Tip: The help files for your custom agent might not display in Help Contents after the Cloud APM server is upgraded. To display the help files, complete these steps:

1. Download the latest version of IBM Agent Builderfrom your Cloud APM subscription in IBM Marketplace.
2. Re-create your custom agent. Make sure to assign a higher version number, fix pack, or patch level in the Agent Information page.
3. Install your custom agent on the monitored host.
4. From the Cloud APM console, click **Help** > **Help Contents** from the navigation bar. Your custom agent help is displayed.

Operating system requirements

Agents that are created by Agent Builder are supported on various operating systems, depending on the monitoring environment and on the settings you select when creating the agent.

In a Tivoli Monitoring environment, agents that are created by Agent Builder can support the following operating systems:

- AIX
- HP-UX
- Linux
- Solaris
- Windows

The agents support the same operating system versions as the OS agents. For details, access the Software Product Compatibility Reports website. Search for the `Tivioli Monitoring` product name and select the `OS Agents & TEMA (Tivoli Enterprise Monitoring Agent)` component check box.

In an IBM Cloud Application Performance Management environment, agents that are created by Agent Builder can support the following operating systems:

- AIX
- Linux
- Windows

The agents support the same versions as the OS agents. For details, use the links in the Component reports section of System requirements (APM Developer Center).

To run your monitoring agent in an Tivoli Monitoring environment, install the appropriate operating system agent on every monitored system where your agent runs.

To run your monitoring agent in an IBM Cloud Application Performance Management environment, install any of the agents shipped with IBM Cloud Application Performance Management on every monitored system where your agent runs.

Note: Agent Builder browsers operate on the data sources and information accessible from the system on which the Agent Builder is run. Ensure that you run Agent Builder on either of the following types of systems:

- A system that runs on the same level as the operating system and monitored applications for which you are developing the agent
- A system that connects to another system that runs on the same level as the operating system and monitored applications for which you are developing the agent

Features specific to IBM Tivoli Monitoring

Agent Builder provides several features that apply only to IBM Tivoli Monitoring.

You can use navigator groups to organize the data that the agent displays in the IBM Tivoli Monitoring navigator views and workspaces. A navigator group combines the data from several attribute groups (data sets) into a single view, while hiding the original separate data sets from the user.

You can use Tivoli Enterprise Portal to create workspaces, situations, and Take Action commands for your agent. You can then use Agent Builder to save the workspaces, situations, and Take Action commands as application support files and bundle them with the agent. Moreover, Agent Builder can also import workspaces, situations, and Take Action commands for other agents and create custom application support files for them.

Agent Builder can generate a Cognos data model for the agent. Use the data model to import agent information into the Cognos Framework Manager, a part of IBM Tivoli Common Reporting, for report creation.

Installing and starting Agent Builder

Before you install IBM Agent Builder, ensure that your system meets the prerequisites. Then use the installation wizard or the silent installation procedure to install Agent Builder.

Tip: For information about installing or modifying an *agent*, see "Installing an agent" on page 1355.

Prerequisites for installing and running Agent Builder

To install and run Agent Builder, your system must meet certain requirements.

To install the Agent Builder, ensure that you have:

- A system with a minimum of 1 GB of free disk space. Agents that you develop will require additional disk space.
- A supported operating system. Agent Builder can run on the following operating systems:
  - Windows Windows
  - Linux Linux (x86 64-bit only)
- Linux If you are using the Linux operating system, you must install the `libstdc++.so.`**5** library. You can install the following packages that provide this library:
  - On Red Hat Enterprise Linux, `compat-libstdc++-33`
  - On SUSE Enterprise Linux, `libstdc++-33`

Windows On a Windows system, you must be able to run Agent Builder as a user with Administrator permissions. These permissions ensure that Agent Builder has an environment consistent with the agents that are developed with it.

Linux On a Linux system, you can run Agent Builder as root or as an ordinary user. However, if you run it as an ordinary user, testing of agents will be limited and in some cases might not be available.

Detailed system requirements for Agent Builder

Use the Software Product Compatibility Reports to view the detailed system requirements for Agent Builder.

Access the Software Product Compatibility Reports website. Search for the `IBM Agent Builder` product name.

Installing Agent Builder

You can use the installation wizard or the silent installation procedure to install Agent Builder.

Tip: Before you install Agent Builder, uninstall any previous versions. For more information about uninstalling, see (“Uninstalling Agent Builder” on page 1148). None of your existing agent information is lost when you uninstall.

Using the installation wizard to install Agent Builder

You can use the installation wizard to install IBM Agent Builder.

Before you begin

Ensure that your system meets the prerequisites. For information about prerequisites, see “Prerequisites for installing and running Agent Builder” on page 1144

Procedure

1. If you are not signed in to IBM Marketplace, sign in with your IBMid and password and go to **Products and services**.

   The **Products and services** page is available to active subscribers. If you have any issues, go to the Cloud Application Performance Management Forum or to Marketplace support.
2. Download the Agent Builder installation archive file:
   a) In the Cloud APM subscription box, click **Manage** > **Downloads**.
   b) Select **Multi-Platform** as the operating system.
   c) Select the IBM Agent Builder package.
   d) Click **Download** and save `IBM_Agent_Builder_Install.tar` to your system.
3. Extract the installation archive file.
4. Use the following command in the extracted image directory to start the installation:
   - Windows `setup.bat`
   - Linux AIX `./setup.sh`

   Important: Run the installation program with the same user ID that you intend to run the Agent Builder with.
5. When the **IBM Agent Builder** window opens, select your language, and click **OK**.
6. On the **Introduction** page, click **Next**.
7. On the **Software License Agreement** page, click **I accept the terms in the license agreement**, and click **Next**.
8. On the **Choose Install Folder** page, click one of the following options:
   - **Next** to install Agent Builder to the directory specified in the **Where Would You Like to Install?** field.

- **Restore Default Folder** to install the Agent Builder in a default directory.
- **Choose** to select a different directory.

Note: The directory name that you choose must not contain the following characters:

```
!
#
%
;
```

If it includes any of these characters, Agent Builder might not start.

9. On the **Pre-Installation Summary** page, click **Install**.
10. On the **Installing IBM Agent Builder** page, wait for the **Install Complete** page to open, then click **Done**.

Results

Windows After the Agent Builder is installed, an option is added to the Start menu and an Agent Builder icon is added to your desktop. The installation log files are in *install_dir*\IBM_Agent_Builder_InstallLog.xml.

Linux AIX After the Agent Builder is installed, the Agent Builder executable file is named *Install_Location*/agentbuilder. The installation log files are in *install_dir*/IBM_Agent_Builder_InstallLog.xml.

Silent installation

You can install Agent Builder by using a silent installation method. This method does not require a graphical environment and can be easily replicated on several hosts.

About this task

The silent installation options file, installer.properties, is included in the installation image at the root of the installation directory. You must modify this file to meet your needs, and then run the silent installer. You can copy this file to other hosts and quickly install Agent Builder on all of them.

Procedure

1. If you are not signed in to IBM Marketplace, sign in with your IBMid and password and go to **Products and services**.

   The **Products and services** page is available to active subscribers. If you have any issues, go to the Cloud Application Performance Management Forum or to Marketplace support.
2. Download the Agent Builder installation archive file:
   a) In the Cloud APM subscription box, click **Manage** > **Downloads**.
   b) Select **Multi-Platform** as the operating system.
   c) Select the IBM Agent Builder package.
   d) Click **Download** and save IBM_Agent_Builder_Install.tar to your system.
3. Extract the installation archive file.
4. Create a copy of the installer.properties file, which is located in the installation image directory.
5. Edit the new file to suit your needs. An example of the contents of this file is:

```
# ---------------------------------------------------------------------
# IBM Agent Builder
#
# (C) Copyright IBM Corporation 2009.  All rights reserved.
#
# Sample response file for silent install
#
# To use this file, use the following command:
```

```
#
# Windows:
#    setup.bat -i silent -f <path>\installer.properties
#
# Linux or AIX:
#    setup.sh -i silent -f <path>/installer.properties
#
# Where
#    <path> is a fully-quailfied path to the installer.properties
#    file (including the drive letter or UNC path name on Windows).
#    <path> cannot contain spaces.
# -------------------------------------------------------------------


# -------------------------------------------------------------------
# This property indicates that the license has been accepted
# -------------------------------------------------------------------
# LICENSE_ACCEPTED=FALSE

# -------------------------------------------------------------------
# This property specifies the install directory
#
# On Windows, the default is:
#     C:\\Program Files (x86)\\IBM\\AgentBuilder
#
# On Linux, the default is:
#     /opt/ibm/AgentBuilder
# -------------------------------------------------------------------
#USER_INSTALL_DIR=C:\\Program Files (x86)\\IBM\\AgentBuilder
#USER_INSTALL_DIR=/opt/ibm/AgentBuilder
```

6. Start the silent installation by running the following command in the extracted installation image directory:

   Windows `setup.bat -i silent -f` *`path`*`/installer.properties`

   Linux AIX `./setup.sh -i silent -f` *`path`*`/installer.properties`

   Where *path* is the fully-qualified path to the `installer.properties` file (including the drive letter or UNC path name on Windows). The path cannot contain spaces.

Starting Agent Builder

After installing Agent Builder, you can start it.

Procedure

- Start the Agent Builder by using one of the following methods
  - Windows On Windows systems:
    - From a command-line type: *`Install_Location`*`\agentbuilder.exe`.
    - Select **Start** > **All Programs** > **IBM** > **Agent Builder**.
    - Click the **Agent Builder desktop icon**.
  - Linux On Linux systems, start the following executable file: *`INSTALL_DIR`*`/agentbuilder`

  Note: When you run the Agent Builder, it prompts you for the location of your workspace directory. The files that create your agents are saved in that directory. You can designate any directory as your workspace.

Setting the default browser in Agent Builder

Linux On Linux systems, you might need to set the Agent Builder default browser so that help panels are displayed.

Procedure

1. Select **Window** > **Preferences** to open the **Preferences** window.
2. Select and expand the **General** node.

3. Select **Web Browser**.
4. Select **Use external web browser**.
5. Select the browser that you want to use.
6. Optional: To add a web browser, complete the following steps
   a) Click **New**.
   b) In the **Name** field, enter a descriptive name for the browser.
   c) In the **Location** field, enter the full path to the browser executable file .
   d) Click **OK**.
7. Click **OK**.

Setting the default Time Stamping Authority in Agent Builder

You can set the Time Stamping Authority for JAR files in the Agent Builder **Preferences** window. If the default Time Stamping Authority signing certificate expires, by setting a new authority, you can continue to verify JAR files.

Procedure

1. Select **Window** > **Preferences** to open the **Preferences** window.
2. Select and expand the **IBM Agent Builder** node.
3. Select **Jar Signing**.
4. Select **Add time stamp to signed JAR files**.
5. Enter the URL of the Time Stamping Authority.
6. Click **OK**.

Uninstalling Agent Builder

Depending on your operating system, you can use different procedures to uninstall Agent Builder.

Procedure

- Linux
  On Linux systems, run the following command:
  a) *INSTALL_DIR*`/uninstall/uninstaller`
     where *INSTALL_DIR* is the name of the directory where Agent Builder is installed.
- Windows
  On Windows 7, Windows Server 2008 R2, and later versions of Windows, complete the following steps:
  a) Open Windows Programs and Features by selecting **Start** > **Control Panel** > **Programs** > **Programs and Features**.
  b) Select **IBM Agent Builder** from the list of installed programs.
  c) Click **Uninstall/Change**.
  d) Click **Uninstall** on the **Uninstall IBM Agent Builder** page.
  e) Click **Done** on the **Uninstall Complete** page.

  Tip: On Windows 7 and Windows Server 2008 R2, you can also go to the **Windows Programs and Features** window by selecting **Start** > **Computer** > **Uninstall or change a program**. Then, continue from step "2" on page 1148.
- Windows
  On other Windows systems, complete the following steps:
  a) From the Windows Control Panel, select **Add/Remove Programs**.
  b) Click **IBM Agent Builder**.

c) Click **Change/Remove**.

- On all operating systems, you can also use the silent uninstallation method. Start the silent uninstallation by running the following command:
  - Windows On Windows systems, `INSTALL_DIR/uninstall/uninstaller.exe -i silent`
  - Linux On Linux systems, `INSTALL_DIR/uninstall/uninstaller -i silent`

Silent uninstallation

You can use the silent uninstallation method to uninstall.

Procedure

- Start the silent uninstallation by running the following command:

```
INSTALL_DIR/uninstall/uninstaller[.exe] -i silent
```

Creating an agent

To start creating an agent in Agent Builder, use the new agent wizard. With this wizard you can set the basic agent configuration and create one data source. You can then work on the agent in Agent Builder to add more data sources and other options, including subnodes and navigator groups.

Naming and configuring the agent

Use the **Agent** wizard to name your agent, set its version, supported operating systems, and other configuration settings.

Procedure

1. Use one of the following ways to start the new agent wizard:

   a) Click the **Create New Agent** icon on the toolbar.

   b) From the Main menu, select **File** > **New** > **Agent**.

   c) From the Main menu, select **File** > **New** > **Other**. In the **Select a Wizard** page, double-click the **Agent Builder** folder, then double-click **Agent**.

   The **Agent** wizard opens.
2. Click **Next**.
3. In the **New Agent Project** page, set the name of the project in the **Project name** field. Agent Builder uses this name for the folder that contains the agent files. You can optionally change the following settings:
   - If you want to store the agent files in a different location, clear **Use default location** and click **Browse** to select the new directory in the **Location** field.
   - You can change how the Eclipse Navigator View displays resources by adding them to various working sets. For more information, see the Eclipse help. To add the agent to Eclipse working sets, select **Add project to working sets** and click the **Select** button to add the sets to the **Working sets** field.
4. Click **Next**.
5. In the **General Information** page, configure the following settings:
   - Type the copyright statement that you want to use for your new agents in the **Copyright** field. This statement must meet your legal requirements for copyrights. This copyright statement is inserted into all files that are generated for the agent; you can edit it later.
   - Select the operating systems for which you want your agent to be built.

Important: If you want to run a full test of the agent inside Agent Builder (for instructions, see "Full agent testing" on page 1350), ensure that:

– If you are running Agent Builder on Windows, the 32-bit version of the operating system is installed.

– If you are running Agent Builder on Linux, the 64-bit version of the operating system is installed.

Important: In some rare cases, you might need to install your agent on a 64-bit system where only a 32-bit operating system agent is installed. In this case, ensure that the 64-bit version of the operating system is not selected and the 32-bit version is selected.

Important: 64-bit Windows Server 2003 R2 and earlier Windows systems are not supported by the agents created using Agent Builder.

6. Click **Next**.
7. In the **Agent Information** page, configure the following settings:

- Set the service name for the agent in the **Service name** field. The name is displayed in the **Manage Tivoli Monitoring Services** window in an IBM Tivoli Monitoring environment and in the **Manage Monitoring Services** utility and Threshold editor in an IBM Cloud Application Performance Management. On Windows systems, it is also the name of the Windows service that runs the agent. The full service name always starts with `Monitoring Agent for`. You enter the remaining part of the name, which normally describes the service that this agent monitors. The name can contain letters, numbers, spaces, and underscores.
- Set a three-character product code for the agent in the **Product code** field. A product code is required for both IBM Tivoli Monitoring and IBM Cloud Application Performance Management. A range of product codes is reserved for use with the Agent Builder. The permitted values are `K00-K99`, `K{0-2}{A-Z}`, and `K{4-9}{A-Z}`.

  Important: These values are for internal use only and are not intended for agents that are to be shared or sold outside your organization. If you are creating an agent to be shared with others, you must send a note to toolkit@us.ibm.com to reserve a product code. The request for a product code must include a description of the agent to be built. A product code is then assigned, registered, and returned to you. When you receive the three-letter product code, you are told how to enable the Agent Builder to use the assigned product code.
- Set a string that uniquely identifies the organization that develops the agent in the **Company identifier** field (`IBM` is reserved). You can take it from the URL of your company; for example, if the company website is `mycompany.com`, use the text `mycompany`.
- Set a string that uniquely identifies the agent in the **Agent identifier** field. By default, Agent Builder sets the Agent identifier to be the same as the Product code.

  Important: The combined length of the **Agent identifier** field and the **Company identifier** field cannot exceed 11 characters.
- Set the agent version in the **Version** field. The agent version contains three digits in the format *`V.R.R`*, where:

  V = Version
  R = Release
  R = Release

  For displaying in the monitoring environment, the *`V.R.R`* value is converted into the following format: `0V.RR.00.00`

  Tip: In the agent editor, a **patch level** field is available. The **patch level** field can be used when you release a fix for an agent, without updating the version.
- If you want your agent to support multiple instances, select the **Support multiple instances of this agent** check box. You can use multiple instances of an agent to monitor several instances of an application on the same host, or to use an agent installed on one host to monitor several software servers on different hosts. When you install an agent that support multiple instances, you can create and configure as many instances as necessary.

What to do next

Click **Next** to define an initial data source for your agent. For more information, see "Defining initial data sources" on page 1151

Defining initial data sources

When creating an agent, define the initial data that the agent is to monitor. You can add more data sources later in the agent editor.

About this task

Define the data sources that your new agent is to monitor by using the **Agent Initial Data Source** page. For detailed instructions about creating data sources from various data providers, see "Defining and testing data sources" on page 1194.

Procedure

1. On the **Agent Initial Data Source** page, select one of the **Monitoring Data Categories** and one of the **Data Sources**.
2. Click **Next** The wizard guides you through the process of defining and configuring any of the data collection types that you specify.

   Tip: You can use this wizard to define a data source or to add a subnode or navigator group for organizing the agent. For more information about subnodes, see "Using subnodes" on page 1314. For more information about navigator groups, which are used only for IBM Tivoli Monitoring, see "Creating a navigator group" on page 1312.
3. If you defined a new data source that might return more than one data row, you are prompted to select key attributes. For more information, see ("Selecting key attributes" on page 1151).
4. After you define the first data source, the **Data Source Definition** window displays. To add another data source, select the agent, or a subnode or navigator group if one is present, and click the **Add to Selected** button.
5. To finish defining data sources, click **Finish**. Aent Builder creates the new agent and opens it in the agent editor.

Selecting key attributes

When an attribute group returns more than one data row, you must select key attributes.

About this task

When an attribute group can return more than one data row, each row represents an entity that is being monitored. Each time monitored data is sampled, the monitoring environment matches a row to the entity that is being monitored and to previous samples for that entity. This matching is done with key attributes. One or more attributes in the attribute group can be identified as key attributes. These key attributes, when taken together, distinguish one monitored entity from another. The key attributes do not change from one sample to the next for the same monitored entity.

Rate and delta attributes are calculated by comparing the current sample to the previous sample. Identical key attributes ensure that the agent is comparing values for the same monitored entity. Similarly, the summarization and pruning agent summarizes samples that have identical key attributes. In addition, any attribute that is set as a key attribute can also be used as a "Display item" in a situation.

You specify the details about your new data source in the **Agent Initial Data Source** page. If the selected data source might return multiple data rows, Agent Builder can sometimes detect the key attributes. Otherwise, it prompts you to select key attributes.

Procedure

- On the **Select key attributes** page, take one of the following steps:
  - Click one or more attributes from the list that are the key attributes for this entity. To select more than one attribute, hold down the Ctrl key.

- If this attribute group returns only one row, select **Produces a single data row**. If this option is selected, no key attributes are necessary because only one monitored entity is ever reported in this attribute group.

Using the Agent Editor to modify the agent

Use the Agent Editor to change, save, and commit a version of your agent.

You can create a new agent in Agent Builder; for more information, see "Creating an agent" on page 1149. After creating an agent, you can modify it using the Agent Editor.

To open an agent that you created in Agent Builder in the Agent Editor, in the **Project Explorer** pane, find the name of the agent and expand it. Under the name of the agent, double-click **Agent Definition**. Alternatively, double-click the `itm_toolkit_agent.xml` filename.

The Agent Editor is a multi-page Eclipse editor that you can use to modify the properties of an existing agent. Each page in the editor corresponds to a specific function of the agent.

The list of available pages is shown in the Outline view under the **Agent Definition** node. You can easily switch to another page by clicking a node in the Outline view. If the Outline view is missing, or hidden behind another view, you can reset the Agent Definition perspective. Reset the perspective by selecting **Window** > **Reset Perspective**. Alternatively, right-click the **Agent Definition** tab and select **Reset** from the menu.

Note: For detailed information and procedures for creating an agent, see "Creating an agent" on page 1149.

The following pages are included in the Agent Editor:

- "Agent Information page" on page 1152
- Data Source Definition page
- Runtime Configuration Information page
- Agent XML Editor page (itm_toolkit_agent.xml)

Note: When you view an Editor page, you can also switch to another page by clicking the tab for the page. Some pages show tabs only when they are selected in the Outline view. You can force a page to have a tab even when it is not selected. To force a page to have a tab, click the pin icon so that the pin in the icon points toward the page.

Agent Information page

The **Agent Information** page is the main page of the Agent Editor.

The **Agent Information** page contains the following information:

- General agent information, including the agent service name and the product code. You can click **Advanced** to set different names for different use, but this setting is normally not needed.
- Agent Content information
  - **Default Operating Systems** link
  - **Self-Describing Agent** link
  - **Environment Variables** link
  - **Watchdog Information** link
  - **Cognos Information** link
  - **Data sources** link
  - **Runtime Configuration** link
  - **Resources** link
  - **Dashboards** link
- **Test Agent** link

- **Generate Agent Wizard** link
- **Commit Agent Version** link

Configuring the time for transient error messages

Agent Editor wizards sometimes display transient error messages. A message is displayed for a short time (by default, 3 seconds) in the header of the wizard. You can configure the duration for which these messages are displayed. To change this setting:

1. Select **Window** > **Preferences** from the Agent Builder menu bar. The **Preferences** window opens.
2. Select **Agent Builder**.
3. Set the **Time (seconds) that transient error message are displayed** setting.
4. Click **OK**.

Default operating systems

Use the **Default Operating Systems** page to change the operating systems for which your agent is built.

Procedure

- To open the **Default Operating Systems** page, click **Default Operating Systems** in the **Agent Content** section of the **Agent Information** page or the **Default Operating Systems** node in the Outline View.
- In the **Default Operating Systems** page, select the operating systems that your agent must support.

  When you generate an installation package for the agent, Agent Builder adds files for the selected operating systems to the package. Data sources that you add to your agent that are not specific to the Windows operating system are available on any of the selected operating systems. The operating systems on which any specific data source is available can be changed from this default selection. To change the Operating Systems available for a specific data source, use the **Operating Systems** pane of the **Data Source Definition** page. If default operating systems are not selected, operating systems must be selected for each specific data source on the **Data Source Definition** page.

  Important: If you want to run a full test of the agent inside Agent Builder (for instructions, see "Full agent testing" on page 1350), ensure that:

  - If you are running Agent Builder on Windows, the 32-bit version of the operating system is installed.
  - If you are running Agent Builder on Linux, the 64-bit version of the operating system is installed.

  Important: In some rare cases, you might need to install your agent on a 64-bit system where only a 32-bit operating system agent is installed. In this case, ensure that the 64-but version of the operating system is not selected and the 32-bit version is selected.

Self-Describing Agent

For the IBM Tivoli Monitoring environment, use the **Self-Describing Agent** page to specify whether the agent's support files are bundled with the agent. For the IBM Cloud Application Performance Management environment, you must leave Self-Describing Agent enabled.

Procedure

- To open the **Self-Describing Agent** page, click **Self-Describing Agent** in the **Agent Content** section of the **Agent Information** page or the **Self-Describing Agent** node in the Outline View.

  Self-description is enabled by default for all new agents that are created with Agent Builder 6.2.3 or later. If the agent is for the IBM Cloud Application Performance Management environment, self-description must be enabled.

  When self-description is enabled for an agent, application support packages are included in the agent image. The inclusion enables the agent to seed the support files for the Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, the Tivoli Enterprise Portal Browser. For more information about self-describing agents, see the *IBM Tivoli Monitoring Installation and Setup Guide* and the *IBM Tivoli*

Monitoring Administrator's Guide. In an IBM Cloud Application Performance Management environment, self-description enables the agent to seed support files onto the Cloud APM server; the seeding is a required step in the environment.

Note: In a IBM Tivoli Monitoring environment, you must have Tivoli Monitoring version 6.2.3 or later installed for the self-describing agent feature to work, and self-description must be enabled in Tivoli Monitoring. By default self-description is turned off in Tivoli Monitoring.

Note: Selecting the **Enable self-description for this agent** check box does not prevent your agent from working on previous versions of Tivoli Monitoring.

Environment variables

Use the **Environment Variables** page to view and modify environment variables that are available to your agent while it is running.

Before you begin

For more information about the **Agent Editor** and **Agent Information** page, see "Using the Agent Editor to modify the agent" on page 1152.

About this task

The environment variables can be defined by you, for access inside a script, or predefined variables that cause the agent to behave in a certain way. See "List of environment variables" on page 1154 for a list of predefined variables.

Procedure

1. To open the **Environment Variables** page, click **Environment Variables** in the **Agent Content** section of the **Agent Information** page. Alternatively, click **Environment Variables** node in the **Outline** view.
2. In the **Environment Variables** page, click **Add** to add a new variable. Alternatively, to edit an existing variable, select it and click **Edit**.
3. In the **Environment Variable Information** window, set the following values:
   - In the **Name** field, type a variable name or select a predefined name from the list.
   - In the **Value** field, type a value for the variable if you want to set a variable for the agent. If you do not enter a value, the agent propagates a value for the existing variable.
   - In the **Description** field, type a description of the variable, or keep the existing description of a predefined variable.

   a) Click **OK**.

      The new variable is listed in the table on the **Agent Information** page.

List of environment variables

Use environment variables to control the behavior of the agent at run time.

Environment variables can be built into the agent by using the **Environment Variables** page. On Windows systems, environment variables are defined in the agent K*XX*ENV file. On UNIX and Linux systems, these variables can be defined in the agent `$CANDLEHOME/config/`*`XX`*`.ini` file. *XX* is the two-letter product code. The agent must be restarted for the new settings to take effect.

Note: Environment variables are not set correctly on a remote system that runs C Shell. Use a different shell if you want to use environment variables.

Table 262. Environment variables. A table listing environment variables, their default values, the ranges valid values and the descriptions for each variable

Environment variable	Default value	Valid values	Description
CDP_*ATTRIBUTE_GROUP*_REFRESH_INTERVAL	Not Applicable	Any non-negative integer	The interval in seconds at which a particular specified attribute group is updated in the background. This variable works in the same way as CDP_DP_REFRESH_INTERVAL, except it targets only the specified attribute group. The attribute group name in the variable name must be in uppercase, even if the actual attribute group name is not.
CDP_DP_CACHE_TTL	55	Any integer greater than or equal to 1.	Data that is collected for an attribute group is cached for this number of seconds. Multiple requests for the same data in this time interval receive a cached copy of the data. This value applies to all attribute groups in the agent.
CDP_*ATTRIBUTE_GROUP*_CACHE_TTL	Value of CDP_DP_CACHE_TTL	Any integer greater than or equal to 1.	Data that is collected for the particular specified attribute group is cached for this number of seconds. Multiple requests for the same data in this time interval receive a cached copy of the data. This value overrides CDP_DP_CACHE_TTL for the specified group. The attribute group name in the variable name must be in uppercase, even if the actual attribute group name is not.
CDP_DP_IMPATIENT_COLLECTOR_TIMEOUT	5 if subnodes are defined, otherwise not set	Any positive integer	The number of seconds to wait for a data collection before a timeout and cached data is returned, even if the cached data is stale. (Cached data is stale if older than CDP_DP_CACHE_TTL seconds). If this variable is not set, the agent waits until the data collection completes. The wait at times can make the Tivoli Enterprise Portal timeout and give up waiting. If no thread pool is configured, this variable is ignored and data collection is done synchronously.
CDP_DP_REFRESH_INTERVAL	60 if subnodes are defined, otherwise not set	Any non-negative integer	The interval in seconds at which attribute groups are updated in the background. If this variable is not set or is set to 0, background updates are disabled. If a thread pool is configured (see variable CDP_DP_THREAD_POOL_SIZE), then the attribute groups can be refreshed in parallel. If there is no thread pool, the updates happen serially, which can take a long time. Logically equivalent to a thread pool size of 1.

Table 262. Environment variables. A table listing environment variables, their default values, the ranges valid values and the descriptions for each variable *(continued)*

Environment variable	Default value	Valid values	Description
`CDP_DP_THREAD_POOL_SIZE`	15 if subnodes are defined, otherwise not set	Any non-negative integer	The number of threads that are created to run background data collections at an interval that is defined by `CDP_DP_REFRESH_INTERVAL`. If this variable is not set or is set to 0, there is no thread pool. If `CDP_DP_THREAD_POOL_SIZE` is set to a value greater than 1 and `CDP_DP_REFRESH_INTERVAL` is set to 0, te value of `CDP_DP_THREAD_POOL_SIZE` is ignored and data collection happens on demand. The Thread Pool Status attribute group shows how the thread pool is running. Use the Thread Pool Status to adjust the thread pool size and refresh interval for best results. By default, the query for this attribute group is not displayed on the agent Navigator tree. You might not remember to include the query in a custom workspace for the agent. However, you can easily view it by assigning the Thread Pool Status query to a base agent level workspace view.
`CDP_JDBC_MAX_ROWS`	`1000`	Any positive integer	The maximum number of rows of data that the JDBC data provider returns. A result set that contains more than this number of rows is processed only up to this maximum value. Queries can be developed to prevent too much data from being returned to IBM Tivoli Monitoring.
`CDP_NT_EVENT_LOG_GET_ALL_ENTRIES_FIRST_TIME`	`NO`	YES, NO	If set to YES, the agent sends an event for every event in the Windows event log. If set to NO, only new events in the Windows event log are sent.
`CDP_NT_EVENT_LOG_CACHE_TIMEOUT`	`3600`	Any integer greater than or equal to 300.	The number of seconds Windows Event log events are cached by the agent. All cached events are returned when the event log attribute group is queried. **Note:** This variable is no longer used. Use the `CDP_PURE_EVENT_CACHE_SIZE` variable.

Table 262. Environment variables. A table listing environment variables, their default values, the ranges valid values and the descriptions for each variable *(continued)*

Environment variable	Default value	Valid values	Description
`CDP_PURE_EVENT_CACHE_SIZE`	`100`	Any positive integer greater than or equal to 1.	Maximum number of events to cache for a log file data source that is configured to process new records, for the Windows Event Log attribute group. And also for JMX monitors and notifications. Each new record in the log causes an event to be sent. This environment variable defines how many events are remembered in a cache by the agent. The cached values are returned when the attribute group is queried.
`CDP_DP_ACTION_TIMEOUT`	20 seconds	Any positive integer greater than or equal to 1.	The number of seconds to wait for a Take Action that is being handled by the agent to complete.
`CDP_DP_SCRIPT_TIMEOUT`	30 seconds	Any positive integer greater than or equal to 10.	The number of seconds to wait for the program started by a script-based attribute group to complete.
`CDP_DP_PING_TIMEOUT`	30 seconds	Any positive integer greater than or equal to 10.	The number of seconds to wait for the program started by a command return code to complete. **Note:** This variable is not related to the ICMP ping data provider.
`CDP_SNMP_MAX_RETRIES`	2	Any positive integer	The number of times to try sending the SNMP request again. The total number of requests that are sent to the SNMP agent is this value plus one if no responses are received.
`CDP_SNMP_RESPONSE_TIMEOUT`	2 seconds	Any positive integer	The number of seconds to wait for each SNMP request to timeout. Each row in an attribute group is a separate request. This timeout value is the number of seconds to wait for a response before you try again. The total timeout for a single row of data is (`CDP_SNMP_MAX_RETRIES` + 1) * `CDP_SNMP_RESPONSE_TIMEOUT`. The total default timeout value is (2+1) * 2 = 6 seconds.

Table 262. Environment variables. A table listing environment variables, their default values, the ranges valid values and the descriptions for each variable *(continued)*

Environment variable	Default value	Valid values	Description
CDP_DP_HOSTNAME	Name of the first installed network interface	An IP address or host name	Sets the preferred host name (network interface) on a multiple interface system. Use this environment variable if the agent binds its listening ports to a non-default network interface address. Used by the SNMP data provider. For Socket data sources, this variable applies if CDP_DP_ALLOW_REMOTE is also set.
CDP_SNMP_ALLOW_ DECREASING_OIDS	NO	YES, NO	If set to YES, the SNMP data providers do not check whether returned OIDs are increasing. Set to YES with caution because the monitored agent might have problems that this check would normally catch.
KUMP_DP_COPY_MODE_SAMPLE_INTERVAL	60	Wait time in seconds	For a log file data provider, specifies how long to wait before it rereads the contents of a file when the agent is defined to **Process all records when the file is sampled**. The time is specified in seconds.
KUMP_MAXPROCESS	100%	5-100%	For a log file data provider, specifies the maximum processor usage to use to process file data. Values range from 5 to 100 percent. The default is 100 percent.
KUMP_DP_SAMPLE_FACTOR	5	Any non-negative integer	For a log file data provider, sets the sampling factor when you select **Process all records when the file is sampled** on the Agent Builder. This wait time ensures that patterns that span multiple records are written before scans are logged for the pattern.
KUMP_DP_EVENT	5	Any non-negative integer	For a log file data provider, sets the sampling frequency for Event data, in seconds.
KUMP_DP_FILE_EXIST_WAIT	YES	YES, NO	For a log file data provider, specifies that the file monitoring thread continues to run if it detects that the monitored file is absent or empty. The thread waits for the file to exist, rechecks every few seconds, and starts or restarts monitoring when the file becomes available.
KUMP_DP_FILE_SWITCH_ CHECK_INTERVAL	600	Any non-negative integer	The frequency in seconds that the log file Data Provider searches for a different monitoring file to switch to when dynamic file name support is enabled.

Table 262. Environment variables. A table listing environment variables, their default values, the ranges valid values and the descriptions for each variable *(continued)*

Environment variable	Default value	Valid values	Description
KUMP_DP_FILE_ROW_ PAUSE_INCREMENT	None	Any non-negative integer	For a log file data provider, specifies how many file records are read before the file monitoring thread pauses. The pause is so that previous updates can be processed. Use this environment variable only if the monitored file receives high-volume bursts of new records and you are concerned that some record updates might be lost.
CDP_COLLECTION_TIMEOUT	60 seconds	Any positive integer	The number of seconds that the agent waits for a response from a data collector that was started in another process. JMX, JDBC, HTTP, and SOAP data collectors are examples.
CDP_SSH_TEMP_DIRECTORY	. (period)	Any valid path string on the remote system	For an SSH enabled Script data provider, specifies a location on the remote system. The script files that are provided with the agent are to be uploaded to this location. A relative lo\cation is relative to the user's home directory. The default of . (period) denotes the user's home directory.
CDP_SSH_DEL_COMMAND	rm -Rf	Any valid delete command string on the remote system	For an SSH enabled Script data provider, specifies the command to start to delete the uploaded script files that are provided with the agent.
CDP_SNMP_SEND_DELAY_ FACTOR	0 milliseconds	Any positive integer	The initial SNMP send is delayed from 0 to the number of milliseconds specified. This variable is only enabled if the thread pool is also enabled. The delay does not apply to all sends, only to the first send made by an attribute group. This variable is useful if the device that is being monitored can sometimes fail to respond correctly if it receives multiple requests at the same time.
CDP_ICMP_PING_REFRESH_ INTERVAL	60 seconds	Any integer greater than or equal to 1	The systems in a device list file are pinged at this interval. If the pings use too much time, there is always a delay of at least CDP_PING_MIN_INTERVAL_DELAY seconds before the pings begin again. Data is refreshed no more frequently than this setting. Data can be refreshed less frequently based on the number of entries in the device list file and the time it takes to receive responses.

Table 262. Environment variables. A table listing environment variables, their default values, the ranges valid values and the descriptions for each variable *(continued)*			
Environment variable	**Default value**	**Valid values**	**Description**
CDP_ICMP_PING_MIN_ INTERVAL_DELAY	30 seconds	Any integer greater than or equal to 1 and less than the CDP Ping refresh interval	After the devices in a device list file are pinged, the next ping refresh interval does not begin until at least this number of seconds elapses.
CDP_ICMP_PING_BURST	10	Any integer greater than or equal to 0	The number of pings that are sent before the agents pauses for the amount of time that is specified by the CDP_ICMP_PING_BURST_DELAY variable. A value of 0 disables this function.
CDP_ICMP_PING_BURST_DELAY	10	Any integer greater than or equal to 0	The amount of time in milliseconds to wait after a set number of pings are sent as defined by the CDP_ICMP_PING_BURST variable. A value of 0 disables this function.
CDP_ICMP_PING_TIMEOUT	2000 milliseconds	Any integer greater than or equal to 1	The number of milliseconds to wait for a ping response. This setting applies to each ping attempt that is made. Ping attempts are made 3 times for each host. If no response is received from any of the 3 attempts, the total time waited for a reply is CDP_ICMP_PING_TIMEOUT multiplied by 3. By default, this value is 6000 milliseconds. Changing the value for CDP_ICMP_PING_TIMEOUT causes the default TIMEOUT enumeration for the Current Response Time attribute to no longer apply. Change the TIMEOUT enumeration to the new value of CDP_ICMP_PING_TIMEOUT multiplied by 3.
CDP_JDBC_CONNECTIONLESS	false	true, false	If set to true, JDBC connections are closed after each data collection attempt. That is, all attribute groups attempt to create their own connection each time data is collected. Connections are not reused if this variable is enabled. If set to false, one connection to the database is made and that connection is shared among the attribute groups.
CDP_SSH_EXCLUDED_ ENVIRONMENT_VARIABLES	None	A comma-separated list of environment variable names	For an SSH enabled Script data provider, specifies the set of local environment variables that must not be set in the environment of the remote system.

Table 262. Environment variables. A table listing environment variables, their default values, the ranges valid values and the descriptions for each variable *(continued)*

Environment variable	Default value	Valid values	Description
CDP_DP_EVENT_LOG_MAX_ BACKLOG_TIME	0 seconds	0, 1, or any integer greater than 1	If set to 0, and CDP_DP_EVENT_LOG_MAX_BACKLOG_EVENTS is not set to 1 or a greater integer, does not process events that are generated while the agent is shut down. 0 is the default. If set to 1, and CDP_DP_EVENT_LOG_MAX_BACKLOG_EVENTS is not set to an integer greater than 1, processes all events that are generated while the agent is shut down. If set greater than 1, and CDP_DP_EVENT_LOG_MAX_BACKLOG_EVENTS is not set greater than 1, processes events that are generated within that value in seconds of the current computer time. For example, if the value is set to 300, at startup, the agent processes all events that are generated within 300 seconds of the current time. Where a value greater than 1 is entered for both CDP_DP_EVENT_LOG_MAX_BACKLOG_TIME and CDP_DP_EVENT_LOG_MAX_BACKLOG_EVENTS variables, either that time interval of events or that number of events is processed. Which variable is chosen depends on which is matched first.
CDP_DP_EVENT_LOG_ *Windows_Event_Log*_MAX_BACKLOG_ TIME	0 seconds (Do not process missed events while the agent is shut down)	0, 1, or any integer greater than 1	If set to

Table 262. Environment variables. A table listing environment variables, their default values, the ranges valid values and the descriptions for each variable *(continued)*

Environment variable	Default value	Valid values	Description
CDP_DP_EVENT_LOG_ MAX_BACKLOG_EVENTS	0 events	0, 1, or any integer greater than 1	If set to 0, and CDP_DP_EVENT_LOG_MAX_BACKLOG _TIME variable is not set to 1 or a greater integer, does not process events that are generated while the agent is shut down. 0 is the default. If set to 1, and the CDP_DP_EVENT_LOG_MAX_BACKLOG _TIME variable is not set to an integer greater than 1, processes all events that are generated while the agent is shut down. If set greater than 1, and CDP_DP_EVENT_LOG_MAX_BACKLOG _TIME is not set greater than 1, processes at most that number of events that are generated while the agent is shut down. For example, if the value is set to 200, then at startup of the agent the 200 events that generated directly before startup are processed. Where a value greater than 1 is entered for both CDP_DP_EVENT_LOG_MAX_BACKLOG _EVENTS and CDP_DP_EVENT_LOG_MAX_BACKLOG _TIME, either that number of events or that time interval of events is processed. Which variable is chosen depends on which is matched first.
CDP_DP_EVENT_LOG_ *Windows_Event_Log*_MAX_BACK LOG_ EVENTS	0 events (Do not process missed events while the agent is shut down)	0 or any integer greater than or equal to 1	If set to
CDP_HTTP_READ_TIMEOUT	10	Any positive integer	The number of seconds to wait for a reply to the HTTP request.
CDP_JAT_THREAD_POOL_SIZE	15	Any positive integer	The number of threads that are used by the Java providers for handling data collection requests. JMX, JDBC, HTTP, and SOAP data providers are the providers that can benefit from this thread pool.
CDP_HTML_OBJECTS_THREAD_ POOL_SIZE	10	Any positive integer	The number of threads that are used to download page objects that are found in URLs monitored with the HTTP data provider.

Table 262. Environment variables. A table listing environment variables, their default values, the ranges valid values and the descriptions for each variable *(continued)*

Environment variable	Default value	Valid values	Description
CDP_HTTP_SOAP_MAX_ROWS	500	Any positive integer	The maximum number of rows that are returned by the HTTP SOAP data provider.
CDP_DP_ALLOW_REMOTE	NO	NO, YES	If set to Yes, the agent allows remote socket connections. If set to No, the agent allows only socket connections from the local host. No is the default.
CDP_DP_INITIAL_COLLECTION_ DELAY	varies	Any positive integer	The number of seconds, after the agent starts, until the thread pool begins its scheduled data collections.

Watchdog information

Use the **Watchdog Information** page to specify configuration information for the Agent Watchdog.

About this task

To open the **Watchdog Information** page, click **Watchdog Information** in the **Agent Content** section of the **Agent Information** page. You can also select the **Watchdog Information** node in the Outline View.

You can specify the following configuration information for the Agent Watchdog:

- **Monitor this agent by default**

  Select this check box to put the agent under management by Agent Management Services when the agent is installed. The agent is monitored for unhealthy behavior or abnormal termination and is restarted by a watchdog.

- **Check frequency (seconds)**

  How often the watchdog checks the agent process for unhealthy behavior or abnormal termination. The default is every 180 seconds.

- **Maximum number of restarts**

  Number of times the Watchdog restarts the agent because of unhealthy behavior or abnormal termination in a 24-hour period before it alerts the administrator of the problem. The period starts at midnight each day. So, the first period from when the agent is started might be "short."

  A restart occurs if the agent goes down for any reason. The Watchdog also stops and restarts the agent if the agent becomes unresponsive or unhealthy, for example. if the memory threshold is crossed. The default is four restarts in a 24-hour period, where the period is measured from midnight to 11:59 p.m. At midnight, the daily restart count for the agent returns to 0 automatically.

- **Memory Threshold Information**

  Size of the agent process (in megabytes) to which the agent can grow before its watchdog deems it unhealthy. There is a separate value for Windows, Linux, and UNIX. If the agent process grows beyond the threshold, the watchdog stops the process and restarts it. There are no defaults for these properties. If no value is specified, the Watchdog does not monitor the process size. The metric uses the working set size on Windows, and the user memory on UNIX and Linux.

If the Watchdog stops the agent, and the maximum number of restarts is reached, the Watchdog sends an alert that the agent exceeded its restart count, and stops doing auto-restarts. The Watchdog still reports whether the agent is up or down assuming it is started in another manner such as through the Tivoli Enterprise Portal.

You must manually restart the agent by using the AMS Start Agent Take Action command so the restart count does not get reset.

The count gets reset in one of the following ways (the Watchdog continues to work and report status, but does not do auto-restarts):

- The clock strikes midnight.
- The user uses the AMS Start Agent Take Action command, which has an input parameter called **resetRestartCount**. If you enter a value of 1 (meaning "true"or "yes"), the daily restart count resets back to 0.

For more information, see the following sections in the *IBM Tivoli Monitoring Administrator's Guide*:

- For Tivoli System Monitor Agents

  Configuring Agent Management Services on Tivoli System Monitor Agents
- For Tivoli Enterprise Monitoring Agents

  Installing and configuring Tivoli Agent Management Services

Cognos information

Use the **Cognos Information** page to specify the information that is used when a Cognos data model is generated for your agent. This information is used only for the IBM Tivoli Monitoring environment.

Procedure

1. To open the **Cognos Information** page, click **Cognos Information** in the **Agent Content** section of the **Agent Information** page or the **Cognos Information** node in the Outline View.
2. In the **Data Source** field, enter the name of the data source that connects Tivoli Common Reporting to the IBM Tivoli Data Warehouse.

   The default value is `TDW`.
3. In the **Schema** field, enter the name of the database schema that is used for the Tivoli Data Warehouse, which is used to fully qualify table names in Cognos reports.

   The default value is `ITMUSER`. This value can be changed in Framework Manager when the generated Cognos model is loaded into Framework Manager.

   The **Add this attribute group to a reporting category** check box in the **Data Source Definition** page determines where in the Cognos model the attribute group is placed. If not selected, the attribute group is placed in the extended attributes folder in the Cognos model. If selected, the attribute group is placed in the selected subfolder (availability or performance) in the Key Metrics folder. For more information about the data source fields, see Table 263 on page 1168.

What to do next

You can use the Cognos data model to create Tivoli Common Reporting reports for your agent, see “Cognos data model generation” on page 1440.

Generate Agent wizard link

When you finish creating or editing the new agent, use the Generate Agent wizard to prepare the installation.

Procedure

- When you finish creating or editing the new agent, on the **Agent Editor Agent Information** page, click the **Generate Agent Wizard** link.

  With the Generate Agent wizard, you can:

  - Generate the agent files with a Tivoli Monitoring installation on the local system. For instructions, see “Installing an agent locally” on page 1355.
  - Create a package so the agent can be installed on other systems. For instructions, see “Creating the agent package” on page 1357.

The Data Source Definition page

Use the **Data Source Definition** page to manipulate data sources.

About this task

The **Data Source Definition** page lists the data sources that are configured for the agent. When you select a data source or attribute in the tree, the page is updated to display the properties for the selected object. Use the fields to modify the properties for the data source or attribute selected.

Note: For detailed instructions about creating data sources from various data providers, see “Defining and testing data sources” on page 1194.

Procedure

- To open the **Data Source Definition** page, click **Data Sources** in the **Agent Content** section of the **Agent Information** page or the **Data Sources** node in the **Outline** view.
- You can add more data sources by clicking **Add to Selected** or right-clicking in the navigation tree and selecting one of the options.
- You can remove data sources and attributes by right-clicking on them and selecting **Remove**.
- You can add, modify, and remove attributes. For instructions, see “Editing data source and attribute properties” on page 1168

Copying data sources by using the Data Source Definition page

Use the **Data Source Definition** page to copy data sources.

Before you begin

Go to the **Data Source Definition** page. For more information, see “The Data Source Definition page” on page 1165

About this task

Data sources that result in attribute groups can be copied to the clipboard and pasted back to this agent or another agent. Data sources that do not result in attribute groups are Availability and Windows Event Log data sources.

Procedure

1. Select the attribute groups that you want to copy.
2. Cut or copy the attribute group by using one of the following methods:

   - Click **Edit** > **Cut** > **Edit** > **Copy** from the menu bar.
   - Right-click one of the selected items and click **Cut** or **Copy** from the menu.
   - Use one of the operating system or Eclipse key strokes that calls the cut or copy action. For example, on Windows systems, pressing **Ctrl-C** calls the copy action.

   To remove data sources from their existing location and place them in the clipboard, use **Cut**. To leave data sources in place and copy them to the clipboard, use **Copy**.
3. Select the parent of an attribute group (the agent, a subnode, or a navigator group) or select an existing attribute group.
4. Paste the selection by using one of the following choices:

   - Select **Edit** > **Paste** from the menu bar.
   - Right-click the node where you want to paste the selection in the tree, and click **Paste** on the menu.
   - Use one of the operating system or Eclipse key strokes that calls the paste action. For example, on Windows systems, pressing **Ctrl-V** calls the paste action.

Results

The attribute groups from the clipboard are placed in the selected parent. Alternatively, if an attribute group is selected, the attribute groups are placed in the parent of the selected attribute group.

If there is a name conflict with another attribute group while pasting, the pasted attribute group name is changed slightly to avoid the conflict.

Runtime Configuration Information page

The **Runtime Configuration Information** page displays the configurable variables in the agent. You can set values for the variables when you install the agent on a monitored host.

These values are made available to command return codes and scripts through the environment. To open the **Runtime Configuration Information** page, click **Runtime Configuration** in the **Agent Content** section of the **Agent Information** page or the **Runtime Configuration** node in the Outline View. The Agent Builder automatically constructs the name of the environment variable from the product code and the label.

You can add and change the configuration properties and provide default values by using the **Runtime Configuration Information** page.

Agent XML Editor page

The **Agent XML Editor** page displays the XML for the agent definition.

The agent definition XML includes the information that is displayed in all other parts of Agent Builder. If you change the XML, the information displayed in Agent Builder reflects the change.

⚠ **Attention:** Do not make any changes in the XML. Such changes can cause errors that might prevent you from generating the agent or negatively affect the functioning of the agent.

Saving your edits and changes

Changes that you make with the editor are not stored until you save them.

Procedure

- Perform a save in one of the following ways:
  - Select **File** > **Save**, selecting the save (diskette) icon.
  - Press **Ctrl+S**

  When you save, a validation occurs to ensure that the information is complete. If problems occur, information about the error is displayed in the Eclipse **Problems** view. If this view is not visible, select **Window** > **Show View** > **Problems**. If you attempt to generate an agent that has errors, an error message is displayed.

  Note: You must correct all errors and save the changes before you can generate and install the agent.

Committing a version of the agent

Commit your agent when you are certain you are finished developing this version of the agent and you are ready to deliver it.

About this task

IBM Tivoli Monitoring systems require that new versions of an agent include all of the information that is contained in the previous versions of that agent that were used in the monitoring environment. Including all information from previous versions is necessary so workspaces, situations, and queries continue to work if the new agent is installed on some monitored hosts, but the old one remains on the others.

After you complete developing and testing an agent, you must commit the agent as the final version for a certain version number. Agent Builder ensures that no information is removed after you commit the agent. Subsequent builds of the agent have a new version number.

There is a limit of 1024 versions.

Remember: If you make changes to an agent that is to be tested and run in an IBM Cloud Application Performance Management environment, you must change the agent version.

Procedure

1. Open the **Agent Editor** window, **Agent Information** page.
2. In the **Commit Agent Version** area, click **commit this level**.
3. Back up the committed agent or check it into your version control system.

What to do next

After you commit an agent, any additional changes to the agent are part of a new version. You must enter the new version number before the additional changes can be saved. Any changes to the new version must not break compatibility with previous versions of the agent.

After you commit the agent, you cannot complete these actions on objects that existed before the agent was committed:

- Delete attributes from an attribute group.
- Delete attribute groups.
- Reorder existing attributes in an attribute group.
- Reorganize existing attribute groups (by using Navigator items).
- Move attribute groups or navigator groups into or out of subnodes.
- Rename attribute groups.
- Rename attributes.
- Change data types of existing attributes.
- Change a subnode name or type if it contains an attribute group that existed before the agent was committed.
- Change a company identifier or agent identifier for the agent.
- Change the product code of the agent. For more information, see ("Changing the product code" on page 1168).

You can complete the following actions after you commit the agent:

- Add new attributes to existing attribute groups.
- Add new attribute groups.
- Reorder new attributes.
- Organize new attribute groups by using navigator items.
- Create new subnode types.
- Add new queries.
- Add new situations.
- Add new workspaces.

Setting a new version number for your agent

To save changes to a committed agent, you must enter a new version number.

Procedure

1. Open the **Agent Editor** window, **Agent Information** page.
2. Enter a version, fix path, or patch level that is higher that current level after the Version prompt.
3. Make the edits your agent.

   Tip: If you commit an agent and forget to change the agent version, you are prompted for the new version when you save any of your changes.

Changing the product code

If you change the product code, you have an agent that is incompatible with any previous version of the agent. Any records of previous commit actions are lost and you are developing a new agent.

Any files, situations, Take Action commands, or workspaces that you exported from IBM Tivoli Monitoring and imported into the agent are deleted from the agent.

If you try to change the product code of an agent that was committed, Agent Builder displays a warning and asks if you want to continue.

When you click **Yes** in the **Agent Product Code** window you are warned that the contents of the agent support files are no longer valid. You are also warned that the files will be removed next time the agent is saved.

Editing data source and attribute properties

When you add data sources to your agent, Agent Builder creates corresponding data sets. You can edit the data sets and attributes in them to provide the necessary monitoring information.

Procedure

To edit or remove information from a data set (attribute group):

1. In the **Agent Content** area of the **Agent Information** page click **Data Sources**.

   The **Data Source Definition** page opens.
2. Select the data set (attribute group).

   The attribute group information area of the page is updated to display the properties for the selected data set.

   Note: Alternatively, if you are on the last page of the **Agent** wizard, you can double-click the data source to open the **Attribute Group Information** window. This window has the same information as the attribute group information area of the **Data Source Definition** page.

   (Table 263 on page 1168) describes the field information that is applicable to all of the data sources. Use the fields to modify the properties for the data source or attribute selected.

Table 263. Fields for editing data sources

Field name	Description	Acceptable values and examples
Attribute group name	Name of the data source as it is displayed in the Tivoli Enterprise Portal or in the IBM Cloud Application Performance Management console	Acceptable values: Descriptive string less than or equal to 32 characters long. It must be unique within the agent. The first character must be a letter and remaining characters can be letters, numbers, or underscores. An underscore is displayed as a space. Do not use spaces or special characters.
Help text	Help text for the data source	Acceptable values: String up 256 characters long.
Produces a single data row	The data source returns 1 row of data. Editable in all sampled data sources.	Example: If you are monitoring physical system memory, choose a single row. A system typically manages all of its memory in a single pool; so only one row of data can be returned.

Table 263. Fields for editing data sources (continued)

Field name	Description	Acceptable values and examples
Can produce more than one data row	The data source can return any number of rows of data. Editable in all sampled data sources.	Example: If you are monitoring disk drives, choose multiple rows because there can be more than one disk in a system. For keys, choose the attributes that distinguish a disk from another. For a disk, the key attribute is disk number, drive letter, volume label, or whatever is appropriate in your environment.
Produces Events	The data source returns event-based data, 1 row of data per event.	Example: An SNMP event-based data source sends notifications (traps) as performance thresholds are crossed. **Note:** Not all data sources can produce events.
Add this attribute group to a reporting category	The category in the generated Cognos model to which the attributes in this attribute group are assigned.	Select the check box to place the attribute group in the selected subfolder (Availability or Performance) in the Key Metrics folder. If the check box is not selected, the attribute group is placed in the **Extended Metrics** folder in the Cognos data model.
Metric Category	The category to which the attributes in this attribute group are assigned.	Select either **Performance** or **Availability**.

Note:

a. The **Produce a single data row** and **Can produce more than one data row** fields do not affect data for an event data source.

b. For more about sampled and event data types, see ("Data types" on page 1189).

c. For information about the fields for a specific data source, see the relevant data provider information in "Defining and testing data sources" on page 1194.

Creating, modifying, and deleting attributes

You can create, modify, or delete attributes in a data set (attribute group).

To work with attributes, open the **Data Source Definition** page. For more information, see "The Data Source Definition page" on page 1165.

Creating attributes

You can add new attributes to a data set.

Procedure

1. Right-click the data source and select **Add Attribute** on the menu.

The **Attribute Information** page is displayed.

Note: The page that is displayed depends on the data source for the attribute.

2. Specify your choices for the new attribute on the **Attribute Information** page.

   See "Fields and options for defining attributes" on page 1172 for information about the fields and options.
3. To add more attributes, select **Add additional attributes** and click **Next**.
4. When finished adding attributes, click **Finish**.

Copying attributes

You can copy attributes from the **Data Source Definition** page.

Procedure

1. In the Agent Editor, **Data Source Definition** page, right-click the attribute that you want to copy, and click **Copy Attribute**.
2. In the **Copy Attribute** window, type the name of the new attribute in the **Name** field, and click **OK**.

Editing attributes

You can edit and change attribute information by using the **Data Source Definition** page.

Procedure

1. Select the attribute that you want to edit.

   The **Attribute Information** pane of the page is updated to show the properties for the selected attribute.
2. Specify your choices for the new attribute information.

   Note: On the last page of the **Agent** wizard (the **Data Source Definition** page), you can double-click the attribute to open the **Attribute Information** window. That window contains the same information as the Attribute Information pane of the **Data Source Definition** page.

Creating derived attributes

You can create an attribute that derives its value from other attributes instead of directly from the data source.

About this task

In the derived attribute, you can perform operations on the values of the source attributes. For example, you can perform basic arithmetic operations on numeric attributes or string concatenation on string attributes.

The basic expression syntax that is used for derived expressions contains functions. These functions provide a more complicated manipulation of data that includes short-term aggregation, conversion from string to integer, and accessing configuration properties and environment variables. In addition, an editor helps you visualize the expression as it is being built.

Procedure

1. On the **Data Source Definition** page, right-click the data source and click **Add Attribute**.
2. On the **Attribute Information** page, type an Attribute name and Help text.
3. Select **Derived from other attribute values**.
4. In the **Formula** field, type the formula text or click **Edit** to enter the formula with a graphical editor.

   See "Formula operators and functions" on page 1183 for information about the operators and functions that can be used in the formula.

   Note: When you click **Edit**, the Formula Editor opens. See "Editing derived attributes" on page 1172 for information about editing derived attributes.

5. Optional: Select or clear the **Interval specific calculations** check box to determine which two attribute sample values are used when the function is calculated.

   Use this option when your formula uses the `rate` or `delta` functions. For more information about **Interval specific calculations**, see "Interval specific calculations" on page 1171. For more information about `rate` and `delta` functions, see "Formula operators and functions" on page 1183.
6. In the **Attribute type** area, click the type of attribute.
7. Click **OK**.

   The **Data Source Definition** page is displayed again with the data source listed in it as before.
8. Click **Finish**.

   Important: If you create a derived attribute that references another derived attribute, ensure that the referenced attribute is listed earlier than the new attribute. If an attribute references another derived attribute that is located later in the list, the agent is unable to display the value for this attribute. If you create such an attribute, Agent Builder displays a warning.

Interval specific calculations

You can choose **Interval specific calculations** when you define a derived attribute that is based on the `rate` or `delta` functions.

You select **Interval specific calculations** on the **Derived Attribute Details** tab of the **Attribute Information** page. For more information, see "Creating derived attributes" on page 1170.

When you use the **Interval specific calculations** selection, it is important to understand the concept of a delta or difference between attribute values. The delta is the difference between the most recent value of the attribute and a previous value of the attribute. The delta is returned directly by the `delta` function and is used by the `rate` function to calculate a result.

The `delta` or `rate` function must always have the `last` function as its only argument. The `last` function specifies which values of an attribute are used to determine the delta. If **Interval specific calculations** is not selected, the previous value that is used is always the second-most-recent value. If **Interval specific calculations** is selected, the previous value that is used is the value whose age (relative to the most recent value) is equal to the collection interval of the requester.

For example, suppose `CDP_DP_REFRESH_INTERVAL` is set to `120` seconds and `attribute A` has the following sampled values:

Time	**Sampled value**
current	`2800`
2 minutes (120 seconds) ago	`2600`
4 minutes (240 seconds) ago	`2499`
6 minutes (360 seconds) ago	`1500`
8 minutes (480 seconds) ago	`1200`
10 minutes (600 seconds) ago	`1000`

When **Interval specific calculations** is not selected, the `delta` function always returns `200`, the difference between the two most recent values, `2800 - 2600`. The same value is returned whether the value is displayed on the Tivoli Enterprise Portal or in the IBM Cloud Application Performance Management console, used in a situation, or a historical collection.

When **Interval specific calculations** is selected, the `delta` function returns a value that depends on the collection interval of the requester.

If a derived attribute with the `delta` function is used in a situation with a 4-minute collection interval, the value that is returned by the `delta` function is `301`, the difference between the most recent value and the value obtained 4 minutes before that, `2800 - 2499`.

If a derived attribute with the `rate` function is used in a situation with a 10-minute (600-second) collection interval, the value that is returned by the `rate` function is 3, the difference between the most recent value and the value obtained 10 minutes before that, divided by the number of seconds in the interval `(2800 - 1000) / 600`.

Note: The Tivoli Enterprise Portal has no inherent collection interval, so delta and rate calculations for Tivoli Enterprise Portal requests always use the most recent and second most recent attribute values, the same result whether **Interval specific calculations** is selected or not.

For `delta` or `rate` to work correctly with **Interval specific calculations**,

- The agent must collect data periodically in the background, and not on demand (`CDP_DP_THREAD_POOL_SIZE` must be greater than 0).
- Every situation or historical collection interval in which the attribute is used must be a multiple of the background refresh interval (`CDP_DP_REFRESH_INTERVAL`).
- The count (the second argument of the last function) must be large enough to accommodate the largest collection interval from a situation or historical collection. For example, if the agent must support 10-minute (600 second) historical collection and `CDP_DP_REFRESH_INTERVAL` is 120 seconds, the count must be at least `6, 1+(600 / 120)`. A count value of 6 ensures that the `last` function returns the newest sample and samples up to 600 seconds old.

Note: If these conditions are not met, input values are likely invalid and a result of `0` is returned.

Editing derived attributes

Use the Formula Editor to edit derived attributes.

The Formula Editor is available on the **Attribute Information** page for a derived attribute, as described in "Creating derived attributes" on page 1170. For more information about the Formula Editor, see "Formula Editor" on page 1178

Removing attributes

You can remove one or several attributes from a data set using the **Data Source Definition** page.

Procedure

- To remove an attribute or attributes, right-click the attribute or attributes and select **Remove** from the menu that is displayed.

  Note: You cannot remove an attribute that is used by a derived attribute. You must first remove the reference by the derived attribute to the attribute you are removing.

Fields and options for defining attributes

Description of the field information and options for the **Attribute Information** page that are applicable to all of the data sources

For information about the specific field information for each of the data sources, see the relevant documentation for each data source.

Table 264. Fields and options for defining attributes

Field names/options	Description	Acceptable values
Attribute name	Name of the attribute as it is displayed in the Tivoli Enterprise Portal or in the IBM Cloud Application Performance Management console	String with the following characters: • A-Z • _ • a-z • 0-9 **Note:** The name must start with A-Z or a-z. The attribute name has a limit of 63 characters and the attribute group name has a limit of 63 characters
Help text	Help text for the attribute	String
Hidden - can only be used in derived attribute	If selected, the attribute is not displayed in the Tivoli Enterprise Portal or in the IBM Cloud Application Performance Management console. See note in the last row.	Not applicable
Derived from other attribute values	Attribute value is to be calculated from values of other attributes	Not applicable
Key Attribute	Attribute is a key in the table. Check whether this attribute helps to uniquely define the object that is being reported on. If the data is warehoused and summarized, the key attributes are used to roll up data in the summary tables.	This option is not available for Perfmon attributes.
Attribute Information pane	The contents of this tab depend on the type of data source to which this attribute belongs. See information in the chapter for the data source you want to monitor for more details. For a derived attribute, In the **Formula** field, enter a formula to calculate the value of the attribute that is based on other attributes or constants. You can type the formula in the **Formula** field or click **Edit** to use the graphical formula editor. See ("Formula Editor" on page 1178).	

Table 264. Fields and options for defining attributes (continued)

Field names/options	Description	Acceptable values
Attribute type	Describes how the attribute is displayed in the Tivoli Enterprise Portal or in the IBM Cloud Application Performance Management console. There are 3 types: • String • Numeric • Time stamp "Attribute types" on page 1174 contains more information about the attribute types.	Table 265 on page 1175 contains descriptions of the numeric attribute type values.
Enumerations	Can be a numeric with scale zero or string value.	Add your enumerations to the table by using the procedure in ("Specifying an enumeration for an attribute" on page 1177). The enumeration name is displayed in the Tivoli Enterprise Portal or in the IBM Cloud Application Performance Management console when the corresponding Value is received in the attribute from the agent. This attribute is used for a set of specific values with identified meanings (for example, 1=UP, 2=DOWN).
Note: In cases where the attribute is used in calculations with other attributes, there are reasons not to display the base value. For instance, a number that represents a byte count wraps so quickly that it is of little use.		

Attribute types

There are three attribute types

The three types of attributes are:

- String
- Numeric
- Time stamp

String attributes

When you select **String**, use the **Maximum size** field to specify the maximum length of the string in bytes. The default size is 64 bytes.

A string value can contain any UTF-8 character. The maximum size is the total length of the buffer that is allocated to contain the string in bytes. Some non-ASCII UTF-8 characters take more than 1 byte, so you must account for this space when you select a maximum size. Data aggregation in the warehouse displays the latest value that is collected during the period.

Numeric

When you specify **Numeric**, you can set a number of options. See Table 265 on page 1175 for information about these options.

Time stamp

A Time stamp attribute is a string attribute with a format that conforms to the `CYYMMDDHHMMSSmmm` format (where C=1 for the 21st century). All 16 characters must be used for scripts or socket clients. When displayed in the Tivoli Enterprise Portal or in the IBM Cloud Application Performance Management console, a time stamp attribute type is displayed in the correct format for the locale.

When you use the browse feature for WMI, the Agent Builder automatically marks attributes whose CIM type is `CIM_DATETIME` as time stamps. The data provider automatically converts WMI attributes to this format.

Numeric aspects of attributes
Descriptions of the size, purpose, scale, and range aspects of attributes.

When you specify a numeric attribute, you must specify the size, purpose, scale, and range of the attribute. For more information, see (Table 265 on page 1175).

Table 265. Numeric attribute options

Numeric aspects	Options and fields	Description
Size	**32 bits** **64 bits**	The value of 32-bit numbers can range from -2147483648 to 2147483647 (roughly -2,000,000,000 to 2,000,000,000). The value of 64-bit numbers can range from -9223372036854775808 to 9223372036854775807 (roughly -9x10^{18} to 9x10^{18})

Table 265. Numeric attribute options (continued)

Numeric aspects	Options and fields	Description
Purpose	**Gauge**	Integer values where the raw values returned are larger or smaller than previous values. Negative values are supported. This type is the default type for integers. Data aggregation in the warehouse produces minimum, maximum, and average values.
	Counter	A positive integer value that contains raw values that generally increase over time. Data aggregation in the warehouse displays the total, high, low, and latest delta values. In the following example of Delta-based calculations, detailed data values in one hour are 9, 15, 12, 20, 22, and delta-based processing has the following rules: • If the current value is greater than or equal to the previous value, the output equals the previous value minus the current value • If the current value is less than the previous value, the output equals the current value • Because 15 is greater than 9, the output equals 6 • Because 12 is less than 15, the output equals 12 • Because 20 is greater than 12, the output equals 8 • Because 22 is greater than 20, the output equals 2 • The TOT_ value is 28, which is the total of outputs • The LOW_ value is 2, which is the lowest of outputs • The HI_ value is 12, which is the highest of outputs
	Property	A property of the object that does not frequently change. Data aggregation in the warehouse displays the latest value that is collected during the period.
	Delta	An integer value that represents the difference between the current value and the previous value for this attribute. Because this attribute is represented as a gauge in the warehouse, data aggregation in the warehouse produces minimum, maximum, and average values.
	Percent change	An integer value that represents the percent change between the current value and the previous value. This type is calculated as: `((new -old)*100)/old`. Because this type is represented as a gauge in the warehouse, data aggregation in the warehouse produces minimum, maximum, and average values.
	Rate of change	An integer value that represents the difference between the current value and the previous value, which is divided by the number of seconds between the samples. It converts a value (such as bytes) to the value per second (bytes per second). Because this type is represented as a gauge in the warehouse, data aggregation in the warehouse produces minimum, maximum, and average values.

Table 265. Numeric attribute options (continued)

Numeric aspects	Options and fields	Description
Scale	**Decimal adjustment**	Scale determines how many decimal places are in the number. Each decimal place reduces the range that is mentioned earlier by a factor of 10. For example, a decimal adjustment of 2 shows two decimal places, and in a 32-bit number the allowable range becomes `-21474836.48` to `21474836.47`. When a non-zero decimal adjustment is specified, the number is manipulated internally as a floating point number. Therefore, the precision of large 64-bit numbers might be reduced.
Range	**Minimum** **Maximum**	Range gives the expected range of the value. If no minimum or maximum ranges are given, the maximum values that are described earlier are used. The range is used to produce a more useful initial view in some graphical Tivoli Monitoring workspace views.
Units		Unit of measurement for a numeric attribute.

Specifying an enumeration for an attribute

Specify a value enumeration by using the **Attribute Information** page.

About this task

Specifying an enumeration for an attribute involves a short procedure. When a value is encountered that has a defined enumeration, the enumeration name is displayed in the Tivoli Enterprise Portal or in the IBM Cloud Application Performance Management console instead of the value.

Procedure

1. In the **Attribute Information** page **Attribute type** area, click **Numeric**.
2. In the **Enumerations** area, click an enumeration, and click **Add**.

   The **Enumeration Definition** window is displayed.
3. Type the name and value of the enumeration in the fields in the window.
4. Click **OK**.

   You can then add more enumerations.

Specifying severity for an attribute used as a status indicator

In an IBM Cloud Application Performance Management environment, a summary dashboard must display a status. You must use an attribute to provide the status value. For this attribute, you must specify values that denote specific status severity.

About this task

The attribute that is used for status indication must be numeric. Select this attribute in the **Dashboard Setup** wizard; for instructions about using this wizard, see "Preparing the agent for Cloud APM" on page 1342.

You can specify values for the attribute that correspond to the Normal, Warning, and Critical severity. Any other value denotes an "Unknown" severity status; you can also define some values as "Not defined" explicitly, and the "Unknown" status user interfaces displayed for these values.

Procedure

1. Select the attribute that you want to edit.

The Attribute Information pane of the page is updated to show the properties for the selected attribute.

2. In the Attribute Information pane, click the **Severity** tab.
3. Select the necessary severity (Normal, Warning, Critical, and Not defined) and click **Edit**.
4. Select **Range** or **Single number**, enter the range of values or the single numeric value, and click **Ok**.
5. Optional: If you need to add another value for the same severity, for example; both 2 and 25 denote warning, click **Add**, select the severity, enter the value, and click **OK**.

Filtering attribute groups

You can create a filter to limit the data that is returned from an attribute group that returns sampled data.

Before you begin

If the attribute group exists, open the **Data Source Definition** page. For more information, see "The Data Source Definition page" on page 1165.

If you want to create an attribute group, follow the steps in "Defining initial data sources" on page 1151and click **Advanced** in the initial data source information page.

Procedure

1. Use one of the following steps to begin creating the filter:
   - If you are creating an attribute group, click **Advanced** in the initial data source information page.
   - If the attribute group exists, select the attribute group in the **Data Source Definition** page and click **Advanced** in the **Data Source Definition** page.
2. In the **Advanced Data Source Properties** page, enter a selection formula. The selection formula that you enter must evaluate to a Boolean result, true, or false.

   In the **Advanced Data Source Properties** page, you can click **Edit** to enter or modify the formula by using the Formula Editor. For more information about the Formula Editor, see "Formula Editor" on page 1178
3. When you finish entering the filter selection formula, click **OK** until you return to the **Data Source Definition** page.

   When the filter is created, the agent uses the filter to evaluate each row of data. When the filter evaluates to *true* for a row of data, the data is sent to IBM Tivoli Monitoring or IBM Cloud Application Performance Management. When the filter evaluates to *false*, the row of data is not sent and is discarded.

What to do next

You can validate that the filter is working as intended by using the test function for the attribute group. For more information about attribute group testing, see "Attribute group testing" on page 1347

Formula Editor

Use the Formula Editor to create and change formulas in Agent Builder.

The Formula Editor, which is a graphical tool, is displayed when you do one of the following tasks:

1. Creating or editing derived attributes, see "Creating derived attributes" on page 1170 and "Editing derived attributes" on page 1172
2. Creating Filtered Attribute groups, see "Creating a filtered attribute group" on page 1311
3. Filtering data from attribute groups, see "Filtering attribute groups" on page 1178

Attention:

- When you create derived attributes, the formula that you create must result in a data type that matches the type of the attribute. For example, if the derived attribute type is a number, the formula you create must evaluate to a numeric result.

- When you create filtered attribute groups or filter data from attribute groups, the formula that you create must result in a Boolean value, "true" or "false".

Note: In the following views, the Formula Editor is shown creating formulae for derived attributes. The views are identical when you use the Formula Editor with filtered attribute groups or to filter data from attribute groups. The views show the heading **Derived Formula Editor** or **Filter Formula Editor** depending on use.

When the Formula Editor is displayed, the current formula is loaded into the editor. If a formula does not exist, you can enter one by typing directly into the formula space in the **Formula Editor** window. Alternatively you can click **Insert** to begin entering a formula by using the editor menu options. The editor contains two views of the formula in the default window, and an option for a third view:

Component view (default)
The components of the edited formula are shown in the **operand** areas and **Operator** field. The operator and its two operands can be edited by using the selection menus.

Formula view (default)
The complete formula is in the formula field in the window. You can edit the formula by typing in this box.

Formula hierarchy tree view (option)
The formula hierarchy tree is displayed by selecting the **Show formula hierarchy** check box. The state of the check box is remembered in subsequent invocations of the Formula Editor.

Changing the Formula Editor component view

Change the component view in the Formula Editor.

About this task

The component that is shown in the component view can be changed in the following ways:

Procedure

- Move the cursor in the formula text.
- Select a different node in the formula hierarchy tree.
- Select **Up one Level** or one of the Edit buttons.

Component types

You can use the Formula Editor to edit the current component and any operands or function arguments of that component. Some components can appear differently in the Formula Editor when selected.

Formula Editor Attribute component

Use the attribute component in the Formula Editor to select and manipulate attributes in formulae.

About this task

You can select an attribute from a list of attributes for the attribute group in the component view of the Formula Editor.

Procedure

1. To work with a specific attribute, select that attribute from the list and click **Edit**

   The **Edit the Selected Attribute** window is displayed.
2. You can manipulate the selected attribute in the following ways:
   - You can replace the attribute with a string or number by selecting **String** or **Number**. The attribute list is replaced by an entry field and the contents are no longer compared to the list of valid attribute names.
   - You can replace the attribute with a function by clicking **Function**. Parentheses are added after the name and the list now contains valid function names to choose from.

- You can type an attribute name instead of selecting one. Typing a name is useful if you did not yet define all of the attributes in this attribute group.
  - A warning is displayed if there is no attribute with the name that was entered.
  - An error is displayed if characters are entered that cannot be part of an attribute name.
  - The **OK** button is disabled until the warning or error is corrected.
- Attributes are not filtered based on type. If an attribute (or any value) of the wrong type is selected or entered, a warning message is displayed.

Formula Editor Literal components

Use the string and number components in the Formula Editor to manipulate literals in formulae.

About this task

A literal is any value that is entered directly in the formula that does not come from an attribute value or from a function. A literal value can be either a string or a number.

Procedure

- You can replace a literal string or number with an attribute by clicking **Attribute**. A valid attribute name must be selected or entered without quotation marks.
- You can replace a literal string or number with a function by clicking **Function**. Parentheses are added after the name and the selection list contains valid function names to choose from.
  - A warning is displayed if a number is entered where a string is expected or vice versa.
  - If **Number** is selected, an error is displayed if the content of the field is not a number. **OK** is disabled until the error is corrected.

Formula Editor Operator component

Use the operator component in the Formula Editor manipulate operators in formulae.

About this task

An operator component shows an operator and its operands.

Procedure

- In the Formula Editor component view select the operator from the **Operator** list, between the two operands. The (%) operator multiplies the first operand by 100, and then divides by the second operand.
- Select the operator `(+ - * / or %)`.
  - The **Left operand** section of the page is before the operator.
  - The **Right operand** section is after the operator.
  - Simple operands (attributes and literals) can be edited without having to change the selected component to the operand as described in "Formula Editor Attribute component" on page 1179 and "Formula Editor Literal components" on page 1180.
  - Complex operands, which consist of other operators or functions, can be edited by clicking **Edit**. This action highlights the operand component instead of the entire operator.

Formula Editor Conditional expression component

The conditional expression component shows a condition, a value to return if the condition is true, and a value to return if the condition is false.

- The expression in the **Condition** section must evaluate to true or false. Operators `(==), (!=), (<), (<=), (>), (>=), (&&), (||), (!)` are available to form expressions that return true or false.

- Simple operands (attributes and literals) can be edited without having to change the selected component to the operand as described in "Formula Editor Attribute component" on page 1179 and "Formula Editor Literal components" on page 1180.
- Complex operands, which consist of other operators or functions, can be edited by clicking Edit. This action highlights the operand component instead of the entire conditional expression.
- See "Formula Editor common options" on page 1181 for information about using the following options: **Insert**, **Remove**, **Up one Level**, and **Edit**.

Related concepts

"Formula Editor" on page 1178
Use the Formula Editor to create and change formulas in Agent Builder.

Formula Editor Function component

Use the function component in the Formula Editor to select and manipulate function components in formulae.

About this task

The function component shows the function and its arguments.

Procedure

- To work with the functions Select the **Function name** from the list in the Formula Editor.
  - The description of the selected function is shown after the function.
  - **Function argument** sections are shown after the function name. The appropriate number of arguments for the selected function are shown. A description specific to the function selected is shown.
  - Simple arguments (attributes and literals) can be edited without having to change the selected component to the operand as described in "Formula Editor Attribute component" on page 1179 and "Formula Editor Literal components" on page 1180.
  - Complex arguments, which consist of operators or other functions, can be edited by clicking **Edit**. This action highlights the argument component instead of the entire function.
- For functions that take a variable number of arguments, add arguments by clicking **Insert** or remove arguments by clicking **Remove** in addition to the actions described in "Formula Editor common options" on page 1181.
- For the `getenv` function, a configuration property can be chosen by clicking **Insert**. If you select the Configuration property choice, the **Configuration Properties** window is displayed.

Formula Editor common options

You can use some options in all views in the Formula Editor

The Formula Editor common options are:

- **Insert**
- **Remove**
- **Up one Level**
- **Edit**

Insert

Insert inserts an operator or a function before the component. The component is demoted to one of the operator operands or one of the function arguments. For example, if you click **Insert** before the `sqrt(attr2)` function, you are asked what you want to insert and the following choices are displayed:

- **An operator with sqrt(attr2) as one of the operator's operands**
- **A function with sqrt(attr2) as the function's first argument**

- **A conditional expression with sqrt(attr2) as the true or false values**

If you click **Insert** before the `getenv` function, you are asked what you want to insert and the following choices are displayed:

- **Configuration property**: use this option to retrieve the value of a configuration property that you have set up for the agent, or else of any environment variable (for example, `JAVA_HOME`) on the host running the agent.
- **An operator with attr2 as one of the operator's operands**
- **A function with attr2 as the function's first argument**
- **A conditional expression attr2 as the true or false values**

Remove

Remove is available only for operators and functions, and is the inverse of **Insert**. When you click **Remove**, you are asked what is to replace the removed operator or function. For example, **Remove** before the `sqrt(attr2)` function shows the following choices:

- **The current argument 1, attr2**
- **A new string, number, or attribute reference**

Select **A new string, number, or attribute reference** to discard the entire tree after the point that is being removed and replace it with a new attribute or literal value.

Click **The current argument** to promote the selected operand or argument to replace the removed operator or function. You can click subsequent choices if there are more arguments or operands. Any other operands or arguments are discarded.

Up one Level

Click **Up one Level** to move up in the tree.

Edit

Click **Edit**, before a complex operand or argument, to make it the component to be edited.

Click **Up One Level** after you click **Edit** to restore the current component to what it was before you clicked **Edit**.

Formula Editor - Formula errors

Correcting formula errors in the **Formula Editor**

The component view is different when there is no formula or the entered formula cannot be parsed. It does not display a formula tree. Instead, it displays an error message.

You can correct a formula with parsing errors by typing directly in the formula field, or by replacing it with a new formula by clicking **Insert**. In this case, **Insert** presents the following choices:

- **An attribute**
- **A string**
- **A number**
- **An operator**
- **A conditional expression**
- **A function**

Related concepts

"Formula Editor" on page 1178

Use the Formula Editor to create and change formulas in Agent Builder.

Formula operators and functions

A reference (including examples) of formula operators and functions that are used in the formula editor.

A derived attribute value is the result of evaluating an expression that is based on constants and other attribute values in the same data source. The expression grammar is the normal mathematical expression - operand operator operand with parentheses used for grouping. Numeric attributes can be combined with other numeric attributes or constants by using the normal mathematical operators: `+ - * /, and` %, which multiplies the **Left operand** by 100 and divides by the **Right operand**. String attributes can be combined with other string attributes or constants with +. You can also use the following described functions. Functions are entered in the format: `function_name(argument_1, argument_2, argument_3)`.

An attribute is represented by its name (the same name you see in the **Data Sources** Information tree). Integer constants are specified as numbers. String constants are surrounded by quotation marks.

You can use the following functions in a formula:

abs
Returns the absolute value of a number

atof
Converts a string to a floating point value

atoi
Converts a string to an integer value. It operates in the same way the normal **C atoi** works: it stops at the first non-decimal character.

average
Returns a single value that is the average of a set of values. The set of values comes from the arguments of the function. Several individual values can be given (for example attribute names or constants), each in a separate argument. Alternatively the last function can be the only argument to this function (to calculate the average of the most recent values of an attribute).

Examples of this function in use are:

```
average (Attr_A, AttrB, Attr_C)
```

```
average (last (Attr_A, 10))
```

ceiling
Returns the least integer that is not less than the argument.

For example, where `attribute_a = 12.4`, `ceiling(attribute_a)` returns the value 13. And, where `attribute_a = -12.4`, `ceiling(attribute_a)` returns the value -12.

delta
The difference between the most recent value of an attribute and a previously collected value of that attribute. The single argument to delta must be the `last` function, which obtains the current and previous values of an attribute. A normal use might look like:

```
delta (last(OtherAttribute, 2))
```

For more information about which attribute values from the last function are used to calculate the delta, see "Interval specific calculations" on page 1171. This function is applicable only for derived attributes, not for attribute group filters.

floor
Returns the greatest integer that is not greater than the argument.

For example, where `attribute_a = 12.4`, `floor(attribute_a)` returns the value 12. And, where `attribute_a = -12.4`, `floor(attribute_a)` returns the value -13.

getenv
Returns the value of the provided environment or "configuration variable".

ipAddressToName
Converts an IP address to a host name. This function requires one argument, an IP address string in dotted decimal notation. If the address cannot be resolved, then the IP address is returned.

itoa
Converts an integer into a string. This function is most useful when you want to concatenate a numeric value onto a string. The derived string + function takes only two string arguments.

last
Returns a list of values for use by the `min`, `max`, `average`, `stddev`, `rate` and `delta` functions. It takes two arguments: the attribute to collect and the number of values to use in the calculation. If the required attribute is an integral value in a string attribute, the first argument can contain the `atoi` function, such as `atoi(numericalStringAttribute)`. The second argument must be a number. It can either be hardcoded as a constant or it can be the result of an `atoi(getenv("ENV_VAR"))` expression. It cannot reference an attribute value.

Examples of this function in use are:

```
average (last (Attr_A, 10))
```

```
last (Attribute_A, ${K01_NUM_COLLECTIONS}))
```

Restriction: You can use the `last` function only once in a specific formula.

matches
Returns a Boolean, true, or false, indicating whether a regular expression matches a value. It takes two arguments, string source and a regular expression whose result the string is compared to. This function is useful for filtering attribute groups.

max
Returns a single value that is the maximum of a set of values. The set of values comes from the arguments of the function. Several individual values can be given (for example attribute names or constants), each in a separate argument. Alternatively the last function can be the only argument to this function (to calculate the maximum of the most recent values of an attribute).

min
Returns a single value that is the minimum of a set of values. The set of values comes from the arguments of the function. Several individual values can be given (for example attribute names or constants), each in a separate argument. Alternatively the last function can be the only argument to this function (to calculate the minimum of the most recent values of an attribute).

nameToIpAddress
Converts a host name to an IP address. This function requires one argument, a host name string. If the address cannot be resolved, then the host name is returned.

NetWareTimeToTivoliTimestamp
Converts a Novell NetWare hexadecimal time value to a Tivoli Monitoring time stamp. This function requires one argument, a special NetWare hexadecimal time value. The attribute type is timestamp.

rate
The rate of change (per second) between the most recent value of an attribute and a previously collected value of that attribute. The single argument to rate must be the `last` function, which obtains the current and previous values of an attribute. A normal use might look like:

```
rate (last(OtherAttribute, 2))
```

For more information about which attribute values from the last function are used to calculate the rate, see "Interval specific calculations" on page 1171. This function is applicable only for derived attributes, not for attribute group filters.

replaceFirst

Replaces the first occurrence of a substring that matches a regular expression with a replacement string. This function takes three arguments. First: the input string. Second: the regular expression which is used to match a substring in the input string. Third: the replacement string. See ("ICU regular expressions" on page 1455) for details on the regular expressions and substitution values that are allowed in the replacement string.

replaceAll

Replaces all occurrences of substrings that match a regular expression with a replacement string. This function takes three arguments. First: the input string. Second: the regular expression which is used to match a substring in the input string. Third: the replacement string. See ("ICU regular expressions" on page 1455) for details on the regular expressions and substitution values that are allowed in the replacement string.

round

Mathematically Rounds the number to the nearest whole number.

sqrt

Returns the square-root of a number

stddev

Returns a single value that is the standard deviation of a set of values. The set of values comes from the arguments of the function. Several individual values can be given (for example attribute names or constants), each in a separate argument. Alternatively the last function can be the only argument to this function (to calculate the standard deviation of the most recent values of an attribute).

StringToTivoliTimestamp

Converts a date and time string to a Tivoli Monitoring time stamp. This function requires two arguments. The first argument is a free-form string representation of the time stamp. The second argument is a format string that identifies how to parse the free-form string representation of a time stamp. (Table 266 on page 1185) describes the valid format parameters. The attribute type is timestamp.

Table 266. Valid format parameters for StringToTivoliTimestamp

Symbol	**Meaning**	**Format**	**Example**
y	Year	yy yyyy	*96* *1996*
M	Month **Note:** Only English month strings are supported.	M or MM MMM MMMM	*09* *Sept* *September*
d	day	d dd	*2* *02*
E	Day of week **Note:** Only English day-of-week strings are supported.	EE EEE EEEE	*Sa* *Sat* *Saturday*
h	Hour in AM or PM (1-12)	hh	*07*
H	Hour in day (0-23)	HH	*00*
m	Minute in hour	mm	*04*

Table 266. Valid format parameters for StringToTivoliTimestamp (continued)			
Symbol	**Meaning**	**Format**	**Example**
s	Second in minute	`ss`	*05*
S	Millisecond	S SS SSS	*2* *24* *245*
a	AM or PM marker	a or aa	*am*
Any other ASCII character	skip this character	`-` (hyphen) ` ` (space) / (forward slash) `:` (colon) `*` (asterisk) `,` (comma)	

Table 267 on page 1186provides examples of string representations of time stamps and the format strings that are used to parse them.

Table 267. StringToTivoliTimestamp examples. A table listing and explaining a few examples of string representations of time stamps.

String representation of the time stamp	**Format string**
96.07.10 at 15:08:56	`yy.MM.dd ** HH:mm:ss`
Wed, August 10, 2010 12:08 pm	`EEE, MMMM dd, yyyy hh:mm a`
Thu 21/01/2010 14:10:33.17	`EEE dd/MM/yyyy HH:mm:ss.SS`

sum
Returns a single value that is the sum of a set of values. The set of values comes from the arguments of the function. Several individual values can be given (for example attribute names or constants), each in a separate argument. Alternatively the last function can be the only argument to this function (to calculate the sum of the most recent values of an attribute).

TivoliLogTimeToTivoliTimestamp
Converts a Tivoli log file time stamp to a Tivoli Monitoring time stamp. This function requires one argument, the string time stamp from a Tivoli log file. The attribute type is timestamp.

tokenize
One token of a tokenized string. This function requires three arguments. The first argument is a string to be split into tokens. The second argument gives one or more characters in the string that separate one token from another. Any occurrence of any of the characters from this argument is used to identify and separate tokens in the first argument. The third argument is the index of the token to return as a result of this function. The first token is index 0, the second token is index 1, and so on. This argument can also be the string LAST to return the last token.

UTCtoGMT
Converts Coordinated Universal Time to a GMT Tivoli Monitoring time stamp. This function requires one argument, the integer time_t value. The attribute type is timestamp.

UTCtoLocalTime
Converts Coordinated Universal Time to a local Tivoli Monitoring time stamp. This function requires one argument, the integer time_t value. The attribute type is timestamp.

The following functions take no arguments and return a number.

count
Keeps a counter that starts at 1 the first time it is called, and increments by 1 each subsequent time it is called. If you use it in an expression that also uses `last`, it matches the number of elements that are stored by `last()`, but only until `last()` reaches its maximum. At that point, `last()` starts deleting the oldest value for each new one, thus staying at the same number of total values, while `count()` keeps increasing forever.

cumulativeSum
Returns the sum of argument values of duplicate events that are represented by a flow control summary event. Or returns the argument if it is a single event from a data source. It takes a single numeric argument. This function applies only to event attribute groups with `event filtering` and summarization turned on.

eventThreshold
Returns the threshold value that is configured for the attribute group which generated the event. A number, with three enumerations:

- `SEND_ALL (-3)`
- `SEND_FIRST (-2)`
- `SEND_NONE (-1)`

The number in parentheses is the raw value. However, the Agent Builder defines the enumerations so by default the text version is visible on the Tivoli Enterprise Portal or in the IBM Cloud Application Performance Management console. If you specify an actual numeric threshold and not one of the three pre-defined choices, that number is returned by this function. The value is an integer > 0. This function applies only to event attribute groups with `event filtering` and summarization turned on.

isSummaryEvent
Returns 0 if it is a single event from a data source, or 1 if the event is a flow control summary event. The displayed values are Event and Summary Event if you use the default attribute for the function. If you create the attribute manually, the displayed values are 0 and 1, unless you define the names as enumerations. This function applies only to event attribute groups with `event filtering` and summarization turned on.

occurrenceCount
The number of matching events that are represented by a flow control summary event, or 1 if it is a single event from a data source. (A flow control summary event includes the first event). This function applies only to event attribute groups with `event filtering` and summarization turned on.

summaryInterval
Returns the summary interval that is configured for the attribute group which generated the event, in seconds. This function applies only to event attribute groups with `event filtering` and summarization turned on.

Examples

Examples of the use of formula operators and functions to created derived and filtered attributes

Example 1 - Derived Attributes

If you have a data source that defines the following attribute type:

Name	String
`xBytes`	Numeric

Name	String
`yBytes`	Numeric
`Virtual_Size`	Numeric

You can define:

- An attribute `totalBytes` to be the sum of `xBytes` and `yBytes`. You enter the formula `xBytes + yBytes`.
- An attribute `yPercent` to be a percentage of the total bytes, which is `yBytes`, can be defined as `yBytes % (xBytes + yBytes)` or `yBytes % totalBytes`.

Example 2 - Derived Attributes

This formula returns the maximum of the recently collected values for the `Virtual_Size`attribute. The number of samples that are collected is the value of the configuration variable, *K4P_COLLECTIONS_PER_HISTORY_INTERVAL* (accessed through `getenv`), converted to a number (through `atoi`):

```
max(last(Virtual_Size,atoi(getenv("K4P_COLLECTIONS_PER_HISTORY_INTERVAL"))))
```

Example 3 - Derived Attributes

This formula returns the square-root of the sum of the squares of the `xBytes` and `yBytes` attribute values:

```
sqrt(xBytes * xBytes + yBtyes * yBytes)
```

Example 4 - Derived Attributes

This formula returns the average of the `xBytes` attribute from the 20 most recent samples of the attribute group. If fewer than 20 samples are collected since the agent was started, it returns the average of the `xBytes` attribute from all samples:

```
average(last(xBytes,20))
```

Example 5 - Filtered Attributes

You have a data source that returns:

```
Name    Type  Size  Used  Free
Memory  MEM      8     4     4
Disk1   DISK   300   200   100
Disk2   DISK   500   100   400
```

You are only interested in the disk usage. The solution is to create a filter to limit the data that is returned. To limit the returned data, you create a simple filter that returns a Boolean, true, or false value, as follows

Disk Filter:

```
Type=="DISK"
```

Now when the filter `Type=="DISK"` is true, the attribute group returns only disk usage data, for example:

```
Name    Type   Size  Used  Free
Disk1   DISK   300   200   100
Disk2   DISK   500   100   400
```

Example 6 - Filtered Attributes

You have a data source that returns:

```
Name   Size  Used  Free
Memory    8     4     4
Disk1   300   200   100
Disk2   500   100   400
```

The data that is returned is similar to the previous example, however, there is not a Type attribute present this time. Here you can use the matches function to find any data rows with a name attribute value that matches "Disk" followed by a number.

Disk Filter:

```
matches(Name, "Disk[0-9]*")
```

Now when the filter matches the string "Disk" followed by a number in attribute Name, only the disk usage data rows are returned:

```
Name   Size  Used  Free
Disk1   300   200   100
Disk2   500   100   400
```

Specifying operating systems

When you define data sources that are not available on all operating systems that the agent supports, you must specify the operating systems where the data source runs.

About this task

By default, the data source provides data on all of the operating systems that are defined at the agent level, as desrcibed in "Default operating systems" on page 1153. You can change the operating systems for each data source.

Procedure

1. To open the Operating Systems section, click **Operating Systems** in the **Data Source Information** page when you add a data source.
2. Select the operating systems on which the data source is to operate.

   Select individual operating systems, all operating systems, all operating systems of a specific type, or the agent default operating systems.

Configuring and Tuning data collection

When an Agent Builder agent is created, you can configure and tune its data collection to achieve the best results.

How you configure and tune your agent can be different for different Agent Builder agents and even between attribute groups in a single agent. Agent Builder agents can include two types of data and they support two basic methods of data collection for the most common type of data.

Data types

An agent collects two types of data:

1. Most Tivoli Monitoring attribute groups represent snapshots of data. Someone asks for the data and it is returned. Agents use this type of data to represent configuration, performance, status, and other information where a one time collection of a set of data makes sense. This data is called *sampled data*.
2. Some Tivoli Monitoring data represents events. In this case, an event happens and the agent must forward data to Tivoli Monitoring. Examples of events are SNMP Traps, Windows Event Log entries, and new records that are written to a log file. For simplicity, these types of data are grouped and referred to as *event data*.

Sampled data

When sampled data is required, a request is sent to the agent for a specific attribute group. The request might be initiated by clicking a workspace in the Tivoli Enterprise Portal. Other things that might initiate a request are a situation that is running, a data collection for the Warehouse, or a SOAP request. When the agent receives the request, the agent returns the current data for that attribute group. Tivoli Enterprise Portal requests target a specific attribute group in a particular Managed System Name (MSN). Situations and historical requests are more interesting, especially in an agent which includes subnodes. When a situation needs data for an attribute group in a subnode, the agent receives one request with a list of the targeted subnodes. The agent must respond with all the data for the requested attribute group for all of the subnodes before Tivoli Monitoring can work on the next request.

The most straightforward way for an agent to satisfy a request is to collect data every time it receives a request from Tivoli Monitoring. Agent Builder agents do not collect data every time. Data is not collected every time because it often takes time or uses resources to collect data. And in many cases the same data is requested many times in a short period. For example, a user might define several situations that run at the same interval on an attribute group and the situations can signal several different conditions. Each of these situations results in a request to the agent, but you might prefer each of the situations to see the same data. It is likely that as each situation sees the same data, more consistent results are obtained, minimizing the demand for system resources by the monitoring agent.

The agent developer can configure agents to optimize data collection by choosing to run the collection in one of the following two modes:

1. **On-demand collection**: The agent collects data when it receives a request and returns that data.
2. **Scheduled collection**: The agent runs data collection in the background on scheduled intervals and returns the most recently collected data when it receives a request.

The agent uses a short-term cache in both of these modes. If another request for data is received while the cache is valid, the agent returns data from the cache without collecting new data for each request. Using data from the cache solves the problem that is caused by multiple concurrent situations (and other types of) requests. The amount of time the data remains valid, the scheduled collection interval, the number of threads that are used for collection and whether the agent runs in on demand or scheduled mode are all defined by environment variables. Using the environment variables, you can tune each agent for the best operation in its environment.

See the following examples that illustrate how the agent works in both modes:

- Agent 1 (*on-demand* collection): A simple agent that collects a small amount of data that is normally accessed only by situations or on an infrequent basis in the Tivoli Enterprise Portal. Data collection is reasonably fast, but it can use up computing and networking resources. This agent is normally defined to run on demand. If no situations are running or no one clicks the Tivoli Enterprise Portal, the agent does nothing. When data is needed, it is collected and returned. The data is placed into the short-term cache so that further requests at about the same time return the same data. This type of collection is likely the most efficient way for this agent to run because it collects data only when someone actually needs it.
- Agent 2 (*scheduled* collection): A complex agent that includes subnodes and collects data from multiple copies of the monitored resource. Many copies of the resource can be managed by one agent. It is normal to run situations on the data on a relatively frequent basis to monitor the status and performance of the monitored resource. This agent is defined to run a *scheduled*collection. One reason for running a *scheduled* collection is the way that situations are evaluated by Tivoli Monitoring agents. Because situations are running on the attribute groups in the subnodes, the agent receives one request for the data from all of the subnodes simultaneously. The agent cannot respond to other requests until all of the data is returned for a situation. If the agent collected all of the data when the request arrived, the agent would freeze when you click one of its workspaces in theTivoli Enterprise Portal. To avoid freezing the agent, the agent builder automatically defines all subnode agents to run as scheduled collection. The agent developer tunes the number of threads and refresh interval to collect the data at a reasonable interval for the data type. For example, the refresh interval can be one time a minute, or one time every 5 minutes.

Environment variables

An agent determines which mode to use and how the scheduled data collection runs based on the values of a set of environment variables. These environment variables can be set in the definition of the agent on the **Environment Variables** panel. Each environment variable is listed in the menu along with the default values. The environment variables can also be set or modified for an installed agent by editing the agent's environment (env) file on Windows or initialization (`ini`) file on UNIX. The environment variables that control data collections for sampled attribute groups are:

- `CDP_DP_CACHE_TTL=<validity period for the cached data - default value 55 seconds>`
- `CDP_DP_THREAD_POOL_SIZE=<number of threads to use for concurrent collection - default value 15 for subnode agents>`
- `CDP_DP_REFRESH_INTERVAL=<number of seconds between collections - default value 60 seconds for subnode agents>`
- `CDP_DP_IMPATIENT_COLLECTOR_TIMEOUT=<amount of time to wait for new data after validity period expires - default value 5 seconds>`

The most important of these variables are `CDP_DP_CACHE_TTL, CDP_DP_REFRESH_INTERVAL`, and `CDP_DP_THREAD_POOL_SIZE`.

If `CDP_DP_THREAD_POOL_SIZE` has a value greater than or equal to 1 or the agent includes subnodes, the agent operates in *scheduled* collection mode. If `CDP_DP_THREAD_POOL_SIZE` is not set or is 0, the agent runs in *on-demand* collection mode.

If the agent is running in *scheduled* mode, then the agent automatically collects all attribute groups every `CDP_DP_REFRESH_INTERVAL` seconds. It uses a set of background threads to do the collection. The number of threads is set by using `CDP_DP_THREAD_POOL_SIZE`. The correct value for the `CDP_DP_THREAD_POOL_SIZE` varies based on what the agent is doing. For example:

- If the agent is collecting data from remote systems by using SNMP, it is best to have `CDP_DP_THREAD_POOL_SIZE` similar to the number of remote systems monitored. By setting the pool size similar to the number of monitored remote systems, the agent collects data in parallel, but limits the concurrent load on the remote systems. SNMP daemons tend to throw away requests when they get busy. Discarding requests forces the agent into a try-again mode and it ends up taking more time and more resources to collect the data.
- If the agent includes a number of attribute groups that take a long time to collect, use enough threads so that long data collections can run in parallel. You can probably add a few more for the rest of the attribute groups. Use threads in this way if the target resource can handle it. Examples of when attribute groups can take a long time to collect are if the script runs for a long time, or a JDBC query takes a long time.

Running an agent with a larger thread pool causes the agent to use more memory (primarily for the stack that is allocated for each thread). It does not however increase the processor usage of the process or increase the actual working set size of the process noticeably. The agent is more efficient with the correct thread pool size for the workload. The thread pool size can be tuned to provide the wanted behavior for a particular agent in a particular environment.

When data is collected, it is placed in the internal cache. This cache is used to satisfy further requests until new data is collected. The validity period for the cache is controlled by `CDP_DP_CACHE_TTL`. By default the validity period is set to 55 seconds. When an agent is running in scheduled mode, it is best to set the validity period to the same value as `CDP_DP_REFRESH_INTERVAL`. Set it slightly larger if data collection can take a long time. When set the validity period in this way, the data is considered valid until its next scheduled collection.

The final variable is `CDP_DP_IMPATIENT_COLLECTOR_TIMEOUT`. This variable comes into play only when `CDP_DP_CACHE_TTL` expires before new data is collected. When the cache expires before new data is collected, the agent schedules another collection for the data immediately. It then waits for this collection to complete up to `CDP_DP_IMPATIENT_COLLECTOR_TIMEOUT` seconds. If the new collection completes, the cache is updated and fresh data is returned. If the new collection does not complete, the existing data is returned. The agent does not clear the cache when `CDP_DP_CACHE_TTL` completes to prevent a problem that is seen with the Universal Agent. The Universal Agent always clears its data cache when the validity period ends. If the Universal Agent clears its data cache before the next collection completes, it has an empty cache for that attribute group and returns no data until the collection completes. Returning no data becomes a problem when situations are running. Any situation that runs after the cache cleared but before the next collection completes sees no data and any of the situations that fire are cleared. The result is floods of events that fire and clear just because data collection is a little slow. The Agent Builder agents do not cause this problem. If the 'old' data causes a situation to fire generally the same data leaves that situation in the same state. After the next collection completes, the situation gets the new data and it either fires or clears based on valid data.

Attribute groups

Agent Builder agents include two attribute groups that you can use to inspect the operation of data collection and to tune the agent for your environment. The attribute groups are Performance Object Status and Thread Pool Status. When these attribute groups are used to tune data collection performance, the most useful data is:

- `Performance Object Status`, `Average Collection Duration` attribute. This attribute shows you how long each attribute group is taking to collect data. Often a small percentage of the attribute groups in an agent represents most of the processor usage or time that is used by the agent. You might be able to optimize the collection for one or more of these attribute groups. Or you can modify the collection interval for one or more groups, if you do not need some data to be as up-to-date as other data. For more information, see ("Examples and advanced tuning" on page 1193).
- `Performance Object Status`, `Intervals Skipped` attribute. This attribute shows you how many times the agent tried to schedule a new collection for the attribute group and it found that the previous collection was still on the queue, waiting to be run, or already running. In a normally behaved agent this attribute value is zero for all attribute groups. If this number starts growing, you tune the data collection, by adding threads, lengthening the interval between collections, or optimizing the collection.
- `Thread Pool Status`, `Thread Pool Avg Active Threads` attribute. You can compare this value to the Thread Pool Size attribute group to see how well your thread pool is being used. Allocating a thread pool size of 100 threads when the average number of active threads is 5 is probably just wasting memory.
- `Thread Pool Status, Thread Pool Avg Job wait and Thread Pool Avg Queue Length attributes.` These attributes represent the time an average data collection spends waiting on the queue to be processed by a thread and the average number of collections on the queue. Because of the way this data is collected, even an idle system indicates that at least an average of one job is waiting on the queue. A larger number of waiting jobs or a large average wait time indicates that collections are being starved. You can consider adding threads, lengthening the interval between collections or optimizing the collection for one or more attribute groups.

Event data

Agent Builder agents can expose several types of event data. Some behavior is common for all event data. The agent receives each new event as a separate row of data. When a row of event data is received, it is sent immediately to Tivoli Monitoring for processing, and added to an internal cache in the agent. Situations and historical collection are performed by Tivoli Monitoring when each row is sent to Tivoli Monitoring. The cache is used to satisfy Tivoli Enterprise Portal or SOAP requests for the data. The agent can use the cache to perform duplicate detection, filtering, and summarization if defined for the attribute group. The size of the event cache for each attribute group is set by `CDP_PURE_EVENT_CACHE_SIZE`. This cache contains the most recent `CDP_PURE_EVENT_CACHE_SIZE` events with the most recent event returned first. There are separate caches for each event attribute group. When the cache for an attribute group fills, the oldest event is dropped from the list.

The Agent Builder agent can expose events for:

- Windows Event Log entries
- SNMP Traps or Informs
- Records added to log files

- JMX MBean notifications
- JMX monitors
- Events from a Java API provider or socket provider.
- Joined attribute groups (where one of the data sources is an event data source)

These events are handled in the most appropriate way for each of the sources. SNMP Traps and Informs, JMX notifications and events from the Java API and socket providers are received asynchronously and forwarded to Tivoli Monitoring immediately. There is no requirement tune these collectors. The agent subscribes to receive Windows Event Log entries from the operating system by using the Windows Event Log API. If the agent is using the older Event Logging API, it polls the system for new events by using the thread pool settings. For joined attribute groups where one of the data sources is an event data source, there is no tuning to apply to the joined attribute group. Though the joined attribute group does benefit from any tuning applied to the event source group.

File monitoring is more complicated. The agent must monitor the existence of the files and when new records are added to the files. The agent can be configured to monitor files by using patterns for the file name or a static name. As the set of files that matches the patterns can change over time, the agent checks for new or changed files every `KUMP_DP_FILE_SWITCH_CHECK_INTERVAL` seconds. This global environment variable governs all file monitoring in an agent instance. When the agent determines the appropriate files to monitor, it must determine when the files change. On Windows systems, the agent uses Operating System APIs to listen for these changes. The agent is informed when the files are updated and processes them immediately. On UNIX systems, the agent checks for file changes every `KUMP_DP_EVENT` seconds. This global environment variable governs all file monitoring in an agent instance. When the agent notices that a file changed, it processes all of the new data in the file and then waits for the next change.

Examples and advanced tuning

Example

Environment variables that are used for more advanced tuning are defined at the agent level. You set the following variables one time and they apply to the all of the attribute groups in the agent:

- `CDP_DP_CACHE_TTL`
- `CDP_DP_IMPATIENT_COLLECTOR_TIMEOUT`
- `KUMP_DP_FILE_SWITCH_CHECK_INTERVAL`
- `KUMP_DP_EVENT`

You can make the following variables apply to individual attribute groups. They still have a global setting that applies to all other attribute groups in the agent:

- `CDP_DP_REFRESH_INTERVAL`
- `CDP_PURE_EVENT_CACHE_SIZE`

If you defined an agent to include the following six attribute groups:

- EventDataOne
- EventDataTwo
- EventDataThree
- SampledDataOne
- SampledDataTwo
- SampledDataThree

You might set the following default variables:

- `CDP_DP_CACHE_TTL=55`
- `CDP_DP_IMPATIENT_COLLECTOR_TIMEOUT=2`
- `CDP_DP_REFRESH_INTERVAL=60`

- CDP_PURE_EVENT_CACHE_SIZE=100

As a result, all of the attribute groups which contain sampled data (SampledDataOne, SampledDataTwo, and SampledDataThree) would be collected every 60 seconds. Each of the event attribute groups (EventDataOne, EventDataTwo, and EventDataThree) would store the last 100 events in their cache.

These settings might work perfectly, or there might be reasons that you must control the settings at a more granular level. For example, what if EventDataOne generally receives 10 times as many events as EventDataTwo and EventDataThree? To further complicate things, there really is a link between EventDataOne and EventDataTwo. When one event is received for EventDataTwo, there are always multiple events for EventDataOne and users want to correlate these events. There is not a single correct setting for the cache size. It would be nice to be able to have EventDataOne store a larger number of events and EventDataTwo store a smaller number. You can achieve this storage by setting CDP_PURE_EVENT_CACHE_SIZE to the size that makes sense for most of the event attribute groups, 100 seems good. Then, you can set CDP_EVENTDATAONE_PURE_EVENT_CACHE_SIZE to 1000. That way all of the corresponding events are visible in the Tivoli Enterprise Portal.

The same thing can be done with CDP_DP_REFRESH_INTERVAL. Set a default value that works for the largest number of attribute groups in the agent. Then set CDP_*attribute group name*_REFRESH_INTERVAL for the attribute groups which must be collected differently. To optimize collection, set the default CDP_DP_REFRESH_INTERVAL to match the CDP_DP_CACHE_TTL value. CDP_DP_CACHE_TTL is a global value so if set to a value less than a refresh interval, unexpected collections might occur.

Defining and testing data sources

Agent Builder supports a number of data providers. You can create data sources from each data provider. The procedure for creating and testing data sources is different for each data provider.

For most data providers, when you create a data source, a data set (attribute group) is added to the agent. The data set contains the information that is gathered by this data source.

A data source with a Process, Windows service, or Program return code data provider uses the special Availability data set. Only one Availability data set can be created in an agent. It contains the information that is gathered by all data sources with a Process, Windows Service, or Program Return Code data provider in this agent.

All Windows log data sources in an agent or subnode place event information into one Event Log data set.

Setting up a data source for Cloud APM

In Cloud APM, you can use data from all data sets in the Details dashboard and to set up thresholds using the threshold manager. If you want to use information from a data set in the summary dashboard for the agent or subnode, including the status indicator, as well as for resource information (service name, address, and port), the data set must produce only one row.

For most data providers, you can select **Produces a single data row** in the data set configuration. If the gathered information would include more than one row, you can click **Advanced** to set up a filter that ensures the correct row is produced (for instructions, see "Filtering attribute groups" on page 1178). You can test your data source to ensure that the gathered information produces the row that you need.

For some data providers, the data set must produce multiple rows. Also, the process, Windows service, and command return code data sources place data into a single Availability data set, which produces multiple rows. In such cases, you must create a filtered data set that produces one row. For instructions about creating a filtered data set (attribute group), see "Creating a filtered attribute group" on page 1311.

Some other data providers produce event data; a row is included for every new event. Do not use these data providers for summary or resource information in Cloud APM.

The following data providers must produce a data set with multiple rows:

- Process (uses the Availability data set)
- Windows service (uses the Availability data set)

- Program return code (uses the Availability data set)
- For some data types, SNMP and JMX
- Depending on the application, Socket and Java API

The following data providers produce event data:

- SNMP event
- Log file
- AIX binary log
- Windows event log
- Depending on the application, Socket and Java API

One of the attributes of the data set must provide a status value. Cloud APM uses this value for the overall status indicator. If the row does not include an attribute that can be used as a status indicator, you can create a derived attribute to calculate the status. You must configure the status severity values; for instructions, see "Specifying severity for an attribute used as a status indicator" on page 1177.

Monitoring a process

You can define a data source that monitors a process or several processes which run on a server. The processes must run on the same host as the agent. For every process, the data source adds a row to the Availability data set.

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **A process** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **A process**.
3. Click **Next**.
4. On the **Process Monitor** page, in the **Process information** area, provide the display name and process name. You can type the process name manually or obtain it by clicking **Browse**. Clicking **Browse** shows a list of processes that are currently running on the local system or on a remote system.

   You can further discriminate processes by selecting the **Use argument match** and **Match full command line** options. For example, if multiple instances of the same processes are running on the system, one instance can be distinguished from another by using these options.

*Table 268. Fields on the **Process Monitor** page.* A table listing the fields in the **Process Monitor** page and their descriptions

Field name	Description	Acceptable values
Display name	Descriptive name for the component of the application that is implemented by the process as it is shown in the Tivoli Enterprise Portal or in the IBM Cloud Application Performance Management console	Descriptive string
Process name	Name of the process that is being monitored	Valid executable file name
Use argument match	Select if you want to match on the process arguments.	On or Off

Table 268. Fields on the ***Process Monitor*** *page.* A table listing the fields in the **Process Monitor** page and their descriptions *(continued)*

Field name	Description	Acceptable values
Argument	Argument string on which to match. Argument matching looks for the provided string as a substring of the arguments. Matching is successful if you provide any part of the arguments as the input string.	String
Match full command line	Specify the entire name of the executable file that might include the path	On or Off
Command line	Matches the provided string against the fully qualified command name that is used to start the process. Command arguments are not included. Fully qualified means the path to the command must be included.	String
Operating systems	Select the operating systems on which this process runs	Any selection

5. If you click **Browse**, the **Process Browser** window opens. This window initially contains detailed information about each process on the Agent Builder system. The information includes the ID, the process name, and the full command line for the process. Select one or more processes or work with the list in the **Process Browser** window by using one or more of the following actions:

   a) To sort the list of processes, click the column heading.

   b) To refresh the information in the window, click the **Refresh** (lightening bolt) icon.

   c) To search for specific processes, click the **Search** (binoculars) icon.

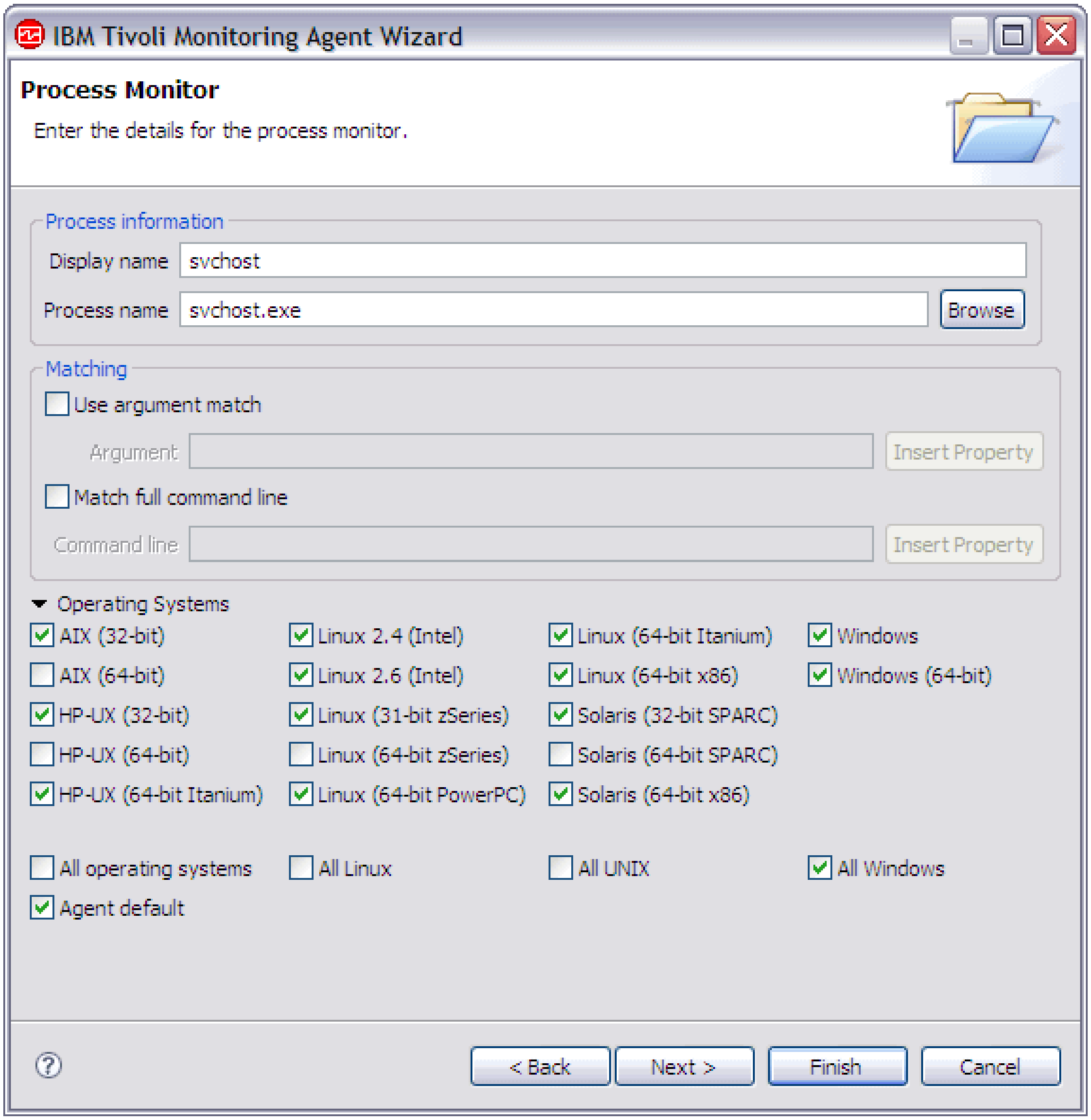
   You can enter a search phrase and select options section to search by process identifier, name, and command line.

   d) To view processes on a different system, select a previously defined system from the **Connection Name** list. Or click **Add** to enter the system information for a new system.

   For more information, see "Defining connections for process browsing" on page 1198. You can load processes from more than one system at a time, and switch between connections while processes are loading for one or more connections.

   Note: When you browse remote systems, the command-line details are available only when you browse through a Tivoli Enterprise Portal Server.

   In the following example, after you select `svchost.exe`, it is shown in the **Process name** field on the **Process Monitor** page (Figure 31 on page 1197).



IBM Tivoli Monitoring Agent Wizard

Process Monitor

Enter the details for the process monitor.

Process information

Display name: svchost

Process name: svchost.exe

Matching

□ Use argument match

Argument:

□ Match full command line

Command line:

Operating Systems

☑ AIX (32-bit)
□ AIX (64-bit)
☑ HP-UX (32-bit)
□ HP-UX (64-bit)
☑ HP-UX (64-bit Itanium)
☑ Linux 2.4 (Intel)
☑ Linux 2.6 (Intel)
☑ Linux (31-bit zSeries)
□ Linux (64-bit zSeries)
☑ Linux (64-bit PowerPC)
☑ Linux (64-bit Itanium)
☑ Linux (64-bit x86)
☑ Solaris (32-bit SPARC)
□ Solaris (64-bit SPARC)
☑ Solaris (64-bit x86)
☑ Windows
☑ Windows (64-bit)

□ All operating systems
□ All Linux
□ All UNIX
☑ All Windows
☑ Agent default

Figure 31. Process Monitor page example

6. Complete the **Process Monitor** page by using the information in (Table 268 on page 1195).

   Note: If the process you described in this monitor is applicable to only some of the operating systems that your application runs on, you might want to create one or more process monitors with the same display name to cover the other operating systems. Add the process monitors one at a time. Ensure that the display name is the same for each monitor, but that the process name can be found on the operating systems that are selected.

7. Do one of the following steps:

   - If you are using the **Agent** wizard, click **Next**.
   - Click **Finish** to save the data source and open the Agent Editor.

What to do next

If you want to use the data from this data source in the summary dashboard for IBM Cloud Application Performance Management, you must create a filtered data set (attribute group) based on the Availability data set and configure it as providing a single row. Use the NAME field to select the row for your process.

You can use the Status field for status; DOWN means that the process is not running, while UP means it is running. In the new filtered attribute group, select the Status field and specify the severity values for it.

If several copies of the process are running, several rows with this process name are present in the Availability data set, and all of then include the UP status. Your filtered data set must be configured to return one row, so any of these rows might be returned, but the Status value is valid in any case.

For instructions, see:

- "Creating a filtered attribute group" on page 1311
- "Specifying severity for an attribute used as a status indicator" on page 1177
- "Preparing the agent for Cloud APM" on page 1342

Defining connections for process browsing

When you define a process data source, you can view and select processes from other systems. However, when the agent runs, it monitors processes that run on the same system as the agent.

About this task

You must have credentials for the other systems or they must be monitored by a Tivoli Monitoring operating system agent.

Procedure

1. To define a connection, click **Add** in the **Process Browser** window.

   You can select either a connection type (Secure Shell (SSH), Windows, or Tivoli Enterprise Portal Server Managed System) or select an existing connection to use as a template.

   To add a Managed System connection, you require a Tivoli Enterprise Server host name, Tivoli Monitoring user name, and password. You also require the managed system name of the remote connection. When a managed system is selected, the table lists the process on the remote system.

   Note: The OS agent must be running on the system you are attempting to browse. The agent must also be connected to a running Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server.

   To add Secure Shell (SSH) or Windows connections, you require a host name, user name, and password.
2. When you add a connection, you can select the connection from the **Connection Name** list in the **Process Browser** window.

   If all the fields required to make the connection are not saved (for example, the password), the **Connection Properties** window for that connection opens. Enter the missing information. For Tivoli Enterprise Portal Server Managed System connections, you must connect to the Tivoli Enterprise Portal Server before you can enter a managed system.
3. Enter your user name and password, and then click the **Refresh** (lightening bolt) icon to connect before you select the managed system.

What to do next

To delete a connection, select the connection and click **Edit** to open the **Connection Properties** window. Select the **Remove this connection** check box and click **OK**.

Monitoring a Windows service

You can define a data source that monitors a service or several services which run on a Windows system. The services must run on the same host as the agent. For every service, the data source adds a row to the Availability data set.

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **A process** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **A Windows service**.
3. Click **Next**.

4. On the **Service Monitor** page, in the **Display name** field, type a description. In the **Service name** field, provide the name of the service application. You can type it manually or click **Browse** to view a list of services that are currently running on the local system or on a remote system.

   If you click **Browse**, the **Service Browser** window opens. This window initially contains detailed information about each service on the Agent Builder system. The information includes the service name, the display name, the state, and the description for the service.

   Note: Local services are not shown when Agent Builder is not running on a Windows system. A remote Windows system must be defined or selected, see ("Defining connections for service browsing" on page 1199).

   Note: The service description is not available when you are browsing through the Tivoli Enterprise Portal Server or from a UNIX or Linux system.

5. Select one or more services or do one or more of the following steps to work with the list in the **Service Browser** window:

   - To sort the list of services, click the column heading.
   - To refresh the information in the window, click the **Refresh** (lightening bolt) icon.
   - To search for a service, click the **Search** (binoculars) icon to open the **Service Search** window. You can search by the service name, display name, and description.
   - To view services on a different system, select a previously defined system from the **Connection Name** list or click **Add** to enter the system information. For more information, see ("Defining connections for service browsing" on page 1199). You can load services from more than one system at a time, and switch between connections while services are loading for one or more connections.

6. After selecting or entering the name of the service, complete one of the following steps:

   - If you are using the **Agent** wizard, click **Next**.
   - Click **Finish** to save the data source and open the Agent Editor.

What to do next

If you want to use the data from this data source in the summary dashboard for IBM Cloud Application Performance Management, you must create a filtered data set (attribute group) based on the Availability data set and configure it as providing a single row. Use the NAME field to select the row for your process.

In the new filtered attribute group, select the Functionality_Test_Status field and specify the severity values for it.

For instructions, see:

- "Creating a filtered attribute group" on page 1311
- "Specifying severity for an attribute used as a status indicator" on page 1177
- "Preparing the agent for Cloud APM" on page 1342

Defining connections for service browsing

In addition to selecting services from the system where Agent Builder is running, you can select services from other Windows systems.

About this task

To select services from other Windows systems, you define a connection to the remote system. You must have credentials for the systems or they must be monitored by a Tivoli Monitoring operating system agent.

Procedure

1. To define a connection, click **Add** in the **Service Browser** window.

The **Select Connection Type** window opens. To add a Managed System connection, you require a Tivoli Enterprise Server host name, Tivoli Monitoring user name and password, and the managed system name. When a managed system is selected, the table lists the service on the remote system.

Note: The OS agent must be running on the system you are attempting to browse and also connected to a running Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server.

You require a host name, user name, and password to add a Windows connection.

2. Select a connection type (Windows, or Tivoli Enterprise Portal Server Managed System) or select an existing connection to use as a template.

   The **Connection Properties** window opens.
3. Complete the Connection Properties.
4. Click **Finish**
5. When you add a connection, you can select the connection from the **Connection Name** list in the **Service Browser** window.

   If the fields necessary to make the connection are not saved (for example, the password), the**Connection Properties** window opens and you can enter the missing information.

   a) For Tivoli Enterprise Portal Server Managed System connections, you must connect to the Tivoli Enterprise Portal Server before you can enter a managed system. Enter your user name and password, and then click the **Refresh** (lightening bolt) icon to connect before you select the managed system.
6. To delete a connection, follow these steps:

   a) Select the connection in the **Service Browser** window.

   b) Click **Edit** to open the **Connection Properties** window.

   c) Select the **Remove this connection** check box.

   d) Click **OK**.

Monitoring data from Windows Management Instrumentation (WMI)

You can define a data source to collect data from Windows Management Instrumentation (WMI) on the system where the agent runs or on a remote system. A data source monitors a single WMI class and places all values from this class into the data set that it produces. If the class provides several instances, the data set has multiple rows; you can filter by instance name to ensure the data set has one row.

Before you begin

If your agent collects data from a remote system by using Windows Management Instrumentation (WMI), it requires permissions to access WMI data on the remote system. The agent can access WMI data on a remote system when you provide credentials of an account with permissions to access WMI data on the system. The Administrator account has the required permissions. In the procedure that follows you can either provide the Administrator credentials or the credentials of another user with the required permissions. For more information about creating a user account with permissions to browse WMI data, see "Creating a user with Windows Management Instrumentation (WMI) permissions" on page 1334.

To collect metrics through the Windows APIs, the agent must be hosted on a Windows operating system. Remote registry administration must be enabled on the remote systems.

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Data from a server** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **WMI**.
3. Click **Next**.
4. On the **Windows Management Instrumentation (WMI) Information** page, complete one of the following steps:

- Type a name for the WMI namespace and a name for the WMI class name in the fields. Then go to step "9" on page 1201
- Click **Browse** to see all of the WMI classes on the system.

  To browse a remote system, select a system from the list (if one is defined). Alternatively click **Add** to add the host name of a Windows system. Provide the credentials of a user account with permissions to access WMI data on the remote system, or provide Administrator credentials for the remote system. The page is updated with the information for the remote system. Browsing is available only when the Agent Builder is run on a Windows system, and can browse only Windows systems.

5. Click the plus sign (+) next to a class to expand the class and show the attributes.
6. From the list, select the class with its associated attributes that you want to specify, and click **OK**.

   Note: You can click the **Search** (binoculars) icon to find your selection in the list. Type a phrase in the **Search phrase** field; specify your preference by clicking either the **Search by name**, **Search by class description**, or **Search by class properties** fields; and click **OK**. If you find the item for which you are searching, select it and click **OK**.

   The **WMI Information** page of the wizard opens again, showing the selected WMI class information.
7. Optional: You can test this attribute group by clicking **Test**. For more information about testing, see "Testing WMI attribute groups" on page 1202
8. Optional: You can create a filter to limit the data that is returned by this attribute group by clicking **Advanced**. For more information about filtering data from an attribute group, see "Filtering attribute groups" on page 1178
9. Click **Next**.

   Note: If you typed the WMI Namespace and WMI class name manually you are brought to the **Attribute Information** page, where you can complete attribute information. On the **Attribute Information** page, you can select **Add additional attributes** if you want to add more attributes. Click **Finish** to complete.
10. On the **Select key attributes** page, select key attributes or indicate that this data source produces only one data row. For more information, see ("Selecting key attributes" on page 1151).
11. Do one of the following steps:
    - If you are using the **Agent** wizard, click **Next**.
    - Click **Finish** to save the data source and open the Agent Editor.
12. You can add attributes and supply the information for them. For more information, see "Creating attributes" on page 1169.

    In addition to fields that are applicable to all data sources (Table 264 on page 1173), the **Attribute Information** page for the WMI data source has the following field:

    Metric name

    Property name from the class you want to collect
13. If you want to set global options for the data source, click **Global Options**.

    Select the **Include remote Windows configuration properties** check box if you want to include this option, and click **OK**.

    For information about Windows remote connection configuration for Windows data sources, see ("Configuring a Windows remote connection" on page 1333).

Testing WMI attribute groups

If you are running Agent Builder on a Windows system, you can test a WMI attribute group within Agent Builder.

Procedure

1. You can start the Testing procedure in the following ways:
   - During agent creation click **Test** on the **WMI Information** page.
   - After agent creation, select an attribute group on the Agent Editor **Data Source Definition** page and click **Test**. For more information about the Agent Editor, see "Using the Agent Editor to modify the agent" on page 1152.

   After you click **Test** in one of the previous two steps, the **WMI Test** window is displayed.
2. Optional: Before you start testing, you can set environment variables and configuration properties. For more information, see "Attribute group testing" on page 1347).
3. Click **Start Agent**.

   A window indicates that the Agent is starting.
4. To simulate a monitoring environment request for agent data, click **Collect Data**.

   The agent queries WMI for data. The **WMI Test** window collects and displays any data in the agent's cache since it was last started.
5. Optional: Click **Check Results** if the returned data is not as you expected.

   The **Data Collection Status** window opens and shows you more information about the data. The data that is collected and displayed by the **Data Collection Status** window is described in ("Performance Object Status node" on page 1388).
6. Stop the agent by clicking **Stop Agent**.
7. Click **OK** or **Cancel** to exit the **WMI Test** window. Clicking **OK** saves any changes that you made.

Related concepts

"Testing your agent in Agent Builder" on page 1346
After you use Agent Builder to create an agent, you can test the agent in Agent Builder.

Monitoring a Windows Performance Monitor (Perfmon)

You can define a data source to collect data from Windows Performance Monitor (Perfmon). A data source monitors a Perfmon object. The counters in the object are placed in attributes in the resulting data set. If the class provides several instances, the data set has multiple rows; you can filter by instance name to ensure the data set has one row.

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Data from a server** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **Perfmon**.
3. Click **Next**.
4. On the **Perfmon Information** page, complete one of the following steps:
   - Type the name of the object in the **Object Name** field, and click **Next** to define the first attribute in the attribute group.

     Note: If you type the name for the Windows Performance Monitor object, it must be the English name.
   - Click **Browse** to view the list of Perfmon objects.

     When the Performance Monitor (Perfmon) Object Browser window initially opens, the window populates with the information from the local system. To browse a remote system, select a system from the list (if one is defined), or click **Add** to add the host name of a Windows system. Provide an Administrator ID and password. The window updates with the information for the remote system. Browsing is available only when Agent Builder is running on a Windows system, and can browse only Windows systems. For example, you cannot add the host name of a Linux or Solaris system to do a remote browse.

  - When you click an object name, the available counters in that object are shown in the window.
    - To sort the Windows Performance Monitor objects or counters, click the column heading.
    - To refresh the information in the window, click **Refresh**.
    - To search for specific objects or counters click the **Search** (binoculars) icon to open the **Performance Monitor Search** window. You can search object names, counter names, or both. The search operation does a substring match and is not case-sensitive.
    - Select an object and click **OK**.
    - The **Perfmon Information** page opens with the name of the selected object in the **Object Name** field.

- If you want to set global options for the data source, click **Global Options**

  Select the **Include remote Windows configuration properties** check box if you want to include this option, and click **OK**.

  For information about Windows remote connection configuration for Windows data sources, see (“Configuring a Windows remote connection” on page 1333).

5. If the Windows Performance Monitor object selected returns multiple instances and you want to filter the results that are based on the instance name:

   a) Select the **Filter by Perfmon Instance Name** check box on the **Perfmon Information** page.

   b) In the **Perfmon Instance Name** field, type the name of the instance to be filtered, or click **Browse** to list the instances available.

   c) To browse a remote system, either select one from the list, or click **Add** to add the host name of a Windows system. After you select a host, provide an Administrator ID and password. The table is updated with the list of instances on the remote system.

   Note: You can also filter by attribute group, see step “9” on page 1203

6. If the selected Windows Performance Monitor Object is to return multiple instances, and you want the instance name to be returned, select **Return Instance Name** on the **Perfmon Information** page.

   Checking this option adds an attribute to the data source that is not shown in the list of attributes. This attribute contains the instance name.

   Note: If you browsed for the selected object, and that object is defined as having multiple instances, this check box is selected automatically.

7. If you did not check the option to return the instance name, the Select key attributes page opens. On the Select key attributes page, select key attributes or indicate that this data source produces only one data row. For more information, see (“Selecting key attributes” on page 1151).

8. Optional: You can test this attribute group by clicking **Test**. For more information about testing, see “Testing Perfmon attribute groups” on page 1204

9. Optional: You can create a filter to limit the data that is returned by this attribute group by clicking **Advanced**.

   For more information about filtering data from an attribute group, see step “Filtering attribute groups” on page 1178

   Note: You can also filter by instance name, see “5” on page 1203

10. Do one of the following steps:

    - If you are using the New Agent wizard, click **Next**.
    - Click **Finish** to save the data source and open the Agent Editor.

      The **Agent Editor Data Source Definition** page shows a list that contains the object and information about the object.

11. You can add attributes and supply the information for them. For more information, see ("Creating attributes" on page 1169).

    In addition to the fields applicable to all data sources, the **Perfmon Attribute Information** page for the data source has the following field:

    Metric name
    Name of the counter for the specific object.

What to do next

For information about Windows remote connection configuration for Perfmon data sources, see "Configuring a Windows remote connection" on page 1333.

Testing Perfmon attribute groups

If you are running Agent Builder on a Windows system, you can test the Perfmon attribute group that you created.

Procedure

1. You can start the Testing procedure in the following ways:
   - During agent creation click **Test** on the **Perfmon Information** page.
   - After agent creation, select an attribute group on the Agent Editor **Data Source Definition** page and click **Test**. For more information about the Agent Editor, see "Using the Agent Editor to modify the agent" on page 1152.

   After you click **Test** in one of the previous two steps, the **Perfmon Test** window is displayed.
2. Optional: Before you start testing, you can set environment variables and configuration properties. For more information, see "Attribute group testing" on page 1347.
3. Click **Start Agent**. A window indicates that the Agent is starting.
4. To simulate a request from the monitoring environment for agent data, click **Collect Data**.

   The agent queries Performance Monitor for data. The **Perfmon Test** window collects and shows any data in the agent's cache since it was last started.

   Note: You might not see useful data for all attributes until you click **Collect Data** a second time. The reason is that some Performance Monitor attributes return delta values, and a previous value is required to calculate a delta value.
5. Optional: Click **Check Results** if the returned data is not as you expected.

   The **Data Collection Status** window opens and shows you more information about the data. The data that is collected and shown by the **Data Collection Status** window is described in "Performance Object Status node" on page 1388.
6. Stop the agent by clicking **Stop Agent**.
7. Click **OK** or **Cancel** to exit the **Perfmon Test** window. Clicking **OK** saves any changes that you made.

Related concepts

"Testing your agent in Agent Builder" on page 1346
After you use Agent Builder to create an agent, you can test the agent in Agent Builder.

Monitoring data from a Simple Network Management Protocol (SNMP) server

You can define a data source to monitor an SNMP server. A data source monitors all data from a single SNMP object identifier (OID) and a single host. if you select an element of the OID registration tree under which other objects are registered, a data set is created for each distinct set of scalar or table values. If an object returns scalar data, the data set has a single row. If an object returns tabular data, the data set has multiple rows.

About this task

Simple Network Management Protocol V1, V2C (note that the version is V2C and not just V2), and V3 are supported by agents.

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Data from a server** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **SNMP**.
3. Click **Next**.
4. On the Simple Network Management Protocol (SNMP) Information page, type the display name or click **Browse** to see all of the objects on the system.

   After you define the data source, you can add an attribute. The OIDs for these attributes can be long and difficult to type correctly. Using the Browse option is an easy way to input the correct OID.

   Note: The browser does not browse the live system, it reads definitions, Management Information Bases (MIBs).

   Note: Clicking the **Refresh** icon clears the in-memory version of the parsed MIB files and reparses the files in the workspace cache. The cache is in the following location: `*workspace_directory*\.metadata\.plugins\ com.ibm.tivoli.monitoring.agentkit\mibs`

   Where:

   workspace_directory
   Identifies the workspace directory that you specified when you initially ran the Agent Builder, see ("Starting Agent Builder" on page 1147).

   a) If the MIB that defines the wanted object is not loaded, click **Manage Custom MIBs** to open the Manage Custom MIBs dialog.

   b) Click **Add** to browse to the MIB file to add. To delete a MIB from the cache, select it and click **Remove**.

   c) Click **OK** to update the cache.

   If there are any errors when the MIBs are parsed, the Manage Custom MIBs dialog remains open. This dialog gives you the opportunity to add or remove MIBs to eliminate the errors.

   Clicking **Cancel** returns the MIB cache to the state it was in when the dialog was opened.

   Agent Builder includes a set of MIBs:

   - `hostmib.mib`
   - `rfc1213.mib`
   - `rfc1243.mib`
   - `rfc1253.mib`
   - `rfc1271.mib`
   - `rfc1286.mib`
   - `rfc1289.mib`
   - `rfc1315.mib`
   - `rfc1316.mib`
   - `rfc1381.mib`
   - `rfc1382.mib`
   - `rfc1443.mib`
   - `rfc1461.mib`
   - `rfc1471.mib`
   - `rfc1493.mib`

- `rfc1512.mib`
- `rfc1513.mib`
- `rfc1516.mib`
- `rfc1525.mib`
- `rfc1573a.mib`
- `rfc1595.mib`
- `rfc1650.mib`
- `rfc1657.mib`
- `rfc1659.mib`
- `rfc1666.mib`
- `rfc1695.mib`
- `rfc1747.mib`
- `rfc1748.mib`
- `rfc1757.mib`
- `rfc1903.mib`
- `rfc1907.mib`
- `rfc2011.mib`
- `rfc2021.mib`
- `rfc2024.mib`
- `rfc2051.mib`
- `rfc2127.mib`
- `rfc2128.mib`
- `rfc2155.mib`
- `rfc2206.mib`
- `rfc2213.mib`
- `rfc2232.mib`
- `rfc2233.mib`
- `rfc2238.mib`
- `rfc2239.mib`
- `rfc2320.mib`
- `rfc3411.mib`

All of these MIBs are standard, IETF defined MIBs. The MIBs are included because they represent common definitions that can be useful in monitoring. Also, many of the MIBs are necessary so that custom MIBs can resolve the symbols that they import.

d) Select an object from the list.

Click the plus sign (+) next to an object to expand and show the levels.

e) From the list, select the object that you want to specify and click **OK**.

The new data source is then listed on the **Data Source Definition** page.

Note: If you select an object that defines other objects (objects that are nested underneath the first object), all of these objects are turned into data sources. If you select a high-level object, many data sources are added.

5. On the **Simple Network Management Protocol Information** page, select the operating systems.
6. Optional: You can test the data source or sources by clicking **Test** on the **Simple Network Management Protocol Information** page.

For more information about testing, see “Testing SNMP attribute groups” on page 1208

7. Optional: You can create a filter to limit the data that is returned by this attribute group by clicking **Advanced**. For more information about filtering data from an attribute group, see “Filtering attribute groups” on page 1178
8. Click **Next**.
9. On the **Attribute Information** page, specify the information for the attribute.
10. Do one of the following steps:
    - If you are using the New Agent wizard, click **Next**.
    - Click **Finish** to save the data source and open the Agent Editor.
11. For more information about adding attributes and supplying the information for them, see “Creating attributes” on page 1169.

    In addition to fields that are applicable to all data sources, the **Attribute Information** page for the SNMP data source has the following fields:

    Metric name
    Arbitrary string

    Object identifier
    Full OID that is registered to the object, not including index values

What to do next

You can use the runtime configuration of the agent to set the monitored host.

To enable Agent Builder to generate 64-bit data types and to handle the maximum value for 32-bit unsigned MIB properties, see “SNMP MIB Parsing options” on page 1207.

SNMP MIB errors

Dealing with errors in SNMP MIBs.

It is not unusual to find errors when you are adding SNMP MIBs. Click **Details>>** in the **Agent Builder Error** window to see what the MIB error is.

One of the most common errors is missing definitions that are defined in other MIBs. You can import several MIBs simultaneously to resolve this problem, or you can incrementally add MIBs until all of the missing definitions are resolved. Agent Builder can use any definitions that are resolved. So you can choose to ignore an error that affects only that part of the MIB that you do not plan to use. The order of the MIBs does not matter because they are all loaded, and then the references are resolved.

SNMP MIB Parsing options

Set your preferences for SNMP MIB parsing

Procedure

1. In the Agent Builder, select **Window** > **Preferences** to open the **Preferences** window.
2. In the navigation pane, expand **IBM Agent Builder**.
3. Click **MIB Parsing** to open the **MIB Parsing** window.

   The MIB parser that is used by Agent Builder uses the grammar that is defined by ASN.1 to parse the MIBs. Some MIBs do not follow the grammar correctly. The parser can relax certain rules to accommodate the most common errors. By relaxing these rules, you can parse non-conforming MIBs.

   Allow types to start with lowercase letters
   Allows types that people write in MIBs, such as values

   Allow numeric named numbers
   Allows numbers that start with uppercase letters

   Allow underscore in value name
   Allows underscore characters

Allow values to begin with uppercase letters
Allows values that start with uppercase letters

Ignore duplicate MIBs
Turns off warning for duplicate MIB modules

4. Optional: Selecting the **Create 64-bit attributes for 32 bit unsigned MIB properties** check box, enables the Agent Builder to generate 64-bit data types to handle the maximum value for 32-bit unsigned MIB properties. Selecting this option does not change any existing agent field definitions. You must browse to the MIB file to create new data sources for these properties.
5. When you are finished editing the preferences, click **OK**.

Testing SNMP attribute groups

You can test the SNMP attribute group that you created within Agent Builder.

Procedure

1. You can start the Testing procedure in the following ways:

   - During agent creation click **Test** on the **Simple Network Management Protocol Information** page.

     Note:

     If the SNMP object selected contains more than one attribute group, you are prompted to select the attribute group to test.
   - After agent creation, select an attribute group on the **Agent Editor Data Source Definition** page and click **Test**. For more information about the Agent Editor, see "Using the Agent Editor to modify the agent" on page 1152

   After you click **Test** in one of the previous two steps, the SNMP Test settings window opens.
2. Select an existing connection from **Connection name** or click **Add** and you are prompted to select a connection type. Alternatively select an existing connection to use as a template, by using the **Create Connection Wizard**
3. After you select a connection type or an existing connection, click **Next** to complete the SNMP connection properties. When complete click **Finish** to return to the SNMP Test settings window.
4. Optional: Before you start testing, you can set environment variables and configuration properties. For more information, see ("Attribute group testing" on page 1347).
5. Click **Start Agent**. A window indicates that the Agent is starting.
6. To simulate a request from Tivoli Enterprise Portal or SOAP for agent data, click **Collect Data**. The agent queries the configured SNMP connection for data.
7. The **Test Settings** window collects and shows any data in the agent's cache since it was last started.
8. Optional: Click **Check Results** if the returned data is not as you expected.

   The **Data Collection Status** window opens and shows you more information about the data. The data that is collected and shown by the **Data Collection Status** window is described in "Performance Object Status node" on page 1388
9. Stop the agent by clicking **Stop Agent**.
10. Click **OK** or **Cancel** to exit the **Test Settings** window. Clicking **OK** saves any changes that you made.

Related concepts
"Testing your agent in Agent Builder" on page 1346

After you use Agent Builder to create an agent, you can test the agent in Agent Builder.

Monitoring events from Simple Network Management Protocol event senders

You can define a data source to collect data from SNMP Trap and Inform events. You must set the port in the agent runtime configuration and configure the servers to send event to the agent host on this port. All the monitored events are placed as rows in a data set.

About this task

Simple Network Management Protocol (SNMP) V1, V2C (note that this version name is V2C and not just V2), and V3 are supported by agents. SNMP Traps and Informs can be received and processed by the agent. Data that is received by this provider is passed to the monitoring environment as events.

For more information about the attribute groups for SNMP events, see ("SNMP Event attribute groups" on page 1414).

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Data from a server** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **SNMP Events**.
3. Click **Next**.
4. In the **Simple Network Management Protocol Event Information** window, do one of the following steps:

   - Click **All Events** to create an attribute group that sends an event for any received SNMP event.
   - Click **Generic Events** to create an attribute group that sends an event for any received generic SNMP event that matches any of the selected generic event types.
   - Click **Custom Events** to create one or more attribute groups that send events for enterprise-specific SNMP events. Click **Browse** to choose the events to be monitored.

     In the **Simple Network Management Protocol (SNMP) Management Information Base (MIB) Browser** window, the events in the selection pane are organized by the MIB module in which they were defined. Expand an SNMP object to show the events in that MIB module. In the list, click the object that you want to specify and click **OK**.

     Select the **Include attributes that show information defined in the trap configuration file** check box if you have a trap configuration file that contains static data for your traps. For more information about the SNMP trap configuration file, see ("SNMP trap configuration" on page 1466).

     Select the **Include variable binding (VarBind) data attribute** check box if you want to include an attribute with all of the variable binding (VarBind) data that is received in the trap protocol data unit (PDU). For more information about this attribute, see the attribute definition ("SNMP Event attribute groups" on page 1414).

     Note:

     a. The browser does not browse the live system; it reads definitions and, Management Information Bases (MIBs). The list of MIBs included with Agent Builder is defined in "Monitoring data from a Simple Network Management Protocol (SNMP) server" on page 1204. MIBs loaded by either SNMP data provider are available in both.
     b. If you select a MIB module or individual events, all the events in that module are converted to separate data sources. One attribute is added for each of the variables that are defined in the event. If you want all the events for the selected modules or traps to arrive in a single event source, select the **Collect events in a single attribute group** check box. If you select individual traps and the **Collect events in a single attribute group** flag is selected, one attribute is added for each of the variables that are defined in each of the events (duplicate variables are ignored). If you select a module, variable attributes are not added.

c. If you want to type your own filter, use the following syntax:

The value of the OID (object identifier) element is used to determine which traps to process for this attribute group.

- **Trap matching:** The OID attribute of the global_snmp_event_settings_for_group element can be a comma-delimited list of tokens. A single token has the following syntax:

  ```
  [enterpriseOID][-specificType]
  ```

- **Example:** "1.2.3.5.1.4,1.2.3.4.5.6.7.8.9-0" The first token matches any trap with an enterprise OID of 1.2.3.5.1.4. The second token matches any trap with an enterprise of 1.2.3.4.5.6.7.8.9 and specific of 0. Because the tokens are listed together in one attribute group, an event received that matches either is processed by that attribute group.

d. Every event that is received is processed only by the first attribute group that matches the received event. Subnode attribute groups are processed first, and then the base attribute groups are processed. The agent developer must ensure that the groups are defined in a way so that events are received in the expected attribute group.

5. In the **SNMP Event Information** window, select the **Subnode Host matching** check box to match events to subnodes. If the SNMP event attribute group is part of a subnode, you can select the **Subnode Host Matching** check box to control whether the event must come from the SNMP agent that is monitored.

   For example: You have an agent to monitor routers, where each subnode instance represents a specific router. You develop an agent to collect data from a router with the SNMP data collector. You also define an attribute group to receive SNMP events sent by that router. Each router instance includes the same data that is defined for the event filter. Therefore, you need another way to make sure that events from your router are shown in the attribute group for that router.

   When subnode host-matching is selected, an event that is sent by the router is compared to the host defined for the SNMP data collector. If the host in use by the SNMP data collector is the same host that sent the received event, the subnode instance processes the SNMP event. Otherwise, the event is passed to the next subnode instance. Address-matching applies only to subnodes. No address-matching is done by the SNMP event attribute groups in the base agent. For the address-matching to work, the subnode definition must contain at least one SNMP attribute group. The SNMP host that is used by SNMP for that subnode instance is the host that is used for matching.

   If the **Subnode Host Matching** check box is clear, your subnode instances do not do this extra comparison. You must allow the user to configure a different OID filter for each subnode in this case. Otherwise, you do not need to include SNMP event attribute groups in the subnode definition.

6. In the **SNMP Event Information** window, select the operating systems.
7. Optional: You can click **Test** in the **SNMP Event Information** window to start and test your agent.

   For more information, see "Testing SNMP event attribute groups" on page 1212

8. Optional:

   In the **SNMP Event Information** window, click **Advanced** to select **Event Filtering and Summarization Options**. For more information, see "Event filtering and summarization" on page 1373.

   a) When you finish selecting **Event Filtering and Summarization Options**, return to the **SNMP Event Information** window. If you previously selected **Custom Events** in the **SNMP Event Information** window, click **Next**, to select key attributes, otherwise skip the next step.

   b) On the Select key attributes page, click one or more key attributes for the attribute group, or click **Produces a single data row**.

9. Click **Next**, or click **Finish** if you are using the new agent wizard to save the agent and open the Agent Editor.
10.

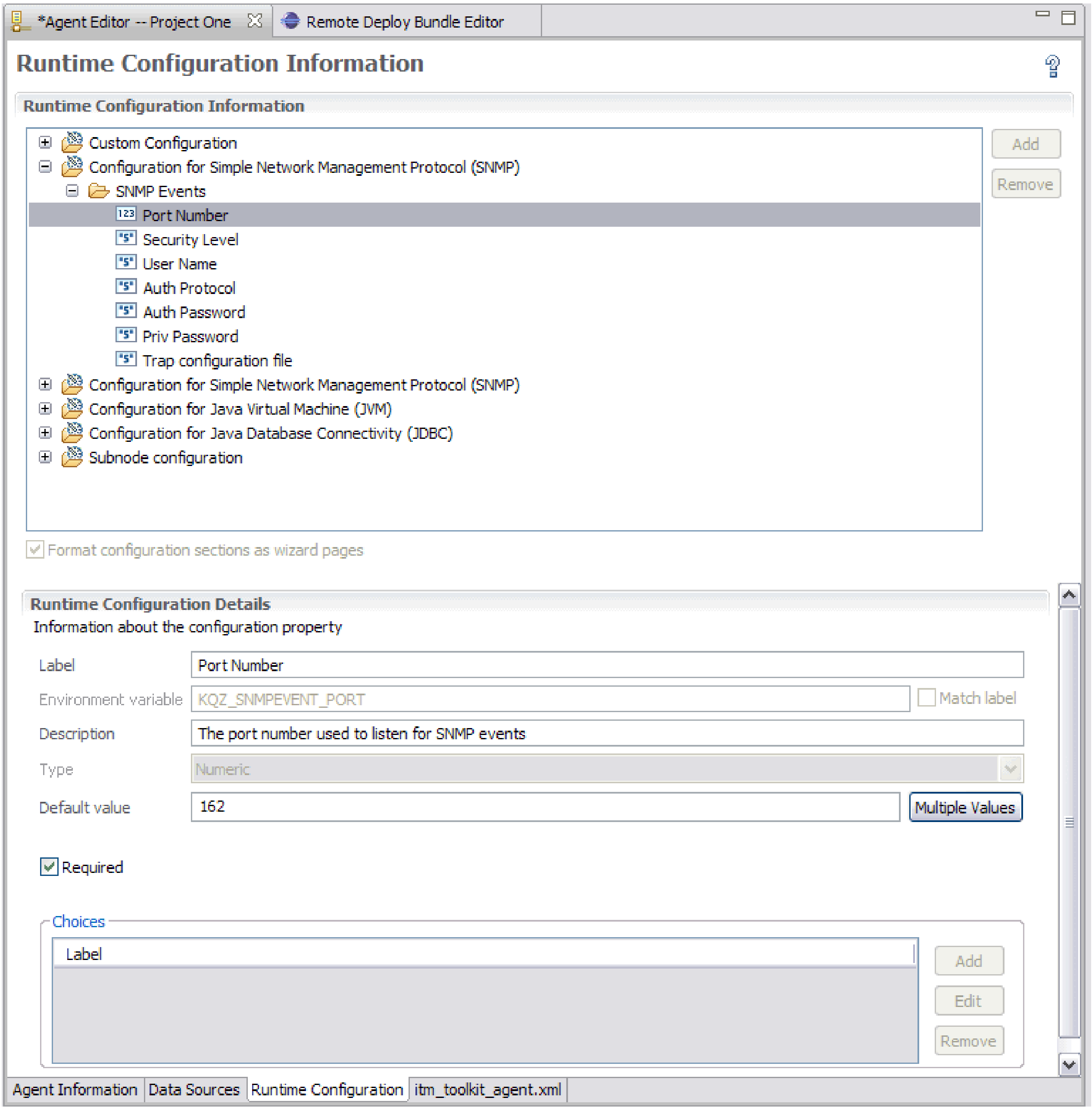
What to do next

For information about adding further attributes, see ("Creating attributes" on page 1169).

SNMP Event Configuration properties

Certain configuration properties are automatically created when an SNMP Event attribute group is added to the agent

After a data source is added, the configuration is displayed on the **Runtime Configuration Information** page of the Agent Editor. For example, Figure 32 on page 1211 shows the configuration sections and some configuration properties that are automatically created when an SNMP Event attribute group is added to the agent.



*Agent Editor -- Project One | Remote Deploy Bundle Editor

Runtime Configuration Information

Runtime Configuration Information

- ⊞ Custom Configuration
- ⊟ Configuration for Simple Network Management Protocol (SNMP)
  - ⊟ SNMP Events
    - Port Number
    - Security Level
    - User Name
    - Auth Protocol
    - Auth Password
    - Priv Password
    - Trap configuration file
- ⊞ Configuration for Simple Network Management Protocol (SNMP)
- ⊞ Configuration for Java Virtual Machine (JVM)
- ⊞ Configuration for Java Database Connectivity (JDBC)
- ⊞ Subnode configuration

Add | Remove

☑ Format configuration sections as wizard pages

Runtime Configuration Details

Information about the configuration property

Label	Port Number
Environment variable	KQZ_SNMPEVENT_PORT ☐ Match label
Description	The port number used to listen for SNMP events
Type	Numeric
Default value	162 [Multiple Values]

☑ Required

Choices

Label

Add | Edit | Remove

Agent Information | Data Sources | Runtime Configuration | itm_toolkit_agent.xml

Figure 32. Runtime Configuration page

The labels, descriptions, and default values of predefined configuration properties can be changed, but variable names and types cannot be changed. The SNMP Events configuration section contains the following properties:

Table 269. SNMP Events configuration properties

Name	Valid values	Required	Description
Port Number	positive integer	Yes	Required port number that is used for listening to events
Security Level	noAuthNoPriv, authNoPriv, authPriv	No	SNMP V3 security level
User Name	String	No	SNMP V3 user name
Auth Protocol	MD5 or SHA	No	SNMP V3 authentication protocol
Auth Password	String	No	SNMP V3 authentication password
Priv Password	String	No	SNMP V3 privacy password
Trap configuration file	File name that includes the path	No	Location of the trap configuration file. If the file is not located by using this configuration property, an attempt is made to find a `trapcnfg` file in the agent bin directory.

No configuration is required for V1 or V2C events. All V1 or V2C events are processed regardless of the source or community name specified. The only supported privacy protocol is DES, so there is no option to specify the privacy protocol. The SNMP V3 configuration options are not required (each can be optionally specified). If you want to specify them, you must specify the appropriate values for the security level you select.

Testing SNMP event attribute groups

You can test the SNMP event attribute group that you created, within Agent Builder.

Before you begin

To test the SNMP event attribute group, use a test program, or application to generate SNMP events.

Procedure

1. You can start the Testing procedure in the following ways:
   - During agent creation click **Test** in the **SNMP Event Information** window.
   - After agent creation, select an attribute group on the Agent Editor **Data Source Definition** page and click **Test**. For more information about the Agent Editor, see "Using the Agent Editor to modify the agent" on page 1152

   After you click **Test** in one of the previous two steps, the **Test Event Setting** window opens.
2. Optional: Before you start testing, you can set environment variables and configuration properties. For more information, see "Attribute group testing" on page 1347. For more about SNMP Event configuration properties, see "SNMP Event Configuration properties" on page 1211.
3. Click **Start Agent**. A window indicates that the Agent is starting.

   When the agent starts, it listens for SNMP events according to its configuration.

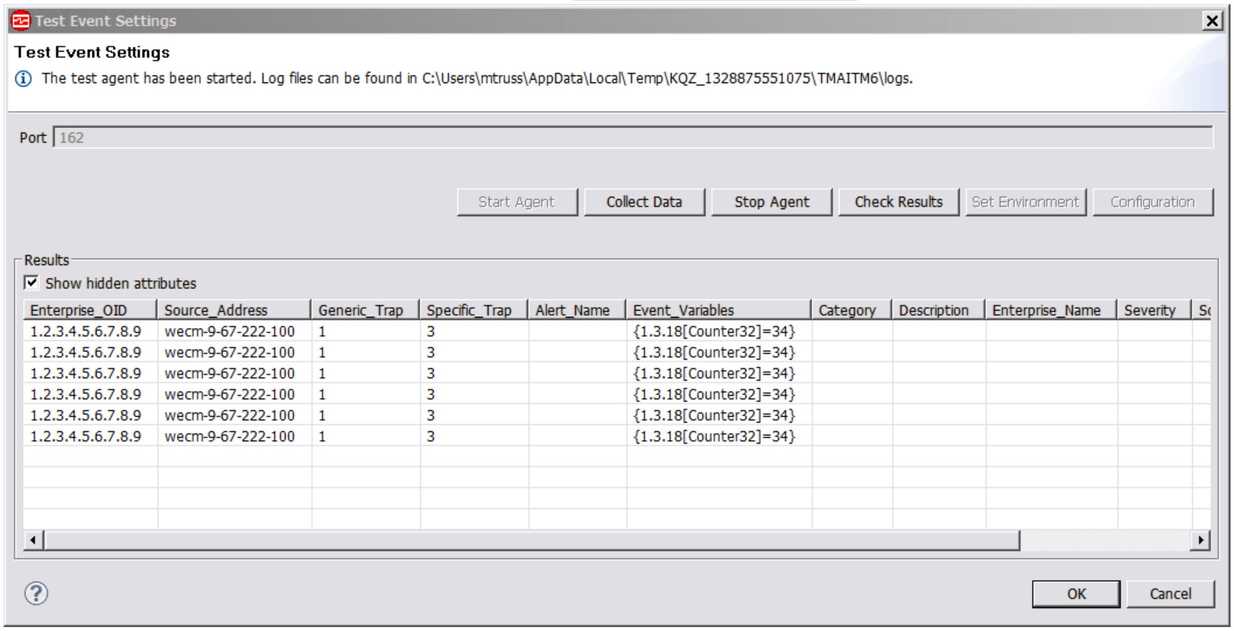
   Note: The agent that starts is a simplified version that includes the one attribute group you are testing.

4. To test your agent's data collection you generate SNMP events that match the agents configuration. You can do this using an application or an event generator.

   When the agent receives SNMP events that match its configuration, it adds the events to its internal cache.

5. To simulate a request from the monitoring environment for agent data, click **Collect Data**.

   The **Test Event Settings** window collects and shows any events in the agent's cache since it was last started. An example data collection is shown in Figure 33 on page 1213



Test Event Settings

Test Event Settings

ⓘ The test agent has been started. Log files can be found in C:\Users\mtruss\AppData\Local\Temp\KQZ_1328875551075\TMAITM6\logs.

Port 162

Results

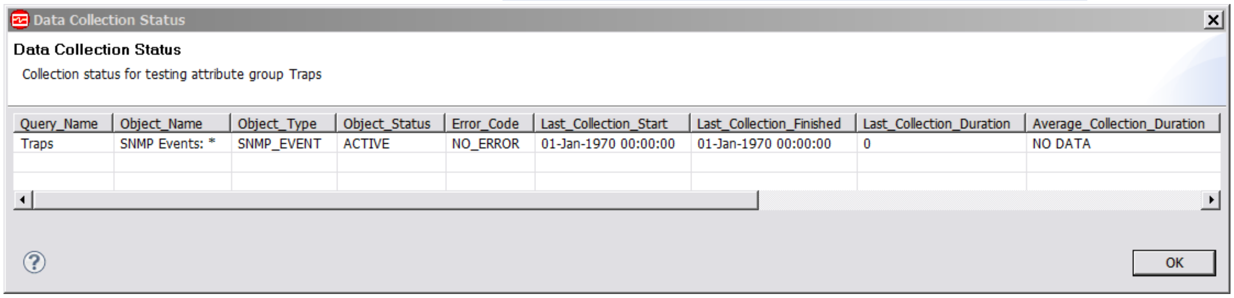
☑ Show hidden attributes

Enterprise_OID	Source_Address	Generic_Trap	Specific_Trap	Alert_Name	Event_Variables	Category	Description	Enterprise_Name	Severity	Sc
1.2.3.4.5.6.7.8.9	wecm-9-67-222-100	1	3		{1.3.18[Counter32]=34}					
1.2.3.4.5.6.7.8.9	wecm-9-67-222-100	1	3		{1.3.18[Counter32]=34}					
1.2.3.4.5.6.7.8.9	wecm-9-67-222-100	1	3		{1.3.18[Counter32]=34}					
1.2.3.4.5.6.7.8.9	wecm-9-67-222-100	1	3		{1.3.18[Counter32]=34}					
1.2.3.4.5.6.7.8.9	wecm-9-67-222-100	1	3		{1.3.18[Counter32]=34}					
1.2.3.4.5.6.7.8.9	wecm-9-67-222-100	1	3		{1.3.18[Counter32]=34}					

*Figure 33. **Test Event Settings** window that shows collected SNMP event data*

6. Optional: Click **Check Results** if the returned data is not as you expected.

   The **Data Collection Status** window opens and shows you more information about the data. An example is shown in (Figure 34 on page 1213). The data that is collected and shown by the **Data Collection Status** window is described in "Performance Object Status node" on page 1388



Data Collection Status

Data Collection Status

Collection status for testing attribute group Traps

Query_Name	Object_Name	Object_Type	Object_Status	Error_Code	Last_Collection_Start	Last_Collection_Finished	Last_Collection_Duration	Average_Collection_Duration
Traps	SNMP Events: *	SNMP_EVENT	ACTIVE	NO_ERROR	01-Jan-1970 00:00:00	01-Jan-1970 00:00:00	0	NO DATA

*Figure 34. **Data Collection Status** window*

7. Stop the agent by clicking **Stop Agent**.
8. Click **OK** or **Cancel** to exit the **Test Event Settings** window. Clicking **OK** saves any changes that you made.

Related concepts

"Testing your agent in Agent Builder" on page 1346

After you use Agent Builder to create an agent, you can test the agent in Agent Builder.

Monitoring Java Management Extensions (JMX) MBeans

You can define a data source to collect data from JMX MBeans. Data from every monitored MBean is placed into a data set. Depending on the MBean, the data set can produce a single row or multiple rows.

About this task

Each JMX data source that you define must identify either a single MBean (single instance) or a certain type of MBean (multiple instances). You must know the Object Name of the MBean or an Object Name pattern for a MBean type that contains the data you want to collect. Use an Object Name pattern to identify only a set of similar MBeans. The set of MBeans that matches the pattern must all provide the data that you want to see in the monitoring table. A typical Object Name pattern looks like `*:j2eeType=Servlet,*`. This Object Name Pattern matches all MBeans that have a j2eeType of servlet. You can expect any MBean matching that pattern to have a similar set of exposed attributes and operations that can be added to your data source. A data source that uses that pattern collects data from each MBean matching that pattern. The attributes that you define for this data source must be available for any MBean matching the Object Name pattern of the data source.

Java Version 5 or later is supported.

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Data from a server** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **JMX**.
3. Click **Next**.
4. On the **JMX Information** page , click **Browse** to see all of the JMX MBeans on the MBean server.

   After you define the data source, you can use the browse function to pre-populate your attribute list. You can then add to, remove from, or modify the attributes that the browser inserted. The names for these attributes can be long and difficult to type correctly. Using the Browse option is an easy way to input the correct name.

   Note: You can manually create JMX data sources by specifying an Object Name and clicking **Next** without using the browser. Manually creating JMX data sources creates two data sources. An event data source that contains predefined attributes for JMX notifications is created. Also, a collection data source is defined containing one attribute that you must specify in the wizard.

   MBean pattern
   Shows the MBean pattern.

   Global JMX Options
   Shows the level of support.

   Support is provided for the following JMX servers:

   - Java 5 operating system MBean Server. Connection is made by using the JSR-160 connector. Notifications and monitors are supported.
   - WebSphere Application Server, version 6 and later. Connectors are provided for both SOAP and RMI protocols. JMX Monitors are not supported because MBeans cannot be created by a remote agent.
   - WebSphere Community Edition and other Apache Geronimo-based application servers. Connection is made through standard JSR-160 connectors. JMX notifications and monitors are supported in versions 1.1 and later.
   - JBoss Application Server, version 4.0 and earlier.
   - JBoss Application Server, JSR-160 connection.
   - WebLogic Server, version 9 and newer. Connector is provided for T3 protocol.

5. The first time that you run the JMX browser, there are no items in the **MBean server** scroll down menu. To add connections, click the **Add**) button.

   Use the **Edit** button to modify or delete the connection that you already defined and selected in the scroll down menu. The connection definitions are stored in the workspace, so that, when you create a connection, it is remembered. Complete the following steps to create a connection. If you already have a connection, skip to the next step.

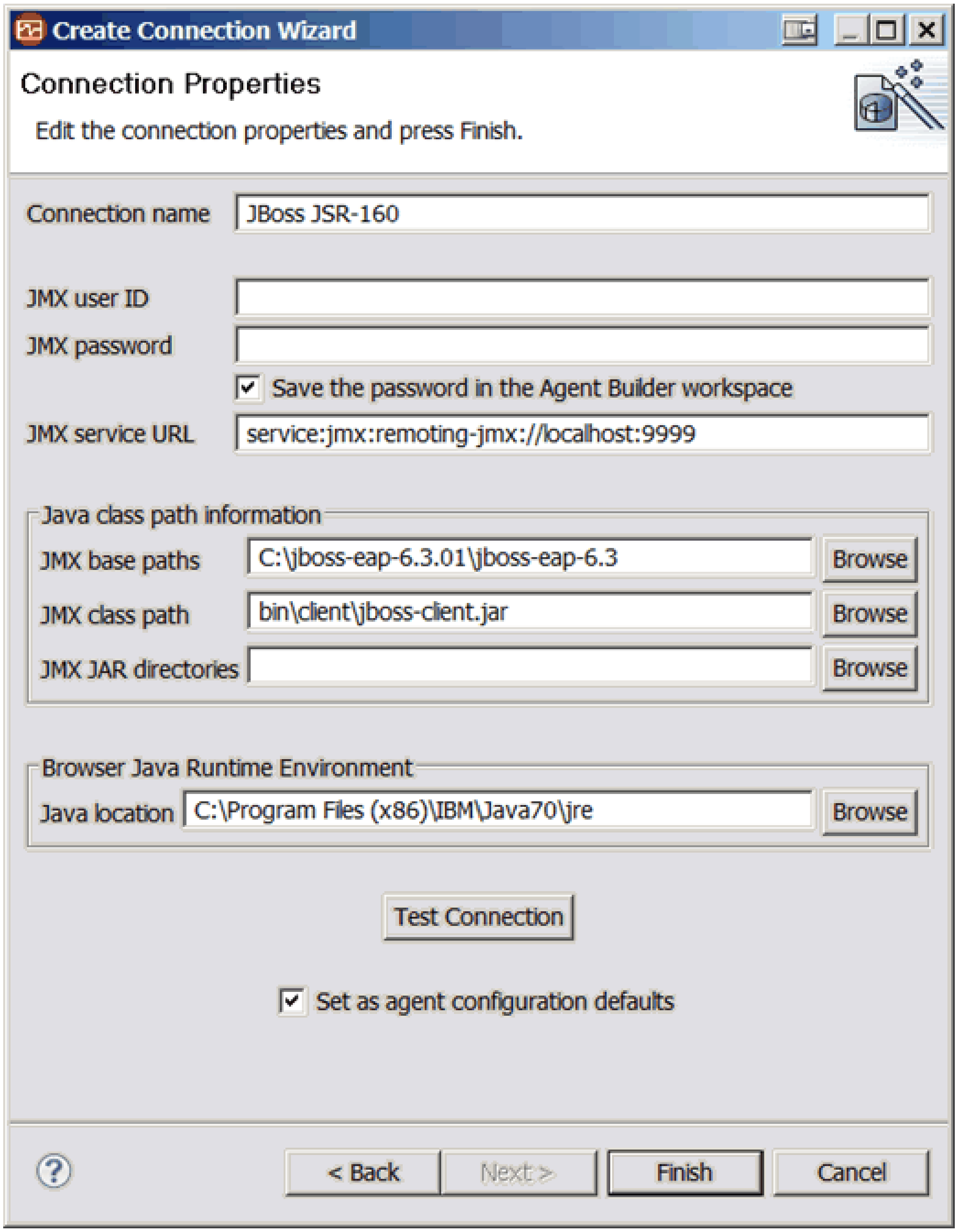
   a) To create a connection to an MBean Server, click **Add** to add a connection or to edit an existing connection.

      The **Java Management Extensions (JMX) Browser** window is shown when no connections are defined.

   b) After you click **Add** to add a connection, the **Select Connection Type** page opens.

   c) Use the MBean Server Connection wizard to connect to an MBean server. The new connections that are listed on the page are selections that you can make to create connection. You can use the list of existing connections to create a new connection using an existing connection as a template. Select one of the new connection types and click **Next** to begin creating a connection.

   d) After you select a connection type, you might be asked to select a more specific type of connection. Two templates that are based on the **Standard JMX Connections (JSR-160)** connection type are shown. Select the template that is most appropriate for your MBean server and click **Next**.



Connection name: JBoss JSR-160

JMX user ID:

JMX password:

☑ Save the password in the Agent Builder workspace

JMX service URL: service:jmx:remoting-jmx://localhost:9999

Java class path information

JMX base paths: C:\jboss-eap-6.3.01\jboss-eap-6.3

JMX class path: bin\client\jboss-client.jar

JMX JAR directories:

Browser Java Runtime Environment

Java location: C:\Program Files (x86)\IBM\Java70\jre

☑ Set as agent configuration defaults

Figure 35. JMX connection properties

The **Connection Properties** page (Figure 35 on page 1216) contains the details on how to connect to an MBean server. You must complete the page with details about your MBean server.

Important: If your data source connects to a remote WebSphere Application Server, ensure that WebSphere Application Server is also installed on the host that is running Agent Builder and set the**Java location** setting to the Java runtime environment that the local WebSphere Application Server uses.

e) Select the **Save the password in the Agent Builder workspace** check box if you want to save the password for this connection.

f) Optional: Select **Set as agent configuration defaults** if you want the defaults for JMX to be copied from these connection properties.

For example, in Figure 35 on page 1216 the default **JMX base path** is `C:\jboss-eap-6.3.01\jboss-eap-6.3`, the **JMX service URL** is `service:jmx:remoting-jmx://localhost:9999` and the **Java location** is `C:\Program Files\IBM\Java70\jre`

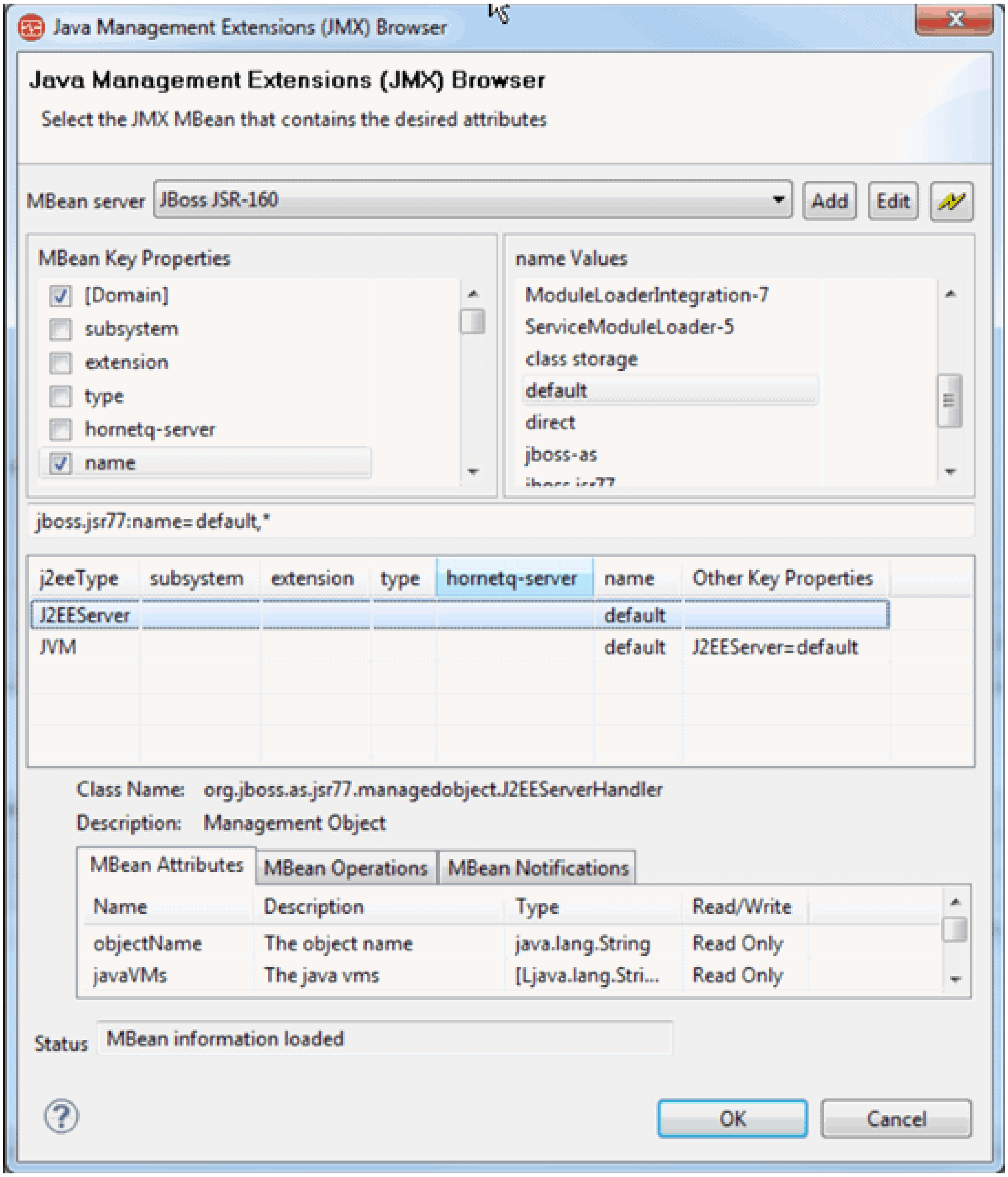
1) After you specify the properties that are required to connect, click **Test Connection** to ensure that the connection can be established. If the connection is not successful, correct the necessary properties.
2) When the connection is successful, click **Finish** to return to the browser that uses the connection you configured.

The Java class path information in the **Connection Properties** page contains three fields. These fields must be completed as necessary to connect to an MBean server that requires Java classes that are not included in the Java runtime environment. Normally, the MBean server you want to connect to must be installed on the same system as the Agent Builder. In this case, specify the directory where the application that contains the MBean server was installed as the **JMX base paths** field. The **JMX Jar Directories** field then lists the directories relative to the Base Paths directory that contain the JAR files that are required to connect to the MBean server. The **JMX class path** field can be used to include specific JAR files. The JAR files that are listed in the **JMX JAR Directories** field are not required to be listed separately in the **JMX class path** field.

Any of the fields can contain more than one reference; separate the entries by a semicolon. These values are the same values that are required when you configure the agent. For more information, see ("JMX configuration" on page 1222).

6. After you select a connection, the JMX Browser downloads information about the MBeans from the JMX server. This information is shown in the following four areas of the **JMX Browser** window (Figure 36 on page 1218):

   Directions for screens that begin with Java Management Extensions (JMX) Browser window to **Runtime Configuration** tab of the Agent Editor: From the **JMX Information** page, select **Browse**. From the browser (JMX Browser with no connection-selected), select **Add**. From the **JMX Connection Selection** page select **JBoss**, then select **Next**. From the **JMX Connection Properties** page, customize two Connection Properties: JBoss provider URL`: jnp://wapwin3.tivlab.raleigh.ibm.com:1099/` and **JBoss Jar Directories**: The full path to the directory that contains the following JAR files: `jbossall-client.jar`, `jboss-jmx.jar`, `jboss-jsr77-client.jar`, `jboss-management.jar`. Select **Finish**. This configuration sets up your JBoss connection so you can get similar screens as shown here.



Java Management Extensions (JMX) Browser

Select the JMX MBean that contains the desired attributes

MBean server: JBoss JSR-160

MBean Key Properties

- ☑ [Domain]
- □ subsystem
- □ extension
- □ type
- □ hornetq-server
- ☑ name

name Values

- ModuleLoaderIntegration-7
- ServiceModuleLoader-5
- class storage
- default
- direct
- jboss-as
- jboss.jsr77

jboss.jsr77:name=default,*

j2eeType	subsystem	extension	type	hornetq-server	name	Other Key Properties
J2EEServer					default	
JVM					default	J2EEServer=default

Class Name: org.jboss.as.jsr77.managedobject.J2EEServerHandler

Description: Management Object

MBean Attributes | MBean Operations | MBean Notifications

Name	Description	Type	Read/Write
objectName	The object name	java.lang.String	Read Only
javaVMs	The java vms	[Ljava.lang.Stri...	Read Only

Status: MBean information loaded

Figure 36. Java Management Extensions (JMX) Browser window

- **MBean Key Properties** area: This area is a collection of every unique Object Name key that is found from all the MBeans on the server. The **[Domain]** entry is special because it is not really a key. However, the **[Domain]** entry is treated as an implied key for the value of the MBean domain. Select an item from this list, and the MBeans that contain that key property are found. The list of values of the key property are shown in the **Selected Key Property Values** list. When you check a key property, it is included in the Object Name pattern for the data source.
- **Selected Key Property Values** area: This area shows the values of the currently selected MBean Key Property from all MBeans. Selecting one of these values checks the MBean key property. The selection also updates the Object Name Pattern shown in the message field with the MBean key property name and value.
- A table lists all MBeans matching the Object Name Pattern: As you select Key Properties and Values from the MBean Key Properties and Selected Key Property Values lists, you see the Object Name Pattern update. You also see the list of MBeans in this table change to reflect the list of MBeans that match the pattern you selected. If you have a pattern that is not matching any MBeans, you

can clear entries in the MBean Key Properties list. You clear entries by clicking the check box next to a key that is being used by your pattern and removing the check mark. Also, you can manually edit the pattern to find the MBeans you are looking for. The pattern *:* selects all MBeans.

You can use this table to browse the MBeans from the server and decide which ones contain the data you want to monitor. To help browse a potentially large number of MBeans, you can sort by any key attribute (from the menu or by clicking a column header). You can also show any key attribute in any column by selecting **Show Key Property** from the menu. When you see a key property value in the table that identifies MBeans you want to monitor, right-click on that value and choose **Select only MBeans with Key Property** from the menu.

- A table that contains details for a selected MBean: The JMX Browser shows you information about a single MBean. To see details for an MBean, you select the MBean from the table that shows the list of MBeans matching the current filter. The key information about the MBean is the list of Attributes, Operations, and Notifications it defines.

To create a data source from the JMX Browser, use the four panels that were described earlier to build an Object Name Pattern. Build the Object Name Pattern to match a set of MBeans that each contains the monitoring data you want to collect. For instance, if you wanted to monitor data from all of the ThreadPool MBeans, use the following steps:

a) Select **type** from the **MBean Key Properties** panel. Selecting **type** causes the values in the **Selected Key Properties Values** to be updated to list all unique values from the type key of any MBean.

b) Select **ThreadPool** from the list of values for the type key. After you select **ThreadPool**, the type key property name is selected in the **MBean Key Properties** panel and the Object Name Pattern is updated to *:type=ThreadPool,*. The list of MBeans is also updated to show only the MBeans that match this pattern.

c) Select one of the MBeans from the MBean list to see the attributes, operations, and notifications available for the MBean. If your MBean list contains more MBeans than you want to monitor, you must continue this procedure of selecting key properties and values. Continue until you have the Object Name Pattern that identifies the set of MBeans you want to monitor. You can also open a menu in the MBean list to update the Object Pattern with key property values shown in the table.

7. When the object name pattern is correct, select an MBean from the table.

   All attributes of the selected MBean are the initial attributes in the new JMX data source. Some attributes might not contain data. After the JMX data source is created, review the attributes and remove any that are not significant. If the selected MBean has no attributes, you are warned that the data source is created with no attributes. If the selected MBean contains notifications, an Event data source is also created to receive notifications from the MBeans.

   Important: For every MBean attribute, Agent Builder creates an attribute in the new data set. For a numeric MBean attribute, Agent Builder creates a numeric attribute. For any object types, including `String`, Agent Builder creates a string attribute containing a string representation of the value. If an object from an MBean attribute is of the `javax.management.openmbean.CompositeData` type, and the Agent Builder browser can read the object itself, it creates several attributes, one for each object embedded in the `CompositeData` object. To include values internal to an object other than a `CompositeData` object (fields or method return values), you need to create an attribute that has a more complex metric name, as described in "Specific fields for Java Management Extensions (JMX) MBeans" on page 1229.

8. Click **Finish** in the filled on JMX Information page.

   Data sources are created based on the MBean that was selected in the previous step. If no MBean was selected, an attribute group with no attributes is created. A warning is shown, giving you a chance to select an MBean. The notification data source has the word, **Event**, at the beginning of the data source name to distinguish it from the data source that shows attributes.

9. To change other JMX options for the agent, click **Global JMX Options**. With these options, you can:

   a) Choose whether JMX monitors are supported by this agent. If you want JMX monitor attribute groups and Take Action commands to be created, select **Include JMX monitor attribute groups and take actions**

See the next section for a description of JMX monitors.

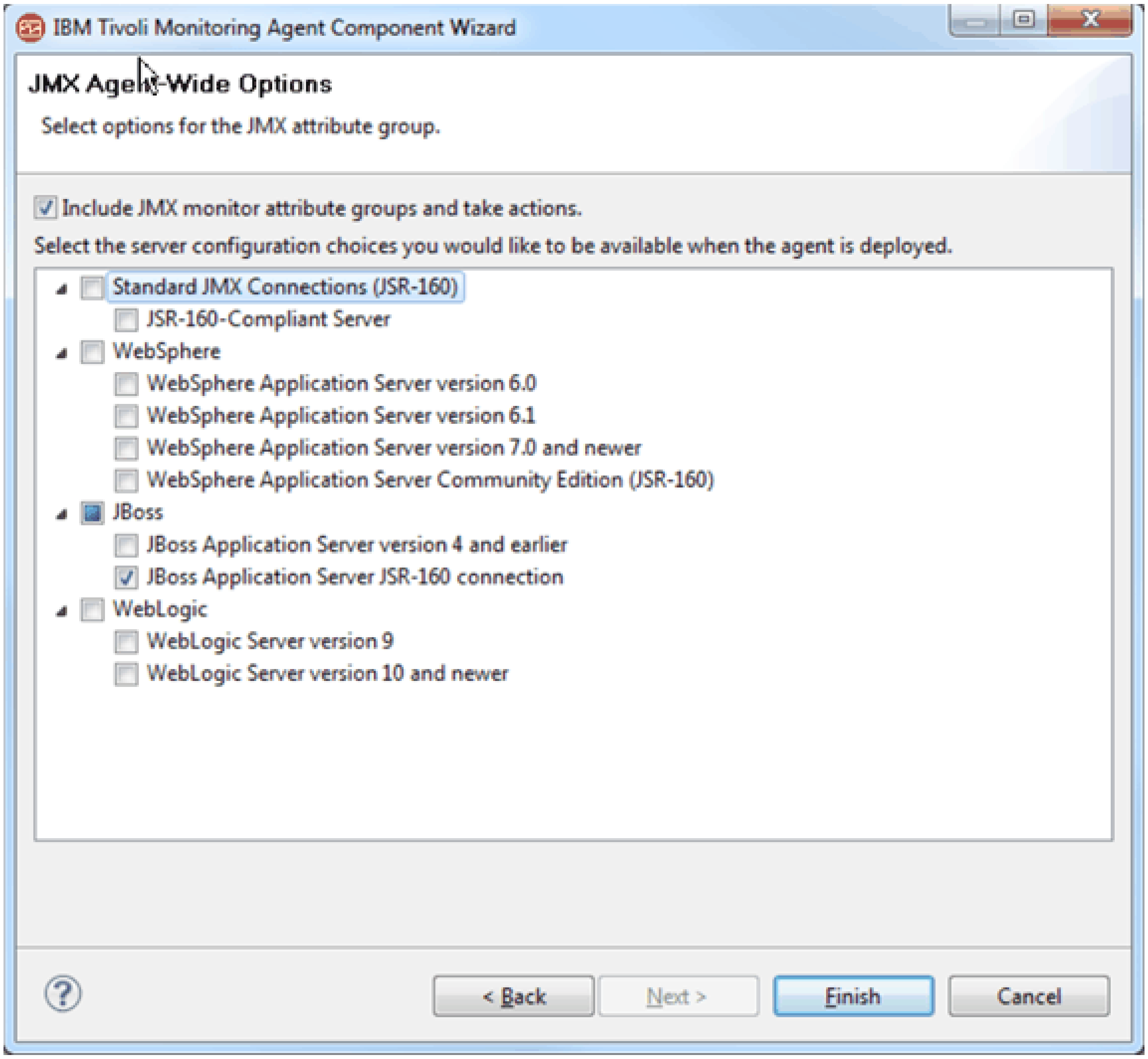
b) Select the types of MBean servers your agent connects to when it is deployed.

   There are several vendor-specific types of servers that are listed, along with a generic JSR-160-Compliant Server for standards-based servers. You can select as many as needed, but you must select only server types that support the MBeans that are being monitored. You must select at least one. If you select more than one, at agent configuration time you are prompted to specify which type of server you want to connect to.

10. Click **OK** after you select the wanted option.
11. Optional: You can test this attribute group by clicking **Test**. For more information about testing, see (“Testing JMX attribute groups” on page 1232)
12. Optional: You can create a filter to limit the data that is returned by this attribute group by clicking **Advanced**. For more information about filtering data from an attribute group, see “Filtering attribute groups” on page 1178
13. Click **Next**.
14. On the **Select key attributes** page, select key attributes or indicate that this data source produces only one data row. For more information, see (“Selecting key attributes” on page 1151).
15. Click **Next**.

    The **JMX Agent-Wide Options** window shows the types of application servers that the Agent Builder supports. If you previously selected **Set as agent configuration defaults** on the **Connection Properties** page, the type of application server that you browsed to is automatically selected.
16. In the **JMX Agent-Wide Options** window (Figure 37 on page 1221), select any other types of application servers to which you want your agent to be able to connect.

    Note: In the example shown, choosing **JBoss Application Server JSR-160 connection** is the same as choosing **JSR-160-Compliant Server** except that different default values are supplied.



JMX Agent-Wide Options

Select options for the JMX attribute group.

☑ Include JMX monitor attribute groups and take actions.

Select the server configuration choices you would like to be available when the agent is deployed.

- □ Standard JMX Connections (JSR-160)
  - □ JSR-160-Compliant Server
- □ WebSphere
  - □ WebSphere Application Server version 6.0
  - □ WebSphere Application Server version 6.1
  - □ WebSphere Application Server version 7.0 and newer
  - □ WebSphere Application Server Community Edition (JSR-160)
- ▣ JBoss
  - □ JBoss Application Server version 4 and earlier
  - ☑ JBoss Application Server JSR-160 connection
- □ WebLogic
  - □ WebLogic Server version 9
  - □ WebLogic Server version 10 and newer

Figure 37. JMX Agent-Wide Options window

17. Do one of the following steps:
    - If you are using the New Agent wizard, click **Next**.
    - Click **Finish** to save the data source and open the Agent Editor.
18. If you want to change the types of application servers to which you can connect after the agent is created, click **Global JMX Options** in the **JMX Data Source Information** area.
19. In the **JMX Agent-Wide Options** window, change any selections that you want.
20. Click **OK**.
21. To view the configuration sections and properties that were automatically generated, click the **Runtime Configuration** tab of the Agent Editor.

    The default value of the `JBoss base paths` property has the value that was entered in the JMX browser.

What to do next

For more information about the attribute groups for JMX events, see "JMX Event attribute groups" on page 1415,

JMX configuration

When you define a JMX data source in your agent, some configuration properties are created for you.

JMX runtime configuration is unique because it provides you with some control over how much configuration is displayed. The JMX client for the agent can connect to several different types of application servers. However, it is not necessary to support all of those types of application servers in any one agent. You can determine which types of application servers to support, and unnecessary configuration sections are not included in the agent.

In most cases, an agent is designed to monitor one JMX application server type. When you create the JMX data source, you can use the JMX Browser. When you use the JMX Browser, the JMX server configuration options that are used to browse the MBean server are added to your agent automatically. To change the types of application servers to which you can connect after the agent is created, click **Global JMX Options** in the **JMX Information** area. In the **JMX Agent-Wide Options** page, change any selections that you want.

You can design a generic agent that monitors more than one type of JMX application server. In this case, more than one JMX server configuration choice can be selected on the **JMX Agent-Wide Options** page. When more than one type of JMX connection is supported, the runtime configuration prompts you for the connection type that are used for that agent instance.

Note: An instance of an agent can connect only to one type of JMX application server. Subnodes can be used to connect to different JMX application servers of the same type within an agent instance. To connect to more than one type of JMX application server, you must configure at least one agent instance for each JMX application server type.

You can view, add, and change the configuration properties by using the Agent Editor. For instructions, see "Changing configuration properties by using the Agent Editor" on page 1333. If a JMX data source is defined in a subnode, you are also able to specify Subnode Configuration Overrides. For instructions, see "Subnode configuration" on page 1320.

If you define a JMX data source in your agent, the agent must use Java to connect to the JMX application server. Java configuration properties are added to the agent automatically.

The following Java configuration properties are specific to the agent runtime configuration:

Java Home

Fully qualified path that points to the Java installation directory

Configure the agent to use the same JVM that the application you are monitoring uses, particularly for the WebLogic Server and WebSphere Application Server.

JVM Arguments

Specifies an optional list of arguments to the Java virtual machine.

Trace Level

Defines the amount of information to write to the Java trace file. The default is to write-only Error data to the log file.

Note: Agent Builder does not require these properties because it uses its own JVM and logging, which is configured through the JLog plug-in.

If you define a JMX data source in your agent, the following required, common configuration fields are added to the agent automatically:

Connection

The type of connection to the MBean server

User ID

User ID that is used to authenticate with the MBean server.

Password

Password for the user ID.

Base paths

Directories that are searched for JAR files that are named in **Class path**, or directories that are named in **JAR directories**, that are not fully qualified. Directory names are separated by a semi-colon (`;`) on Windows, and by a semi-colon (`;`) or colon (`:`) on UNIX systems.

Class path

Explicitly named JAR files to be searched by the agent. Any that are not fully qualified are appended to each of the Base Paths until the JAR file is found.

JAR directories

Directories that are searched for JAR files. Directory names are separated by a semi-colon (`;`) on Windows, and by a semi-colon (`;`) or colon (`:`) on UNIX systems. The JAR files in these directories are not required to be explicitly identified; they are found because they are in one of these directories. Subdirectories of these directories are not searched. Any directory names that are not fully qualified are appended to each of the Base Paths until the directory is found.

Note: For remote monitoring, the JAR files and all of their dependent JAR files must be installed locally on the computer where the agent is running. These JAR files are the files that are required to connect to the application that is being monitored. These JAR files must be configured in **JAR directories**, and in **Base paths** and **Class path**. In addition, locally install a supported JVM for the application you are monitoring and specify the path in the **Java Home configuration** field.

Examples:

- For WebLogic 10, the class path is `server/lib/wlclient.jar;server/lib/wljmxclient.jar`. The base path points to the WebLogic application server directory where the `server/lib` directory is located.
- For WebSphere, the base path points to the location where the WebSphere Application Server is installed. Multiple base paths are listed in this example to provide a default for Windows and UNIX. The class path lists the JAR files relative to the base path. The relative value `lib` for the **JAR directories** field causes all JAR files in this directory under the base path to be loaded.
  - **Base paths**: `C:\Program Files\IBM\WebSphere\AppServer;/opt/IBM/WebSphere/AppServer`
  - **Class path**: `runtimes/com.ibm.ws.admin.client_6.1.0.jar;plugins/com.ibm.ws.security.crypto_6.1.0.jar`
  - **JAR directories**: `lib`

Depending on which JMX server types are selected in the JMX Agent-Wide Options page, some or all of the following configuration properties are added. Default values are provided by the Agent Builder, and can be modified:

JSR-160 Compliant Server connection-specific configuration properties:

JMX Service URL

JMX Services URL to connect to for monitoring.

WebSphere Application Server version 6.0 and later connection-specific configuration properties:

Host name

Host name of the system where the application server you are monitoring is located. For local monitoring, the name is the local system name. For remote monitoring, the name is the host name of the system where the application server is located.

Port

Port number to use on the host name to be monitored.

Connector protocol

Connector protocol to be used by the monitoring connection. RMI and SOAP are supported.

Profile name

Name of the profile to use for configuring the connection.

JBoss Application Server (non JSR-160) connection-specific configuration properties:

JNDI Name
JNDI Name that is used to look up the MBean server.

Provider URL
JMX Services provider URL to connect to for monitoring.

WebLogic Server connection-specific configuration properties:

Service URL
JMX Services provider URL to connect to for monitoring that includes the JNDI name.

Note: If WebSphere administrative security is enabled, you must make sure that client login prompts are disabled in the appropriate client connection properties files. For RMI connections, to prevent clients from prompting the user, you must modify the *com.ibm.CORBA.loginSource* property in the `sas.client.props` file in the profile properties directory of your WebSphere Application Server. For a SOAP connection, you must modify the *com.ibm.SOAP.loginSource property*in the `soap.client.props` file in the same directory. In both cases, the *loginSource* property must be set to not contain a value.

You can view, add, and change the configuration properties by using the Agent Editor. See ("Changing configuration properties by using the Agent Editor" on page 1333). If a Windows data source is defined in a subnode, you can also specify Subnode Configuration Overrides. See "Subnode configuration" on page 1320.

JMX notifications

In addition to providing monitoring data when requested, some MBeans also provide notifications.

A notification is an object that is generated by an MBean that is passed to registered listeners when an event occurs.

Agents that are built by the Agent Builder can define attribute groups that contain values from notifications rather than MBeans.

When the agent is started, a `notification listener` is registered with each MBean that matches the MBean pattern of the attribute group. The attribute group then displays one row per notification received. Each column contains one item of data from the notification. The data wanted from the notification is defined by a column value similar to the way column data is defined for MBeans.

For non-event based attribute groups, data is collected when needed. For event-based attribute groups, the agent maintains a cache of the last 100 events received. These events are used to respond to requests from the Tivoli Enterprise Portal. The events are forwarded immediately for analysis by situations and warehousing.

JMX monitors

In addition to providing monitoring data when requested, some MBeans also provide monitors.

The JMX Provider supports the ability for an agent to create JMX Monitors. A JMX Monitor is an MBean that the JMX agent creates on the JMX Server. It monitors the value of an attribute of another MBean and sends a notification when that value meets some criteria. Thresholds are defined that enable the Monitor to report on specific attribute values.

Not all application servers support the creation of monitors from a JMX client, which is true for current releases of WebSphere Application Server. JMX Monitors and Take Action commands can be included in your agent by selecting **Include JMX monitor attribute groups and take actions** under **Global JMX Options**.

Any MBean that reports on an attribute of another MBean can be considered a monitor. In practice, JMX defines three concrete monitor classes, which are the types of monitors that are created. The following concrete monitor types are created:

- String monitor – watches a string attribute, reports equality, or inequality of that string.
- Gauge monitor – watches a variable numeric attribute, reports up or down movement beyond threshold values.

- Counter monitor – watches an increasing numeric attribute, reports when it reaches a threshold value or increases by a certain amount.

The following attribute groups might be automatically added to the agent to collect or represent JMX Monitor notifications:

- `Registered Monitors`

  This attribute group displays all of the JMX Monitors that are added by the user.
- `Counter Notifications`

  This attribute group reports all notifications that are received from Counter Monitors.
- `Gauge Notifications`

  This attribute group reports all notification received from Gauge Monitors.
- `String Notifications`

  This attribute group reports all notifications that are received from String Monitors.

Take Action commands for JMX monitors

A monitor is created by running a Take Action command.

Three Take Action commands are defined, one to create each type of monitor, and a fourth Take Action is defined to delete an existing monitor. A 256-character limit applies to Take Action commands.

The monitor attribute groups are a part of every JMX agent that is built, including all agents that are built by the Agent Builder. The four Take Action commands are available to all agents, though they cannot be used unless it is a JMX agent.

JMX Add String Metric Watcher

Use this Take Action command to create a monitor to watch a string attribute.

Parameters

MBean pattern
All MBeans matching this pattern are monitored by this monitor.

Observed attribute
Name of the MBean string attribute that is being watched.

Notify match
True if a notification is to be sent when the monitored string matches a reference value, false if not (defaults to false).

Notify differ
True if a notification is to be sent when the monitored string does not match the reference value, false if not (defaults to true)

Reference value
String to compare with the observed attribute.

A default means that the argument is not specified.

Example: Request a notification when a service is stopped

```
STRING_METRIC_WATCHER [*:type=Service,*] [StateString] [true] [false] [Stopped]
```

Where:

`*:type=Service,*`
MBean pattern: Monitors any MBean with a key property named type whose value is Service.

`StateString`
Observed attribute: A string attribute that is common to all MBeans of `type=Service`.

true
Notify match: You want a notification to be sent to your agent when the `StateString` attribute matches your reference value of Stopped.

false
Notify differ: You do not want to be notified when the Service attribute does not match Stopped.

Stopped
Reference value: When the `StateString` attribute changes to the value Stopped, a notification is sent.

JMX Add Gauge Metric Watcher
Use this Take Action command to create a monitor to watch a gauge attribute.

Parameters

MBean pattern
All MBeans matching this pattern are monitored by this monitor.

Observed attribute
Name of the MBean string attribute that is being watched.

Difference mode
True if the value monitored is the difference between the actual current and previous values of the attribute. False if the value monitored is the actual current value of the attribute (defaults to false).

Notify high
True if a notification is to be sent when an increasing monitored value crosses the high threshold, false if not (defaults to true).

Notify low
True if a notification is to be sent when a decreasing monitored value crosses the low threshold, false if not (defaults to true).

High threshold
Value that the observed attribute is expected to stay under.

Low threshold
Value that the observed attribute is expected to stay over.

Example: Request a notification when free memory goes under 10 Mb

```
GAUGE_METRIC_WATCHER [ServerInfo] [FreeMemory] [false] [false] [true] [30000000] [10000000]
```

Where:

***:type=ServerInfo**
MBean pattern: Monitors any MBean whose name has a single key property named type whose value is ServerInfo.

FreeMemory
Observed attribute: Numeric attribute that fluctuates up or down, this one indicating the amount of free memory in the application server.

false
Difference mode: Monitors the actual attribute value, not the difference between one observation and another.

false
Notify high: Notification is not sent when free memory goes up.

true
Notify low: Notification is not sent when the free memory becomes too low.

30000000
High threshold: Even though you are not concerned with passing a high threshold, you need a reasonable high threshold value. A second low threshold notification does not occur until the attribute value hits or passes the high threshold.

10000000
Low threshold: Low threshold value that you want to be notified about.

JMX Add Counter Metric Watcher
Use this Take Action command to create a monitor to watch a counter attribute.

Parameters

MBean pattern
All MBeans matching this pattern are monitored by this monitor.

Observed attribute
Name of the MBean string attribute that is being watched.

Initial threshold
Value that the observed attribute is compared.

Offset
Value added to the threshold after the threshold is exceeded to create a changed threshold.

Modulus
Maximum value of counter after which it rolls over to 0.

Difference mode
True if the value monitored is the difference between the actual current and previous values of the attribute. False if the value monitored is the actual current value of the attribute (defaults to false). This mode effectively turns on rate-of-change monitoring.

Granularity period
Frequency with which measurements are taken (defaults to 20 seconds). Most important if difference mode is true

Example: Request a notification when any server has three or more errors

```
COUNTER_METRIC_WATCHER [*:j2eeType=Servlet,*] [errorCount] [3] [4] [] [diff] [gran]
```

Where:

`*:j2eeType=Servlet,*`
MBean pattern: Monitors any J2EE servlet MBean whose name has a single key property named type whose value is `ServerInfo`

`errorCount`
Observed attribute: Increasing numeric attribute, this one indicating the number of errors of the servlet.

3
Initial threshold: You want to be notified when `errorCount` meets or exceeds 3.

4
Offset: When you get a notification for three errors, 4 is to the previous threshold of 3 to make a new threshold of 7. A second notification will be sent after `errorCount` reaches 7; a third at 11; a fourth at 15, and so on. Zero or none is not valid because it expects the counter to always increase and not increasing the offset would not make sense for a counter.

Modulus:
errorCount has no architected maximum value, so use an unreasonably high value.

false
Difference mode: You are concerned with absolute error counts. Difference is true if you are interested in the rate that errorCount was increasing.

Granularity period: Not set, so take the default granularity period of 20 seconds. Granularity period is available for all monitor types. However, it is shown with a counter monitor so that a meaningful rate of change (with difference mode=true) can be determined.

JMX Delete Metric Watcher
Use this Take Action to delete a monitor.

Parameter

Number
Monitor number as shown in the REGISTERED_MONITORS table

Example: Delete monitor number 2

```
DELETE_WATCHER [2]
```

Where:

2=
Number of monitor to be deleted.

JMX operations

In addition to providing monitoring data when requested, some MBeans also provide operations.

Agents that have JMX data sources include the `JMX_INVOKE`Take Action command that you can use to run JMX operations against the server you are monitoring.

Take Action command syntax

The action has the following syntax:

```
JMX_INVOKE [MBean pattern] [Operation name] [Argument 1] [Argument 2]
[Argument 3] [Argument 4]
```

Where:

MBean pattern
MBean query that selects the MBeans on which the operation runs. If the pattern matches more than one MBean, the operation runs on each of the matched MBeans.

Operation name
Name of the MBean operation to run.

Argument 1, Argument 2, Argument 3, Argument 4
Optional arguments that can be provided to the MBean operation. Arguments must be a simple data type such as a string or an integer.

The `JMX invoke` Take Action command returns success if the operation is successfully run. If the operation returns a value, that value is written to the JMX data provider log file.

Example: Start an operation to reset a counter

This action runs the resetPeakThreadCount operation on the Threading MBeans:

```
JMX_INVOKE [*:type=Threading,*] [resetPeakThreadCount][] [] [] []
```

Where:

:type=Threading,
MBean Pattern: This pattern matches all MBeans that have a type of Threading.

resetPeakThreadCount
Operation name: The operation that is run on every MBean that matches the pattern.

[] [] [] []
Argument 1, 2, 3, 4: The arguments are not needed for this operation. They are specified only to comply with the syntax of the action.

Example: Start an action with an argument

This action runs the getThreadCpuTime operation on the Threading MBeans. The result is logged to the JMX data provider trace file.

```
JMX_INVOKE [*:type=Threading,*] [getThreadCpuTime] [1] [] [] []
```

Where:

:type=Threading,
MBean Pattern: This pattern matches all MBeans that have a type of Threading

getThreadCpuTime
Operation name: The operation that is run on every MBean that matches the pattern.

1
Argument 1: The thread id that is being queried.

[] [] []
Argument 2, 3, 4: These arguments are not needed for this operation. They are specified as empty arguments to comply with the Take Action command syntax.

Running the JMX_INVOKE Take Action command

The agent developer cannot expect the user to run the `JMX_INVOKE` Take Action command. Instead, more actions must be developed that run the `JMX_INVOKE` Take Action. If possible in these actions, hide the details such as the operation name and the MBean pattern from the user.

Starting and stopping JMX monitors

JMX monitors are persistent across starts and stops of the agent and the JMX server.

If the agent detects that the JMX server was recycled, it reregisters the monitors. If the agent is recycled, monitors are reregistered. The monitor definitions are stored in a file that is named `default_`*`instanceName`*`.monitors` where *instanceName* is the agent instance name or default if it is a single instance agent. This file is in the following directory (note that *xx* denotes the two character product code):

- Windows systems: `TMAITM6/k`*`xx`*`/config`
- UNIX and Linux systems: *`architecture`*`/`*`xx`*`/config` (see "New files on your system" on page 1362 for information about determining the architecture value)

If the agent is restarted, it uses the monitor definitions file to restore the monitors.

Specific fields for Java Management Extensions (JMX) MBeans

The syntax of the metric name for a JMX Attribute group must follow certain rules when specified on the **Attribute Information** window.

The syntax of the metric name for a JMX Attribute group consists of tokens that are separated by a period. The tokens form primary values and optionally secondary values:

- **Primary value**: a value that is obtained directly from the MBean or Notification in a specific row of the table. Primary values from an MBean are obtained either from an MBean attribute or from the invocation of an MBean operation (method call). Primary values from a Notification are obtained from a field or invocation of a method on the Notification object. Primary values can be primitive types, or can be Java objects.
- **Secondary value**: a value that is obtained by further processing a primary value or other secondary value. Secondary values are processed internally to the engine and do not involve calls to the JMX server. If the primary (or other secondary value) is a Java object, a secondary value is the result of fetching a public field from that object. A secondary value can also be the result of a method call on that

object. Such secondary values are obtained by using Java introspection of the primary (or other secondary) Java object. If the primary (or other secondary) value is a Java String in the form of an MBean name, the secondary value can be the domain. The secondary value can also be any of the properties that make up the MBean name.

The following syntax describes the format for the **Metric name** field:

```
Metric Name    =    PrimaryValue [ .SecondaryValue ]
PrimaryValue    =    Attribute.attributeName |
        Method.methodName |
        Domain |
        Property.propertyName |
        Field.fieldName |
        Name
SecondaryValue    =    Field.fieldName |
        Method.methodName |
        Domain |
        Property.propertyName |
        Explode |
        ElementCount
```

```
propertyName    =    the name of a key property in an MBean ObjectName
attributeName    =    the name of an MBean attribute
methodName    =    a zero-argument operation of an MBean or a zero-argument method
of a Notification or other Java object.
methodName(argument) = A single-argument operation of an MBean or a
single-argument method of a Notification or other Java object. The
argument will be passed to the method as a string.
fieldName    =    the name of a public instance variable in a Notification or
other Java object
notificationMethod    =    the name of a public zero-argument method of a
Notification object
```

By including only a primary value in the metric name definition, the data that is collected can be any of the following items:

- MBean domain
- MBean string value
- Key property from the MBean name
- Numeric or string attribute value on an MBean attribute (including the full name of another MBean). A numeric or string return value from an operation of a MBean.
- Value of a numeric or string public instance variable in a Notification object
- Numeric or string return value from an operation of a Notification.

By adding a secondary value to the definition of a metric, you can drill down into the primary value of a Java object. Also, you can start a public method or fetch a public instance variable.

By adding a secondary value to another secondary value in the definition of the metric, you can drill down into a secondary value object. You can continue as deeply as objects are nested inside an MBean or a Notification.

Tokens that make up primary and secondary values are either keywords or names. In most cases, a keyword token is followed by a name token. The following table shows some examples:

Metric name sample	Attribute group type	Description of the data returned
`Domain`	MBean	The domain portion of the MBean (the part before the colon).
`Name`	MBean	The full string representation of the MBean.
`Attribute.serverVendor`	MBean	MBean attribute serverVendor.
`Method.getHeapSize`	MBean	The value that is returned by the getHeapSize() on the MBean.

Metric name sample	Attribute group type	Description of the data returned
`Property.j2eeType`	MBean	The value of j2eeType is extracted from the MBean name.
`Field.Message`	Event (Notification)	The **Message** field in a notification.

The keywords `Attribute`, `Method`, and `Field` can return Java objects which contain other data. You can run operations on those objects by appending secondary value definitions. More examples:

Metric name sample	Attribute group type	Description of the data returned
Attribute.deployedObject.Method.getName	MBean	Takes the deployedObject attribute from the MBean and gets the result of the getName() method.
Attribute.eventProvider.Method.getException.Method.getDescription	MBean	Goes 3 deep: an attribute named eventProvider is presumed to be an object which has a getException() method. This method returns an object with a getDescription() method. That method is called and the return value is put in the column.
Attribute.HeapMemoryUsage.Method.get(used)	MBean	Takes the HeapMemoryUsage attribute from the MBean and gets the result of the get(String value) method. The string that is used is passed to the method as the argument. Only 1 argument can be provided and it must be a literal string value. Shows how you can collect data from an open MBean composite data structure.

`Domain` and `Property` can be used as keywords in secondary values if the previous value returned a String in the format of an MBean name. For example:

Metric name sample	Attribute group type	Description of the data returned
`Attribute.jdbcDriver.Property.name`	MBean	The attribute jdbcDriver returns an MBean name, and the key property, name, is extracted from the MBean name.
`Attribute.jdbcDriver.Domain`	MBean	The attribute jdbcDriver returns an MBean name, and the domain is extracted from the MBean name.

The `ElementCount` and `Explode` keywords run operations on arrays or collections of data.

- `ElementCount` – returns the number of elements in an array.
- `Explode` – explodes a row into several rows, one new row for each element of an array.

Examples of each of the keywords:

Metric name sample	Attribute group type	Description of the data returned
Attribute.deployedObjects.ElementCount	MBean	The MBean attribute deployedObjects is an array, and this column contains the number of elements in the array.
Attribute.deployedObjects.Explode.MBean.Property.j2eeType	MBean	Causes the table to have 1 row for each element in deployed objects. This column contains the j2eeType of the deployed Object.
Attribute.SystemProperties.Method.values.Explode.Method.get(key)	MBean	Causes you to get one new row for each entry in an open MBean tabular data structure. Each tabular data structure contains a composite data structure with an item named key, which is returned.

Testing JMX attribute groups
You can test the JMX attribute group that you created within Agent Builder.

Procedure

1. You can start the Testing procedure in the following ways:
   - During agent creation click **Test** on the **JMX Information** page.
   - After agent creation, select an attribute group on the Agent Editor **Data Source Definition** page and click **Test** . For more information about the Agent Editor, see "Using the Agent Editor to modify the agent" on page 1152.

   After you click **Test** in one of the previous two steps, the **JMX Test** window is displayed
2. Select a connection from the list available under **Connection Name** or alternatively click **Add** to add a connection and follow the procedure that is detailed under "Monitoring Java Management Extensions (JMX) MBeans" on page 1214.
3. Optional: Before you start testing, you can set environment variables, configuration properties, and Java information.

   For more information, see "Attribute group testing" on page 1347. For more about JMX configuration, see "JMX configuration" on page 1222.
4. Click **Start Agent**.

   A window indicates that the Agent is starting.
5. Click **Collect Data** to simulate a request from Tivoli Enterprise Portal or SOAP for agent data.

   The agent monitors the JMX Server for data. The **JMX Test** window collects and shows any data in the agent's cache since it was last started.
6. Optional: Click **Check Results** if the returned data is not as you expected.

   The **Data Collection Status** window opens and shows you more information about the data. The data that is collected and displayed by the Data collection Status window is described in "Performance Object Status node" on page 1388
7. Stop the agent by clicking **Stop Agent**.
8. Click **OK** or **Cancel** to exit the **JMX Test** window. Clicking **OK** saves any changes that you made.

Related concepts
"Testing your agent in Agent Builder" on page 1346

After you use Agent Builder to create an agent, you can test the agent in Agent Builder.

Monitoring data from a Common Information Model (CIM)

You can define a data source to receive data from a Common Information Model (CIM) data source. A data source monitors a single CIM class and places all values from this class into the data set that it produces. If the class provides several instances, the data set has multiple rows; you can filter by instance name to ensure the data set has one row.

About this task

This task describes the steps to configure a Common Information Model (CIM) data source.

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Data from a server** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **CIM**.
3. Click **Next**.
4. On the **Common Information Model (CIM) Information** page **CIM information** area, make one of the following choices:
   - Complete the **Namespace** and **CIM class name** fields for the data that you want to collect.
   - Click **Browse** to browse a CIM repository on a specific system.

     The **Common Information Model (CIM) Class Browser** window is displayed. This browser connects to a CIM server and provides you with information about the classes that exist on that server.

     To browse a remote system, select a system from the **Hostname** list (if one is defined). Alternatively click **Add** to add the host name of the system where the CIM server is located.

     The syntax for specifying the host name is `http[s]://`*`hostname`*`:`*`port`*. If you provide the host name only, the Common Information Model (CIM) Class Browser connects by using a default URL of `http://`*`hostname`*`:5988`.

     If you provide a protocol without specifying a port, 5988 is used as the default for `http` or 5989 as the default for `https`.

     If you provide a port without specifying a protocol, `http` is used with the port provided.

     Provide a user ID and password for an account with read permission to the objects in the namespace that you want to browse. The window is updated with the information for the remote system.

     Agent Builder attempts to discover the namespaces available on the CIM Server. The discovered namespaces are displayed in the **Namespace** list. However, the Agent Builder might not be able to discover all namespaces that are available on the server. If you want to browse a namespace that is not listed in the **Namespace** list, click the plus (+) icon next to the **Namespace** list. Enter the name of the namespace in the field and click **OK**. If the namespace is present on the CIM server, the classes that are defined in the namespace are listed. The namespaces you type are saved and put into the **Namespace** list the next time you browse that particular CIM server.

     When you select a namespace from the **Namespace** list, the Agent Builder collects all of the class information for that particular namespace. Then, the Agent Builder caches this information so you can switch between namespaces quickly. If you want to force the Agent Builder to recollect the class information for a particular namespace, select the namespace and click **Connect**. Clicking **Connect** deletes any cached information, and causes the Agent Builder to recollect the class information.

     You can click the **Search** (binoculars) icon to find your selection in the list. Type a phrase in the **Search phrase** field; specify your preference by clicking either the **Search by name** or **Search by**

class properties fields; and click **OK**. If you find the item for which you are searching, select it and click **OK**.

5. On the Common Information Model (CIM) Information page, **Operating systems** area, select the operating systems on which the collection is to take place.
6. If you typed the Namespace and CIM class name in the **CIM information** area, do the following steps:

   a) Click **Next** to display the **Attribute Information** page and define the first attribute in the attribute group.

   b) Specify the information about the **Attribute Information** page, and click **Finish**.
7. If you browsed the CIM information, the Select key attributes page is displayed. On the Select key attributes page, select key attributes or indicate that this data source produces only one data row. For more information, see ("Selecting key attributes" on page 1151).
8. If you browsed to the CIM information, click **Finish**.
9. Optional: You can test this attribute group by clicking **Test**. For more information about testing, see "Testing CIM attribute groups" on page 1234
10. Optional: You can create a filter to limit the data that is returned by this attribute group by clicking **Advanced**. For more information about filtering data from an attribute group, see "Filtering attribute groups" on page 1178
11. Do one of the following steps:

    a) If you are using the **Agent** wizard, click **Next**.

    b) Click **Finish** to save the data source and open the Agent Editor.

CIM configuration

Details about CIM configuration properties.

If you define a CIM data source in your agent, CIM configuration properties are added to the agent automatically. You can view, add, and change the configuration properties by using the Agent Editor. For instructions, see "Changing configuration properties by using the Agent Editor" on page 1333). If a CIM data source is defined in a subnode, specify Subnode Configuration Overrides. For instructions, see "Subnode configuration" on page 1320.

The following connection-specific configuration properties are on the CIM configuration page:

CIM Local or Remote
Local or remote authentication to the CIM server. Local/Remote Default value is Remote

CIM user ID
The user ID used to access the CIM server

CIM password
The password to access the CIM server

CIM host name
The host name to be accessed for CIM data

CIM over SSL
Use SSL for communication with the CIM server. The options are Yes and No. The default value is No.

CIM port number
The port number that is used for communication that is not secure.

CIM SSL port number
The port number that is used for secure communication. The default value is 5989. (The default value for Solaris 8 is normally different.)

Testing CIM attribute groups

You can test the CIM attribute group that you created, within Agent Builder.

Procedure

1. Start the Testing procedure in the following ways:

- During agent creation click **Test** on the **CIM Information** page.
- After agent creation, select an attribute group on the Agent Editor **Data Source Definition** page and click **Test**. For more information about the Agent Editor, see "Using the Agent Editor to modify the agent" on page 1152

After you click **Test** in one of the previous two steps, the **Test Settings** window is displayed

2. Optional: Set environment variables and configuration properties before you start testing.

   For more information, see "Attribute group testing" on page 1347.
3. Select or add a **Host name**.

   For more about adding a **Host name**, see "Monitoring data from a Common Information Model (CIM)" on page 1233
4. Click **Start Agent**.

   A window opens indicating that the Agent is starting.
5. To simulate a request from Tivoli Enterprise Portal or SOAP for agent data, click **Collect Data**.

   The agent queries the CIM Server for data. The **Test Settings** window collects and shows any data in the agent's cache since it was last started.
6. Optional: Click **Check Results** if the returned data is not as you expected.

   The **Data Collection Status** window opens and shows you more information about the data. The data that is collected and displayed by the **Data Collection Status** window is described in "Performance Object Status node" on page 1388
7. Stop the agent by clicking **Stop Agent**.
8. Click **OK** or **Cancel** to exit the **Test Settings** window. Clicking **OK** saves any changes that you made.

Related concepts

"Testing your agent in Agent Builder" on page 1346
After you use Agent Builder to create an agent, you can test the agent in Agent Builder.

Monitoring a log file

You can define a data source to receive data from a text log file. The agent periodically parses the lines that are added the log file, and produces event information based on these lines. You can configure the way in which the agent parses the log into events. You can also configure the agent to filter and summarize the data. The resulting events are placed in a data set.

Before you begin

Note: The agent monitors log files that are in the same locale and code page that the agent runs in.

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Logged Data** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **A Log File**.
3. Click **Next**.
4. On the **Log File Information** page, type the name of the log file you want to monitor in the **Log File Information** area.

   The file name must be fully qualified.

   a) Optional: Part of the log file name can come from a runtime configuration property. To create a log file name, click **Insert Configuration Property** and select a configuration property.

   b) Optional: The file can also be a dynamic file name. For more information, see ("Dynamic file name support" on page 1464).
5. In the **Field Identification** area, click one of the following options:

   Fixed number of characters
   When selected, limits the number of characters.

With this option, each attribute is assigned the maximum number of characters it can hold from the log file. For example, if there are three attributes A, B, and C (in that order), and each attribute is a String of maximum length 20. Then, the first 20 bytes of the log record go into A, the second 20 into B, and the next 20 into C.

Tab separator
When selected, you can use tab separators.

Space separator
When selected, multiple concurrent spaces can be used as a single separator.

Separator Text
When selected, type in separator text.

Begin and End Text
When selected, type in both Begin and End text.

XML in element
When selected, type the name of the XML element to use as the record, or click **Browse** to define the element.

If you clicked **Browse**, the **XML Browser** window is displayed. If you use the browse function, the Agent Builder identifies all possible attributes of the record by looking at the child tags and their attributes.

Note: Unless you click **Advanced** and fill out the information in that window, the following assumptions are made about information that you complete:

- Only one log file at a time is monitored.
- Each line of the log file contains all the fields necessary to fill the attributes to be defined.

For more information about log file parsing and separators, see ("Log file parsing and separators" on page 1243).

6. Optional: Click **Advanced** on the **Log File Information** page to do the following by using the **Advanced Data Source Properties** page:

- Monitor more than one file, or monitor files with different names on different operating systems or monitor files with names that match regular expressions.
- Draw a set of fields from more than one line in the log file.
- Choose **Event Filtering and Summarization Options**.
- Produce output summary information. This summary produces an additional attribute group at each interval. For more information about this attribute group, see "Log File Summary" on page 1399. This function is deprecated by the options available in the Event Information tab.

a) To monitor more than one log file, click **Add** and type the name.

If more than one file is listed, a unique label must be entered for each file. The label can be displayed as an attribute to indicate which file generated the record. It must not contain spaces.

b) Optional: To select the operating systems on which each log file is to be monitored, follow these steps:

1) Click in the **Operating systems** column for the log file.

2) Click **Edit**.

3) In the **Operating Systems** window, select the operating systems.

4) Click **OK** to save your changes and return to the **Advanced Data Source Properties** page.

c) Optional: Select **File names match regular expression** if the file name you are providing is a regular expression that is used to find the file instead of being a file name.

For more information, see "ICU regular expressions" on page 1455. If you do not check this box, the name must be an actual file name. Alternatively it must be a pattern that follows the rules for file name patterns that are described in "Dynamic file name syntax" on page 1464.

d) Optional: Select **One directory element matches regular expression** to match one subdirectory of the file name path with a regular expression.

You can select this option only if you also selected **File names match regular expression** in the previous step.

If regular expression meta characters are used in the path name, the meta characters can be used in only one subdirectory of the path. For example, you can specify `/var/log/[0-9\.]*/mylog.*` to have meta characters in one subdirectory. The `[0-9\.]*` results in matching any subdirectory of `/var/log` that consists solely of numbers and dots `(.)`. The `mylog.*` results in matching any file names in those`/var/log` subdirectories that begin with `mylog` and are followed by zero or more characters.

Because some operating systems use the backslash `(\)` as a directory separator it can be confused with a regular expression escape meta character. Because of this confusion forward slashes must always be used to indicate directories. For example, Windows files that are specified as `C:\temp\mylog.*` might mean the `\t` is a shorthand tab character. Therefore, always use forward slashes `(/)` on all operating systems for directory separators. The `C:/temp/mylog.*` example represents all files in the `C:/temp` directory that start with `mylog`.

e) In the **When multiple files match** list, select one of the following options:

- **The file with the highest numerical value in the file name**
- **The biggest file**
- **The most recently-updated file**
- **The most recently-created file**
- **All files that match**

Note: When you select **All files that match**, the agent identifies all files in the directory that match the dynamic file name pattern. The agent monitors updates to all of the files in parallel. Data from all files is intermingled during the data collection process. Its best to add an attribute by selecting **Log file name** in **Record Field Information** to correlate log messages to the log files that contain the log messages. Ensure that all files that match the dynamic file name pattern can be split into attributes in a consistent manner. If the log files selected cannot be coherently parsed, then its best to select **Entire record** in **Record Field Information** to define a single attribute. For more information about specifying **Record Field Information** for attributes, see step ("8" on page 1239).

f) Choose how the file is processed.

With **Process all records when the file is sampled**, you can process all records in the entire file every time the defined sampling interval for the log monitor expires. The default interval is 60 seconds. This interval can be modified by using the *KUMP_DP_COPY_MODE_SAMPLE_INTERVAL* environment variable (specifying a value in seconds). The same records are reported every time unless they are removed from the file. With this selection, event data is not produced when new records are written to the file. With **Process new records appended to the file**, you can process new records that are appended to the file while the agent is running. An event record is produced for every record added to the file. If the file is replaced (first record changes in any way), the file is processed and an event is produced for each record in the file.

Note: If appending records to an XML log file, the append records must contain a complete set of elements that are defined within the XML element you selected as **Field Identification**.

g) If you chose to process new records that are appended to the file, you can also choose how new records are detected.

With **Detect new records when record count increases**, new records can be detected when the number of records in the file increases, whether the size of the file changes. This feature is useful when an entire log file is pre-allocated before any records are written to the file. This option can be selected for files that are not pre-allocated, but it is less efficient than monitoring the size of the file. With **Detect new records when the file size increases**, you can determine when a new entry is appended to a file in the typical way. There might be a brief delay in recognizing that a monitored file is replaced.

h) If you selected **Detect new records when the file size increases**, you can also choose how to process a file that exists when the monitoring agent starts.

   Ignore existing records disables event production for any record in the file at the time agent starts. **Process ___ existing records from the file** specifies production of an event for a fixed number of records from the end of the file at the time the agent starts. **Process records not previously processed by the agent**: Specifies for restart data to be maintained by the monitoring agent so the agent knows which records were processed the last time that it ran. Events are produced for any records that are appended to the file since the last time the agent was running. This option involves a little extra processing each time a record is added to the file.

i) If you selected **Process records not previously processed by the agent**, you can choose what to do when the agent starts and apparently the existing file was replaced.

   Process all records if the file has been replaced: If information about the monitored file and the restart data information do not match, events are produced for all records in the file. Examples of mismatches include: The file name is different, the file creation-time is different, the file-size decreased, the file last modification time is earlier than before. **Do not process records if the file has been replaced**: If the information about the monitored file and the restart data information do not match, disables processing of existing records in the file.

j) Click the **Record Identification** tab to interpret multiple lines in the log file as a single logical record.

   Note: If you select **XML in element** as the field identification on the **Log File Information** page, the **Record Identification** tab does not display.

   - **Single line** interprets each line as a single logical record.
   - **Separator line** you can enter a sequence of characters that identifies a line that separates one record from another.

     Note: The separator line is not part of the previous or the next record.
   - **Rule** identifies a maximum number of lines that make up a record and optionally a sequence of characters that indicate the beginning or end of a record. With **Rule**, you can specify the following properties:
     - **Maximum non-blank line** defines the maximum number of non-blank lines that can be processed by a rule.
     - **Type of rule**: Can be one of:
       - **No text comparison** (The Maximum lines per record indicates a single logical record).
       - **Identify the beginning of record** (Marks the start of the single logical record).
       - **Identify the end of record** (Marks the end of the single logical record).
     - **Offset**: Specifies the location within a line where the Comparison String must occur.
     - **Comparison Test**: Can either be **Equals**, requiring a character sequence match at the specific offset, or **Does not equal**, indicating a particular character sequence does not occur at the specific offset.
     - **Comparison String** defines the character sequence to be compared.
   - **Regular Expression** identify a pattern that is used to indicate the beginning or end of a record. By using **Regular Expression**, you can specify the following properties:
     - **Comparison String** defines the character sequence to be matched.

       OR
     - Beginning or end of record:
       - **Identify the beginning of record** marks the start of the single logical record.
       - **Identify the end of record** marks the end of the single logical record.

k) If you did select **Process all records when the file is sampled** earlier, click the **Filter Expression** tab. By clicking **Filter Expression** you can filter the data that is returned as rows based on the values of one or more attributes, configuration variables or both.

   If you selected **Process new records appended to the file** earlier you cannot create a filter expression. For more information about filtering data from an attribute group, see ("Filtering attribute groups" on page 1178).

l) If you selected **Process new records appended to the file** earlier, click the **Event Information** tab to select **Event Filtering and Summarization Options**.

   For more information, see ("Event filtering and summarization" on page 1373).

   Note: The Summary tab can be present if the agent was created with an earlier version of Agent Builder. The summary tab is now deprecated by the Event Information tab

7. Optional: Click **Test Log File Settings** on the **Log File Information** page to start and test the data source. . Click **Test Log File Settings** after you select the options for the log source. When you test the log file data source and supply log file content, Agent Builder creates the attributes in the group automatically, based on the results of parsing the log. For more information about testing, see "Testing log file attribute groups" on page 1244.
8. Use the following steps if you did not use the test function earlier and you typed the log file name in the **Log File Information** area of the **Log File Information** page:

   a) Click **Next** to display the **Attribute Information** page and define the first attribute in the attribute group.

   b) Specify the information, on the **Attribute Information** page, and click **Finish**.

   Note: When a log file attribute group is added to an agent at the default minimum Tivoli Monitoring version (6.2.1) or later, a `Log File Status` attribute group is included. For more information about the `Log File Status` attribute group, see ("Log File Status attribute group" on page 1430).

   Along with the fields applicable to all data sources, the **Attribute Information** page for the log file data source has some additional fields in the **Record Field Information** area.

   The **Record Field Information** fields are:

   Next field
   Shows the next field after parsing, by using the delimiters from the attribute group (or special delimiters for this attribute from the Advanced dialog).

   Remainder of record
   Shows the rest of the record after previous attributes are parsed. This attribute is the last attribute, except for possibly the log file name or log file label.

   Entire record
   Shows the entire record, which can be the only attribute, except for possibly the log file name or log file label.

   Log file name
   Shows the name of the log file.

   Log file label
   Shows the label that is assigned to the file on the advanced panel.

   Note: Use the **Derived Attribute Details** tab only if you want a derived attribute, and not an attribute directly from the log file.

9. Click **Advanced** in the **Record Field Information** area to display the **Advanced Log File Attribute Information** page.

   a) In the **Attribute Filters** section, specify the criteria for data to be included or excluded.

   Filtering attributes can enhance the performance of your solution by reducing the amount of data processed. Click one or more of the attribute filters:

   - **Inclusive** indicates that the attribute filter set is an acceptance filter, meaning that if the filter succeeds, the record passes the filter, and is output.

- **Exclusive** indicates that the attribute filter set is a rejection filter, meaning that if the attribute filter succeeds, the record is rejected, and is not output.
- **Match all filters** indicates that all filters defined to the filter must match the attribute record in order for the filter to succeed.
- **Match any Filter** indicates that if any of the filters that are defined to the filter match the attribute record, the filter succeeds.

b) Use **Add**, **Edit**, and **Remove** to define the individual filters for an attribute filter set.

c) To add a filter, follow these steps:

1) Click **Add**, and complete the options in the **Add Filter** window as follows:

a) The **Filter criteria** section defines the base characteristics of the filter, including the following properties:

- **Starting offset** defines the position in the attribute string where the comparison is to begin.
- **Comparison string** defines the pattern string against which the attribute is defined.

  Type a string, pattern, or regular expression that is used by the agent to filter the data read from the file. The records that match the filter pattern are eliminated from the records that are returned to the monitoring environment, or are the only records returned. The result depends on whether you choose for the filter to be inclusive or exclusive.
- **Match entire value** checks for an exact occurrence of the comparison string in the attribute string. Checking starts from the starting offset position.
- **Match any part of value** checks for the comparison string anywhere in the attribute string. Checking starts from the starting offset position.

b) **The comparison string is a regular expression** indicates that the comparison string is a regular expression pattern that can be applied against the attribute string.

Regular expression-filtering support is provided by using the International Components for Unicode (ICU) libraries to check whether the attribute value examined matches the specified pattern.

To effectively use regular expression support, you must be familiar with the specifics of how ICU implements regular-expressions. This implementation is not identical to how regular expression support is implemented in `Perl`, `grep`, `sed`, Java regular expressions, and other implementations. See "ICU regular expressions" on page 1455 for guidance on creating regular expression filters.

c) **Define an override filter** indicates that you want to provide a more specific filter comparison that overrides the base characteristics previously defined. This additional comparison string is used to reverse the filter result. When the filter is **Inclusive**, the override acts as an exclusion qualifier for the filter expression. When the filter is **Exclusive**, the override acts as an inclusion qualifier for the filter expression. (For more about **Inclusive** and **Exclusive**, see step "9" on page 1239, and the examples that follow). The override filter has the following properties:

- **Starting offset** defines the position in the attribute string where the comparison is to begin.
- **Comparison string** defines the pattern string against which the attribute is matched.

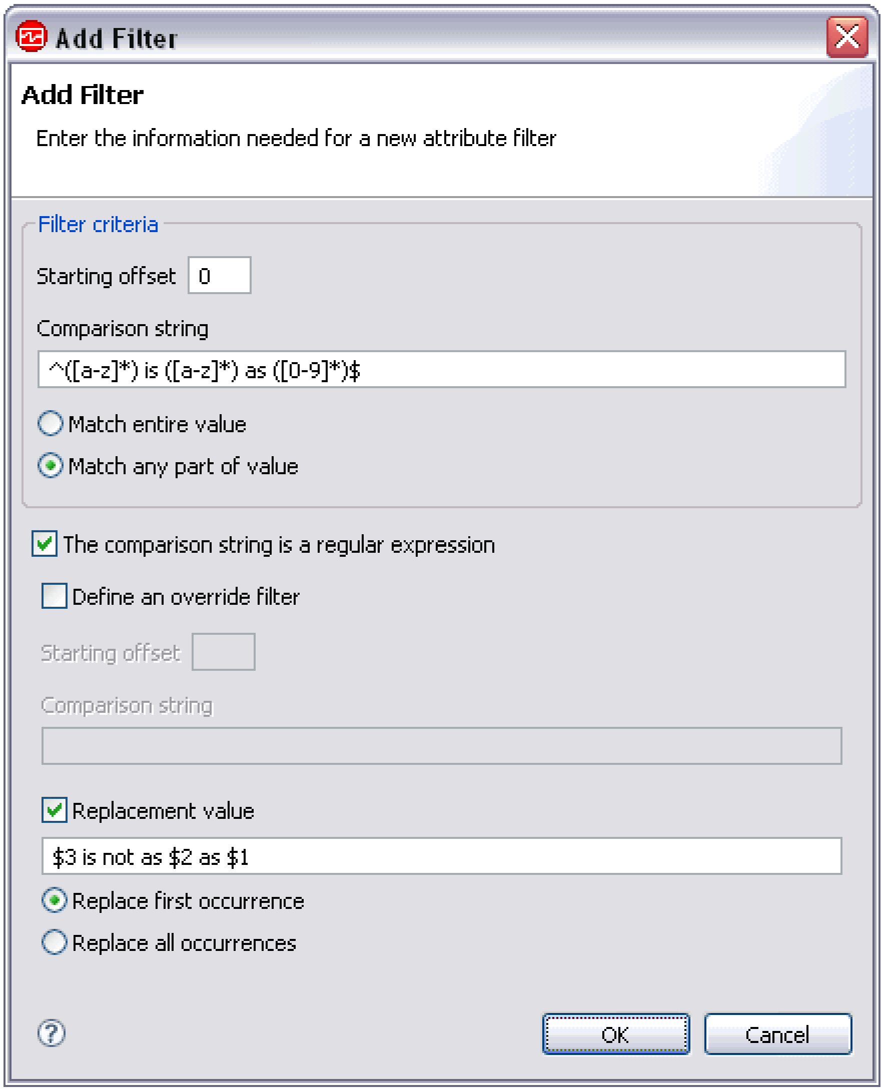
  Type a regular expression that is used by the agent to filter the data read from the file. The records that match the filter pattern are eliminated from the records that are returned to the monitoring environment, or are the only records returned. The result depends on whether you choose for the filter to be inclusive or exclusive.

d) **Replacement value** can be used to alter the raw attribute string with a new value. See "ICU regular expressions" on page 1455 for more details about special characters that can be used.

e) **Replace first occurrence** replaces the first occurrence that is matched by the comparison string with new text.

f) **Replace all occurrences** replaces all occurrences that are matched by the comparison string with new text.

2) Click **OK**.



Add Filter

Enter the information needed for a new attribute filter

Filter criteria

Starting offset: 0

Comparison string: `^([a-z]*) is ([a-z]*) as ([0-9]*)$`

○ Match entire value

● Match any part of value

☑ The comparison string is a regular expression

□ Define an override filter

Starting offset:

Comparison string:

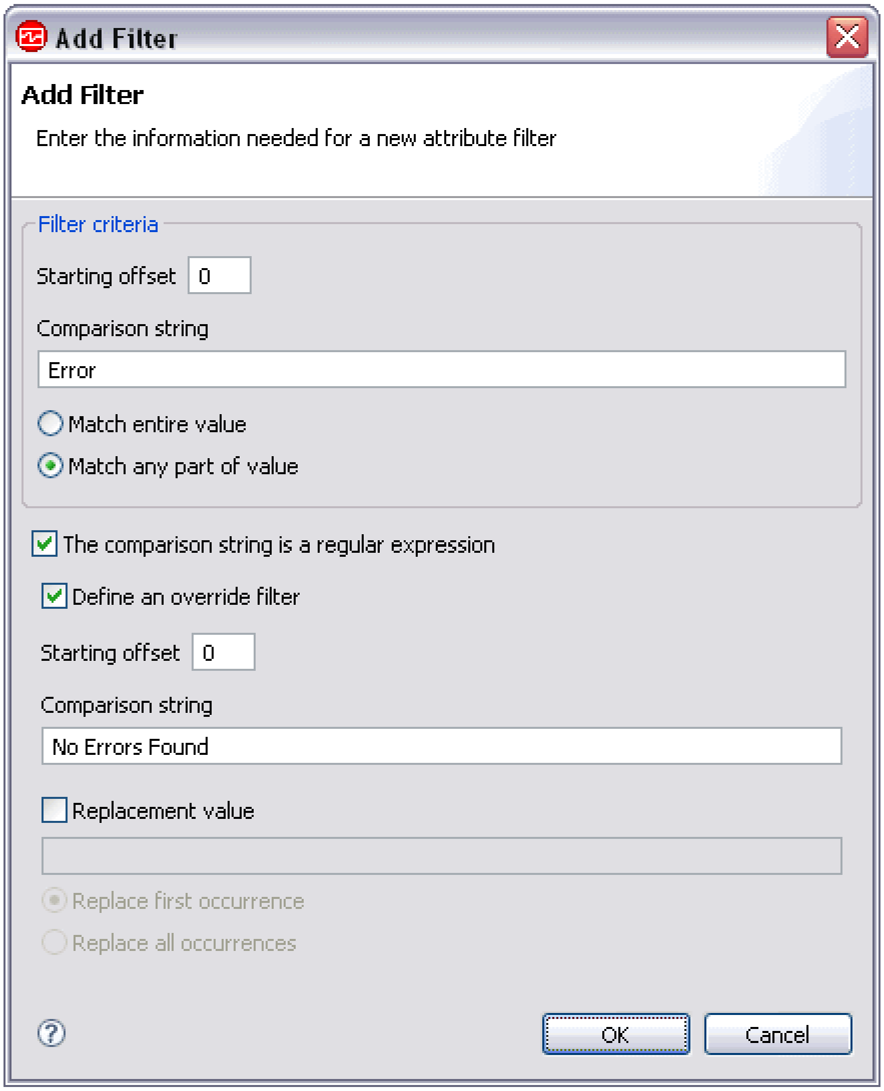
☑ Replacement value: `$3 is not as $2 as $1`

● Replace first occurrence

○ Replace all occurrences

Figure 38. ***Add Filter*** *example 1*

If the attribute string is `abc is easy as 123`, then the replaced string that is displayed in the Tivoli Enterprise Portal or IBM Cloud Application Performance Management console as `123 is not as easy as abc`.



Add Filter

Add Filter

Enter the information needed for a new attribute filter

Filter criteria

Starting offset: 0

Comparison string: Error

○ Match entire value

● Match any part of value

☑ The comparison string is a regular expression

☑ Define an override filter

Starting offset: 0

Comparison string: No Errors Found

☐ Replacement value

● Replace first occurrence

○ Replace all occurrences

Figure 39. ***Add Filter*** *example 2*

If the attribute string is `Unrecoverable Error reading from disk`, and the filter is **Inclusive**, then the attribute is displayed in the Tivoli Enterprise Portal or IBM Cloud Application Performance Management console. If the attribute string is `No Errors Found during weekly backup` and the filter is **Inclusive**, then the attribute is not displayed.

d) In the **Field Identification** section of the **Advanced Log File Attribute Information** page, specify how to override the attribute group field delimiters for this one attribute only. Click one of the attribute filters, and complete the required fields for the option:

- **Number of characters**: Enter the limit for the number of characters.
- **Tab separator** specifies the use of tab separators.
- **Separator Text**: Enter the separator text that you want to use.
- **Begin and End Text** Enter both **Begin** text and **End** text.

e) In the **Summary** section of the **Advanced Log File Attribute Information** page, click the **Include attribute in summary attribute group** check box to add the attribute to the summary attribute group.

This attribute group is produced when a user turns on log attribute summarization.

f) Click **OK**.

10. If you used the test function in step (“7” on page 1239), the **Select key attributes** page is displayed. On the **Select key attributes** page, select key attributes or indicate that this data source produces only one data row.

    For more information, see (“Selecting key attributes” on page 1151).

11. Do one of the following steps:

    - If you are using the New Agent wizard, click **Next**.
    - Click **Finish** to save the data source and open the Agent Editor.

    Note: When a log file attribute group is added to an agent with the default minimum Tivoli Monitoring version (6.2.1) or later, a `Log File Status` attribute group is included. For more information about the `Log File Status` attribute group, see (“Log File Status attribute group” on page 1430).

Log file parsing and separators

You can change the default separator that is used to separate one or more attributes in a log file record.

When you create a log file attribute group, a separator is by default assigned. The default separator is a tab. The separator is used by the agent to parse and delimit the data for each attribute in the data row. You can change the default attribute separator to be:

- A fixed number of characters
- A space
- A different character or characters
- A specific beginning and end text
- An XML element.

You change the default separator that is used for all attributes in the group in the following ways:

1. When you are creating the attribute group, on the **Log File Information** page.
2. After you create the attribute group, by opening the **Agent Editor** > **Data Sources** tab, selecting the attribute group and choosing a separator in the **Field Identification** area.

You can also optionally assign specific separators to one or more individual attributes. You can assign specific separators for individual attributes to use:

- A fixed number of characters.
- A tab separator
- A space separator
- A different character or characters
- A specific beginning and end text.

You change the separator that is used for individual attributes in the following ways:

1. By selecting **Advanced** on the **Attribute Information** page when you are creating an attribute.
2. By opening the **Agent Editor** > **Data Sources** tab, selecting the attribute and selecting **Advanced** on the **Log File Attribute Information** tab.

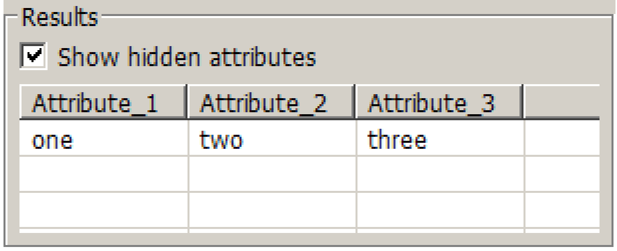
Example 1 - Simple log file output

Some log file records have clear and regular separators, for example:

```
one,two,three
```

Here the `","` character is a clear and regular separator between the three pieces of data on the row. In this case, select **Separator Text** and specify `","` as the default separator for the attribute group. There is no need to change or define other separators.

Defining this separator for a log file that contains the data row that is shown earlier in this example is shown in the following output:



Results

☑ Show hidden attributes

Attribute_1	Attribute_2	Attribute_3
one	two	three

Figure 40. Example attribute value output when Agent parses a simple log file data row.

Example 2 - Complex log file output

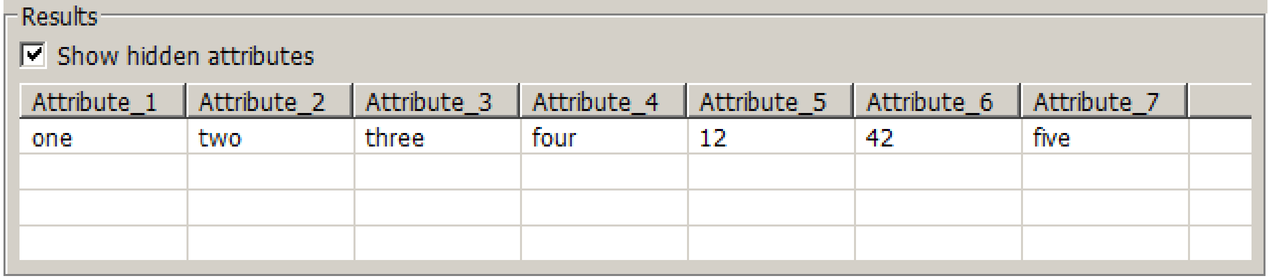
Some log files can contain data rows that have irregular or changing separators, for example:

```
one,two,three,[four]12:42,five
```

In this example an assignment of separators to attribute definitions that you can use is:

1. In the previous example you set the default separator to ",". This separator is used for all attributes unless you over-ride it with a specific separator. In this example the default separator of "," is correct to use again for the first three attributes in the row.
2. For the fourth attribute, assume the string between the"[" and "]" is a value that you want to extract. In this case when you define the fourth attribute, you assign a separator type **Begin and End Text** with begin and end text values of "[" and "]".
3. For the fifth attribute, assume that you want to extract the values between the "]" and ":" characters. In this case when you define the fifth attribute, you assign separator type **Separator Text** set to ":".
4. For the sixth attribute, your default attribute group separator "," is fine again.
5. For the seventh attribute, you do not need to specify a separator as it is the last attribute.

Defining these separators on a log file that contains the data row that is shown earlier in this example is shown in the following output:



Results

☑ Show hidden attributes

Attribute_1	Attribute_2	Attribute_3	Attribute_4	Attribute_5	Attribute_6	Attribute_7
one	two	three	four	12	42	five

Figure 41. Example attribute value output when Agent parses a complex log file data row.

The procedure to define the attribute separators is described under step “5” on page 1235 of “Monitoring a log file” on page 1235.

Testing log file attribute groups

You can use Agent Builder to test the log file data set (attribute group) that you created. If no attributes are defined for the group, the testing process defines them automatically.

Before you begin

If any attributes are already defined for this data set and you want to define attribtes automatically during testing, use the agent editor to remove all the existing attributes from the data set. For instructions, see “Removing attributes” on page 1172.

Procedure

1. You can start the Testing procedure in the following ways:

   - During agent creation click **Test Log File Settings** on the **Log File Information** page.
   - After agent creation, select an attribute group on the Agent Editor **Data Source Definition** page and click **Test Log File Settings**. For more information about the Agent Editor, see "Using the Agent Editor to modify the agent" on page 1152.

   After you click **Test Log File Settings** in one of the previous two steps, the **Parse Log** window opens.

2. Select the source of the log data for testing:

   - **Use attribute group settings**: use the file name and location specified in the data source. By default, the data source processes only the information that is added to the log file after the testing process is started. You can use this option if the log file is being updated in real time.
   - **Specify a sample file**: provide a sample log file. With this setting, the testing procedure parses the entire contents of the log file. With this option, you can test the data source and create the attributes for it immediately, based on an existing sample. Specify the path and name of the file in the **Log file name** field or use the **Browse** button to select the file.

3. Optional: Before you start testing, you can set environment variables and configuration properties.

   For more information, see ("Attribute group testing" on page 1347).

4. Click **Start Agent**.

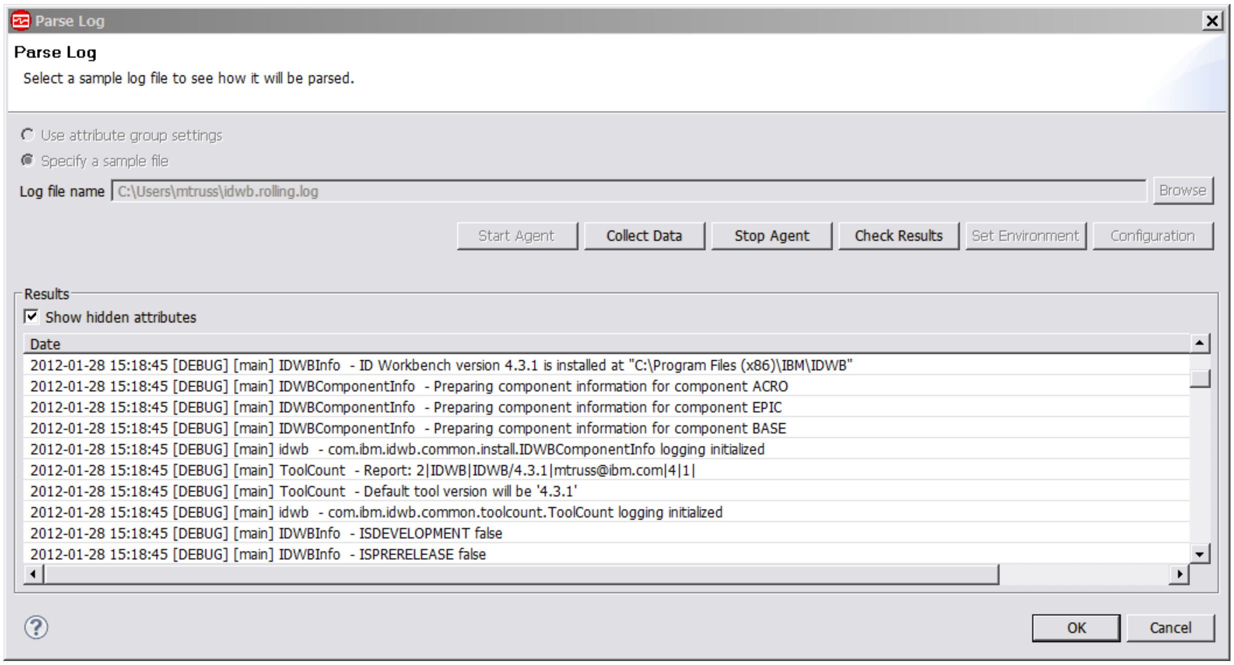
   A window opens indicating that the Agent is starting. When the agent starts, it monitors the configured log file for new records

5. To test your agent's data collection, generate new records in the monitored log file.

   When new records are added to the log file, the agent parses them according to its configuration and updates the corresponding attribute values in its cache.

6. To simulate a request from Tivoli Enterprise Portal or SOAP for agent data, click **Collect Data**.

   The **Parse Log** window collects and shows any new attribute values in the agent's cache since it was last started. An example data collection is shown in Figure 42 on page 1245



Select a sample log file to see how it will be parsed.

○ Use attribute group settings

● Specify a sample file

Log file name C:\Users\mtruss\idwb.rolling.log

☑ Show hidden attributes

```
2012-01-28 15:18:45 [DEBUG] [main] IDWBInfo  - ID Workbench version 4.3.1 is installed at "C:\Program Files (x86)\IBM\IDWB"
2012-01-28 15:18:45 [DEBUG] [main] IDWBComponentInfo  - Preparing component information for component ACRO
2012-01-28 15:18:45 [DEBUG] [main] IDWBComponentInfo  - Preparing component information for component EPIC
2012-01-28 15:18:45 [DEBUG] [main] IDWBComponentInfo  - Preparing component information for component BASE
2012-01-28 15:18:45 [DEBUG] [main] idwb  - com.ibm.idwb.common.install.IDWBComponentInfo logging initialized
2012-01-28 15:18:45 [DEBUG] [main] ToolCount  - Report: 2|IDWB|IDWB/4.3.1|mtruss@ibm.com|4|1|
2012-01-28 15:18:45 [DEBUG] [main] ToolCount  - Default tool version will be '4.3.1'
2012-01-28 15:18:45 [DEBUG] [main] idwb  - com.ibm.idwb.common.toolcount.ToolCount logging initialized
2012-01-28 15:18:45 [DEBUG] [main] IDWBInfo  - ISDEVELOPMENT false
2012-01-28 15:18:45 [DEBUG] [main] IDWBInfo  - ISPRERELEASE false
```

   Figure 42. Parse Log window that shows parsed log file attribute values

7. Optional: Click **Check Results** if the returned data is not as you expected.

   The **Data Collection Status** window opens and shows you more information about the data. The data that is collected and shown by the Data collection Status window is described in "Performance Object Status node" on page 1388

8. The agent can be stopped by clicking **Stop Agent**.
9. Click **OK** or **Cancel** to exit the **Parse Log** window. Clicking **OK** saves any changes that you made.

Related concepts

“Testing your agent in Agent Builder” on page 1346
After you use Agent Builder to create an agent, you can test the agent in Agent Builder.

Monitoring an AIX Binary Log

You can define a data source to monitor AIX binary error logs through the errpt command. You can also configure it to filter and summarize the data. The resulting events are placed in a data set.

About this task

Log Monitoring supports the monitoring of AIX binary error logs through the errpt command. The errpt command generates an error report from entries in an error log. It includes flags for selecting errors that match specific criteria. This support for the monitoring of AIX binary error logs through the errpt command is modeled on the support for the same function in the Tivoli Monitoring UNIX Logs Agent (product code kul or ul).

When you supply the Agent Builder with an **errpt** command string, it processes the events that result from running this command. Agent Builder enforces the same constraints on this command that the Monitoring Agent for UNIX Logs does. In particular, you must use the **-c** (concurrent mode) option so that the command runs continuously, and you cannot use the **-t** option or the following options that result in detailed output: **-a**, **-A**, or **-g**.

An Agent Builder agent that monitors the AIX **errpt** command automatically includes the same information as a Monitoring Agent for UNIX Logs does. For more information about the attribute groups for AIX binary error logs, see “AIX Binary Log attribute group” on page 1401.

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Logged data** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **AIX Binary Log**.
3. Click **Next**.
4. On the **Binary Log Information** page, enter an errpt command.

   The default value is:

   ```
   errpt -c -smmddhhmmyy
   ```

   The agent searches for the 'mmddhhmmyy' string and replaces it with the actual date and time on startup. Only the first occurrence of the string is replaced.

   You can supply your own errpt command but Agent Builder enforces the same constraints on this command that the Monitoring Agent for UNIX Logs does. In particular, you must use the**-c** (concurrent mode) option so that the command runs continuously, and you cannot use the **-t** option or the following options that result in detailed output: **-a**, **-A**, or **-g**.
5. (Optional) Click **Advanced** to select filtering and summarization options for events. For more information, see “Controlling duplicate events” on page 1373.
6. Do one of the following steps:
   - If you are using the **Agent** wizard, click **Next**.
   - Click **Finish** to save the data source and open the Agent Editor.

Related reference

“AIX Binary Log attribute group” on page 1401

The AIX Binary Log attribute group displays events from the AIX Binary Log as selected by the provided `errpt` command string.

Monitoring a Windows Event Log

You can define a data source to collect data from a Windows event log. You can configure it to filter the data. The resulting events are placed in the Event Log data set.

About this task

You can collect data from the Windows event log by using the type, source, or ID of events. You use these parameters to filter the log events that the Windows system gathered. The agent compares each new event in the monitored event log against the specified filter. If the event matches one of the event types, event sources, and event IDs specified in the filter, it passes.

For example, if the Event log filter is for the Application log, specify **Error** as the event type. This choice matches all events that are logged to the Application log with an event type value of `error`. If you add the **Diskeeper** and **Symantec AntiVirus** event sources, the agent matches all error events from either of these sources. You can add specific event IDs to refine the filter further. No direct association exists between the event type, event source, and event ID. If one of the values for each matches an event, the event matches.

By default, only events that are generated after the agent starts are processed. However, you can enable the agent when it restarts to process log events that are generated while the agent is shut down. For more information about enabling the agent to process events generated while the agent is shut down, see step "6" on page 1247.

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Logged Data** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **Windows Event Log**.
3. Click **Next**.
4. On the **Windows Event Log** page, select the name from one of the logs in the **Windows Event Log name** list, or type a name for the event log.

   The list is constructed from the set of logs on the current system, for example:

   ```
   Application
   Security
   System
   ```
5. In the **Windows Event Log** page, specify whether you want to filter the results by using one or more of the following mechanisms:

   - "Filtering by event type" on page 1248
   - "Filtering by event source" on page 1249
   - "Filtering by event identifier" on page 1249

   Note: You must select at least one of these filter criteria.
6. To process log events that are generated while the agent is shut down, on an agent restart, click **Offline Event Settings** on the **Windows Event Log** page.

   The **Windows Event Log Bookmark Settings** window opens.
7. Select one of the following bookmarking options:

   Note: These options apply to all Windows event logs being monitored.

   - **Do not collect any offline events**: Events that are generated while the agent is shut down are not processed. This option is the default option.

- **Collect all offline objects**: All events that are generated while the agent is shut down are processed.
- **Specify custom collection settings**: You can enter a value to throttle the processing of old events that are based on a time value, or number of events, or both. By using this option, you ensure that the monitoring environment is not overloaded with events when the agent starts.

  For example, if 100 is entered in **The maximum number of events to collect** field and 30 is entered in the **Restrict collection based on a time interval (in seconds)** field. The number of events that are processed is either the last 100 events that are generated before the agent starts, or any event that is generated within 30 seconds of agent start. Which result depends on the variable that is matched first.

  When you enter a value for the maximum number of events to collect, the *CDP_DP_EVENT_LOG_MAX_ BACKLOG_EVENTS* environment variable is added. When you enter a value to restrict collection that is based on a time interval, the *CDP_DP_EVENT_LOG_MAX_BACKLOG_TIME* environment variable is added. When either or both of these variables are added, the `*eventlogname_productcode_instancename_subnodename*.rst` file is created containing the last event record that is processed for the event log. This file is in the `*%CANDLE_HOME%*\tmaitm6\logs` directory and is used when the agent is restarted to process old events that are generated while the agent was shut down.

8. If you want to set global options for the data source, click **Global Options** on the **Windows Event Log** page

   The **Global Windows Data Source Options** window opens.
9. Select the **Include remote Windows configuration properties** check box if you want to include this option, and click **OK**.

   For information about Windows remote connection configuration for Windows data sources, see "Configuring a Windows remote connection" on page 1333.
10. After you specify the filter and click **OK**, on the **Windows Event Log** page, do one of the following steps:
    - If you are using the **Agent** wizard, click **Next**.
    - Click **Finish** to save the data source and open the Agent Editor. The name of the new Windows Event Log is shown on the **Agent Editor Data Source Definition** page.

What to do next

For information about Windows remote connection configuration for Windows Event Log data sources, see "Configuring a Windows remote connection" on page 1333.

Filtering by event type

Filter Windows Event Log results by event type

Procedure

1. In the **Windows Event Log** page, select **Filter by event type**.
2. Select one or more of the following Event types:
   - **Information**
   - **Warning**
   - **Error**
   - **Success Audit**
   - **Failure Audit**
3. Click **Finish** to complete.

Filtering by event source

Filter Windows Event Log results by event source

Procedure

1. Select **Filter by event source** and click **Add** in the **Event sources** area of the **Windows Event Log** page.

   The **Event Source** window opens.
2. Make one of the following choices.
   - Type the event source name and click **OK**.
   - Click Browse **Browse** to find and select an event source from a list and click **OK**.

     The name that you selected is shown in the **Event Source** window.

     Note:

     a. To sort the list of event sources, click the column heading.
     b. To refresh the information in the window, click the **Refresh** icon.
     c. To search for specific event sources, click the **Search** (binoculars) icon.
3. Click **OK** to see the new event source filter in the Event sources list in the **Windows Event Log** window.

Filtering by event identifier

For the Windows Event Log data source, you can filter events by event identifier.

About this task

To filter by event identifier, use the following procedure:

Procedure

1. Select **Filter by event identifier** and click **Add** in the **Event identifiers** area of the **Windows Event Log** window.

   The **Event Identifier** window is displayed.
2. If you know that you want to monitor specific events from an application, specify the numbers of the event as the application defines it. Type an integer as the event identifier and click **OK**.

   The new numeric event identifier filter is displayed in the Event identifiers list in the **Windows Event Log**.

   Note: Each event identifier must be defined individually.
3. If you want to modify a Windows event log, select it and click **Edit**.
4. If you want to delete a Windows event log, select it and click **Remove**.
5. You can add more event logs to the list, or click **Finish**.

Monitoring a command return code

You can define a data source to monitor an application or system by using a *command return code*. The agent runs the command, collects the return code, and adds the result to the Availability data set.

About this task

A user-created script, executable file, query, or system command can return a code. A command return code is an application-specific mechanism for determining whether the application or monitored system is available. The agent runs the specified command and determines the state of the application or monitored system by examining the return code.

The command must present a unique return code for each descriptive state. The command must also define a message to be used by the agent for each of these return codes. The command can use environment and configuration variables within the user created script, executable file, query, or system command. The command must not use environment or configuration variables on the command-line invocation of the command, with only the following exceptions available: *AGENT_BIN_DIR*, *AGENT_ETC_DIR*, *AGENT_LIB_DIR*, *CANDLE_HOME*, and *CANDLEHOME*.

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, select **Command or script** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **A command return code**.
3. Click **Next**.
4. On the **Command Return Code** page, **Command return code information** area, type the display name.
5. Use the following substeps to define and describe command lines that you want your command return code to use.

   Note: Define a command for each operating system that is supported by the agent. Commands can be shared, but the total set of operating systems for all of the commands must equal the set of agent supported operating systems.

   a) Click **Add** in the **Commands** area of the **Command Return Code** window to open the **Command Information** window.

   b) Type a command line and select an operating system from the list in the **Operating Systems** area of the **Command Information** window.

      Note:

      1) For a Windows command, you must type the full name of the command. For example, `command_to_run.bat` and not just `command_to_run`.
      2) Place quotation marks around the name so that it is not parsed by the command interpreter. For example, type `"this is a test.bat"argument` and not `this is a test.bat argument`.
      3) You can click a command and click **Edit** to modify it, or click **Remove** to delete it.

   c) Click **Add** in the **Return Codes** area of the **Command Information** window.

   d) Select a return code type from the list that is shown in the **Return Code Definition** window

      You can assign the following states to the test return codes:

      - `ALREADY_RUNNING`
      - `DEPENDENT_NOT_RUNNING`
      - `GENERAL_ERROR`
      - `NOT_RUNNING`
      - `OK`
      - `PREREQ_NOT_RUNNING`
      - `WARNING`

   e) Type a numeric value for the return code type that you selected.

      The return code value is an integer that specifies a defined return code for the command return code. For portability between operating systems, use a return code value of `0 - 255`. For a command that runs only on Windows, the return code value can be `-2147483648 - 2147483647`.

   f) Define a message for each return code so that the message and code can be shown together. Click **Browse** to set up the message text.

      The message window lists messages that are defined in the agent. The **Messages** (list) window opens.

Note:

1) You can select text that was entered previously by selecting it in the list of message texts instead of clicking **Browse**. Then, continue to Step 5k.
2) Until you define messages, the list remains blank. You can use **Edit** to alter a defined message and **Remove** to delete one or more messages that you defined.

g) In the **Messages** (list) window, click **Add**

The **Message Definition** window opens.

Note: The message identifier is automatically generated for you.

h) Enter some text that describes the meaning of the new message in the **Message text** field.

i) Click **OK**.

The **Messages** (list) window opens showing the new message .

j) To verify the message and make it permanent, select it in the list and click **OK**.

The new return code type, value, and text are shown in the **Return Code Definition** window.

k) If you want this return code to be available to other commands on other operating systems for this command return code, select **Global return code applies to all commands**. If you want this return code to be available only to this command, leave **Local return code applies only to this command** selected.

l) Click **OK** in the **Return Code Definition** window.

m) Define at least two return codes before you leave the **Command Information** window. One return code to indicate no problems with the availability, another to indicate whether a problem occurred. If you want to add another return code, return to step c.

n) Optional: In the **Command Information** window, **Command files** area, click **Add** if you want to select one or more scripts or executable files for the agent to run.

The file or files are copied into the project folder of the agent under `scripts/`*`operating system`*, where *operating system* is a variable that depends on what you selected in the **Operating Systems** area of the **Command Information** window. These files are also packaged and distributed with the agent. To edit the definition of an existing command file, or the original command file since copied into the project, select the file and click **Edit**. See ("Editing a command file definition" on page 1253).

o) Click **OK** in the **Command Information** window.

Note: The command files table is where you define any external files that you want to include in the agent package. These files are copied into the project directory and packaged with the agent for distribution.

6. If you have other return codes that are not already defined, define and describe global return codes that your command return code can use.

a) Click **Add** in the **Global return codes** area of the **Command Return Code** page.

Note: The return codes that are defined here are global. This means that the return codes are appropriate for all of the commands that are defined for the command return code. (They are not shared between command return codes). In addition, you can define return codes when you enter the command information. The return codes that are defined here can be global or local. Local return codes are only appropriate for this specific command. This hierarchy is useful if you have a return code that is the same across all operating systems. (For instance, a return code of 0 means that everything is functioning correctly. You can define it at the global level, and then all defined commands interpret 0 in this way.) If none of the other operating systems return a 5, you can define the return code of 5 only for the Windows command. If you define a return code at the local command level that is already defined at the global level, the command level is used. You can use this method to override return codes on specific operating systems. For instance, if on all UNIX operating systems, a return code of 2 means one thing, but, on Windows, it means something different. You can define a return code of 2 at the global level as expected by the UNIX operating systems. Then, in the command for Windows, you can redefine return code 2 for the meaning on Windows.

b) Select a return code type from the list that is shown in the **Return Code Definition** window.

   You can assign the following states to the test return codes:

   - `ALREADY_RUNNING`
   - `DEPENDENT_NOT_RUNNING`
   - `GENERAL_ERROR`
   - `NOT_RUNNING`
   - `OK`
   - `PREREQ_NOT_RUNNING`
   - `WARNING`

c) Type a numeric value for the return code type that you selected. The return code value is an integer that specifies a defined return code for the command return code.

d) Click **Browse** to set up the message text and its associated meaning. You must define a message for each return code so that the message and code are shown together.

   The **Messages** window lists messages that are defined in the agent.

   Note:

   1) Until you define messages, the list remains blank. You can use **Edit** to alter a defined message and **Remove** to delete one or more messages you defined.
   2) You can select text that was entered previously by selecting it in the **Message text** list instead of clicking **Browse**. Then, continue to Step 6h.

e) In the **Messages** (list) window, click **Add** to see a **Message Definition** window, where you can type text that describes the meaning of the new message.

f) Click **OK**.

g) The **Messages** (list) window opens with the new message. To verify the message and make it permanent, select it in the list and click **OK**.

h) When the new text, type, and value are shown in the **Return Code Definition** window, click **OK**.

i) On the Command Return code page, when you finish defining the return codes and commands for all supported operating systems, do one of the following steps:

   - If you are using the New Agent wizard, click **Next** or click **Finish** to save the data source and open the Agent Editor.
   - If you are using the New Agent Component wizard, click **Finish** to return to the Agent Editor.

What to do next

If you want to use the data from this data source in the summary dashboard for IBM Cloud Application Performance Management, you must create a filtered data set (attribute group) based on the Availability data set and configure it as providing a single row. Use the NAME field to select the row for your process.

In the new filtered attribute group, select the Status field and specify the severity values for it.

For instructions, see:

- "Creating a filtered attribute group" on page 1311
- "Specifying severity for an attribute used as a status indicator" on page 1177
- "Preparing the agent for Cloud APM" on page 1342

Editing a command file definition

You can change the command file that is imported into the project, or import changes to the existing command file into the project.

Procedure

1. Select the file in the **Command files** area of the **Command Information window**.
2. Click **Edit** to open the **Import Command File** window.

   From the **Import Command File** window, you can get the status of the command file. You can also change the location of the original source file, and recopy the source file into the agent.
3. Choose one of the following steps:

   - Click **OK** to schedule a copy of the file to occur the next time that the agent is saved.
   - Click **Copy Immediately** to copy the file without first saving the agent.

   Note: The **Copy Immediately** option is not available when you access the **Import Command File** window from the New Agent wizard.

File Separation & Consolidation

You can use the Separate and Consolidate functions to move files in and out of operating system-specific folders in the agent.

When a file is first added to the agent, a single copy is added in the `scripts/all_windows` folder, the `scripts/all_unix` folder, or the `scripts/common` folder. The `scripts/common` folder is used if the file is used on both Windows and UNIX.

To place different copies of the file on different operating systems (for example, a binary executable file), click **Edit** and click **Separate**. The file is removed from the common folder and copied into operating system-specific folders. Then, you can replace individual copies of the file with ones appropriate for specific operating systems.

Note: Java resource files must remain in the `scripts/common` folder. You cannot click **Separate** to make separate copies of Java resource files for individual operating systems.

If you separated the files into operating-system-folders, you can use **Consolidate** to move them back into a common folder. If you created the agent in an Agent Builder version that did not support common folders, use **Consolidate** to move them back into a common folder. If any of the copies of the file differ from one another, you are prompted to select the file to use as the common file. All other copies are discarded.

Monitor output from a script

You can define a data source to collect data from a script or external program. Use it when application data is not available through a standard management interface or when you need to provide a summary of multi-row data in a single row. The agent runs the script and collects its output. Each line in the script output is parsed into a row of the resulting data set.

Data can be collected from either a local or remote system. The output of the script or program must contain only values for each attribute within the attribute group. To return multiple rows of data, the data for each row must be separated by a line break. The attributes in each row of data are separated by the separators you define. For more information about separators, see "Script parsing and separators" on page 1254

The command can use environment and configuration variables within the user-created script, executable file, query, or system command. The command cannot use environment or configuration variables on the command-line invocation of the command, with only the following exceptions available: `AGENT_BIN_DIR`, `AGENT_ETC_DIR`, `AGENT_LIB_DIR`, `CANDLE_HOME`, and `CANDLEHOME`.

The agent monitors script output that is written by using the same locale and code page that the agent runs in.

Collecting script data from a remote system

To collect script or program data from a remote system, Agent Builder uses a Secure Shell (SSH)

To collect data from a remote system, Agent Builder creates a Secure Shell (SSH) session and starts the script or external program on the remote system. The agent establishes and logs on to an SSH session. The agent then uploads the scripts to the remote system, starts the script or external program, and retrieves the output. The agent can be configured to keep the session open or reestablish the session for each invocation. If the session is kept open, the script can be reused or uploaded for each invocation. By default, a single SSH session is used and the scripts are reused for each invocation.

Agent Builder supports use of only SSH Protocol Version 2 with Rivest, Shamir, and Adleman (RSA) or Digital Signature Algorithm (DSA) keys. The agent is either authenticated by user name and password, or by public key authentication. The generation and distribution of the public keys is an administrative task that must be done outside of the agent and Agent Builder.

To run a Take Action command that is written against a Secure Shell (SSH) enabled script data provider on the remote system, see "SSHEXEC action" on page 1471.

Restriction: If your agent was built with an Agent Builder version before 6.3 and it has a script data provider that uses SSH, the provider fails when run with IBM Tivoli Monitoring version 6.3 or later. To resolve this issue, rebuild the agent with the current version of Agent Builder.

The restriction is because IBM Tivoli Monitoring version 6.3 uses a newer version of the Global Secure ToolKit (GSKit) API. You must rebuild the agent with Agent Builder 6.3 or later to run it with IBM Tivoli Monitoring version 6.3 or later. If you build the agent with Agent Builder 6.3, it can also run with earlier versions of IBM Tivoli Monitoring.

Script parsing and separators

You can change and assign specific script separators to one or more attributes.

When you create a script attribute group, a single character text separator is by default assigned. The default separator is ";". The separator is used by the agent to parse and delimit the data for each attribute in the data row. You can change the default separator to use a different character. You can also assign specific separators to one or more individual attributes.

You can assign specific separators for individual attributes that:

- Take a fixed number of bytes from the output.
- Separate one attribute from the next with a custom separator, which can be more than one character.
- Delimit an attribute value with a string at the beginning and end of the value.
- Return the rest of the text as the attribute value (whether it contains embedded separators or not).

You can use one or more of these separators to extract attribute values from the data rows.

Example 1 - Simple script output

Some scripts can output data rows with clear and regular separators, for example:

```
Row One;1;2
Row Two;3;4
Row Three;5;6
```

Here the ";" character is a clear and regular separator between the three pieces of data on each row. In this case, the default separator is fine, so there is no need to change or define other separators. It is not difficult to imagine a similar script output where the separator is a different character, as in the following example.

```
Row One-1-2
Row Two-3-4
Row Three-5-6
```

In this example the separator is changed from a ";" character to a "-" character. In this case when you define the attributes, change the default separator to use the "-" character.

Example 2 - Complex script output

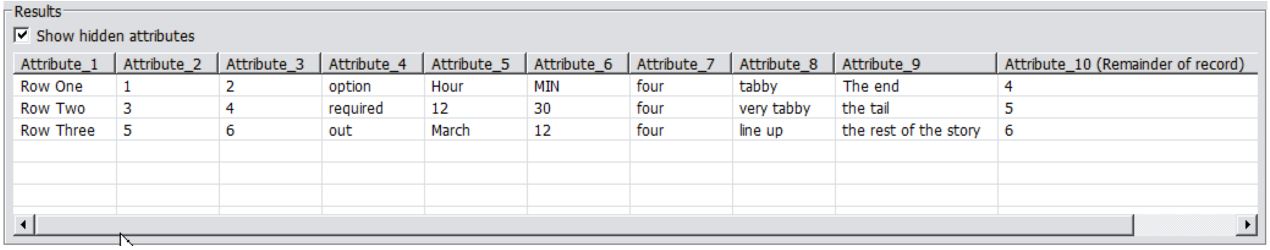
Some scripts can output data rows that have irregular or changing separators, for example:

```
Row One;1;2;[option]Hour:MIN;fourtabby     The end;4
Row Two;3;4;[required]12:30;fourvery tabby the tail;5
Row Three;5;6;[out]March:12;fourline up    the rest of the story;6
```

In this example an assignment of separators to attribute definitions that you can use is:

1. Initially the default separator ";" is fine for the first three attributes in each data row. In this case, you assign the separator type **Separator Text** set to ";" when you define each attribute, this setting is the default one.
2. For the fourth attribute, assume the string between the"[" and "]" is a value that you want to extract. In this case when you define the fourth attribute, you assign a separator type **Begin and End Text** with begin and end text values of "[" and "]".
3. For the fifth attribute, assume that you want to extract the values between the "]" and ":" characters. In this case when you define the fifth attribute, you assign separator type **Separator Text** set to ":".
4. For the sixth attribute, the default separator ";" is fine again, accept the default.
5. For the seventh attribute, you would like to extract the string in the next four characters "four". There is not a clear separator at the end of this string. You can assign a number of characters to define the separation from the next attribute. You assign a separator type **Number of characters**, and specify four characters as the length.
6. For the eighth attribute you would like to extract the strings tabby, very tabby and line up. In this case, you can assume that all of these strings are followed by a tab character. In this case, you assign a separator of type **Tab separator**.
7. For the ninth attribute, you revert again to the default separator type to extract the remaining text to this attribute.
8. For the 10th attribute, you specify **Remainder of record** to assign the remainder of the data row to this attribute

Defining these separators on a script that outputs the data rows that are shown earlier in this example is shown in the following output:



Results

☑ Show hidden attributes

Attribute_1	Attribute_2	Attribute_3	Attribute_4	Attribute_5	Attribute_6	Attribute_7	Attribute_8	Attribute_9	Attribute_10 (Remainder of record)
Row One	1	2	option	Hour	MIN	four	tabby	The end	4
Row Two	3	4	required	12	30	four	very tabby	the tail	5
Row Three	5	6	out	March	12	four	line up	the rest of the story	6

Figure 43. Example attribute value output when Agent parses complex script output.

The procedure to define the attribute separators is described under step "10" on page 1257 of "Steps for monitoring output from a script" on page 1255.

Steps for monitoring output from a script

Configure your agent to receive data from a script data source.

Before you begin

See "Monitor output from a script" on page 1253

About this task

Use the following procedure to monitor output from a script:

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, select the option **Command or script** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **Output from a script**.
3. Click **Next**.
4. On the **Command List** page , click **Add** to display a **Command Information** window.

   Note: Selecting the **Enable data collection using SSH** check box enables SSH for this attribute group. If this check box is not selected, the attribute group runs locally.

   Note: If a command exists that can be run on the operating system on which the Agent Builder is running, the **Test** option is enabled. You can use **Test** to test a command that you defined.
5. In the **Command Information** area in the **Command Information** window, type a command name with the necessary arguments in the **Command** field, and a separator in the **Separator** field.

   Note:

   a. Scripts in Windows are frequently started without specifying the `.bat` or `.cmd` extension on the command line. For remote execution, a shell environment must be installed and you must specify the `.bat` or `.cmd` in the script data source command for the script to run. `Cygwin` is an example of a shell environment that is available for Windows. Linux, Red Hat, and AIX. To verify that a shell environment exists, SSH or log on to the remote host and enter the command:

      ```
      PATH=$PATH:. <command>
      ```

      If the command runs, then a shell environment exists.

   b. Use quotation marks around the name so that it is not parsed by the command interpreter. For example, `this is a test.bat argument` becomes:

      ```
      "this is a test.bat" argument
      ```

   c. Environment variables and configuration variables can be used in the user-provided script, but cannot be part of the command line that starts the script. The following variables are exceptions to this rule:

      AGENT_BIN_DIR
      The directory where the agent places binary files or scripts

      AGENT_ETC_DIR
      The directory where the agent places configuration files

      AGENT_LIB_DIR
      The directory where the agent places shared libraries or dynamic-link libraries

      CANDLEHOME
      The Linux or UNIX Tivoli Monitoring installation directory

      CANDLE_HOME
      The Windows Tivoli Monitoring installation directory

   d. If the SSH data collection option is being used, the command line is run relative to the user's home directory on the remote system. If you are uploading scripts or executables to the remote system, they are copied to the location specified in the agent's environment variable *CDP_SSH_TEMP_DIRECTORY*. The location defaults to the user's home directory on the remote system. On some systems, you might need to define the command line with a relative path, such as `./Script.sh`.
6. In the **Operating Systems** area, select one or more operating systems. When you collect data from a remote system by using SSH, Operating Systems is a property of the system on which the agent is installed. It is not the Operating System of the remote system. It is advised that you select the **All operating systems** check box when you use the SSH data collection features.
7. Optional: If one or more user-defined files are necessary to run the command, click **Add** in the Command files area to specify the files from your system.

The files are copied into the project folder of the agent under `scripts/`*`operating system`*, where *operating system* is a variable that depends on what you selected in the **Command Information** window. These files are also packaged and distributed with the agent. If you want to edit the definition of a command file you already added, or changed the contents of, select the file and click **Edit**. See "Editing a command file definition" on page 1253.

8. Click **OK**. The **Command List** page is displayed.
9. To test the command, use the following steps:

   a) Click **Test** to open the command information and display the **Test Command** window. To test the script on a remote system, select a system from the **Connection name** list or click **Add** to add the host name of a system.

   b) Use the **Test Command** window to change the command, default separator, and attribute separators, and to view how these changes affect the data that is returned.

      1) Type the command and separator in the fields if they are not already entered.

         Note: You can specify other separators by using the **Attribute Information** window at attribute creation time or by using the Agent Editor to modify an existing attribute. For more information about the Agent Editor, see "Using the Agent Editor to modify the agent" on page 1152 and for more information about manipulating data source and attributes, see "Editing data source and attribute properties" on page 1168

      2) Before you start testing, you can set environment variables and configuration properties. For more information, see ("Attribute group testing" on page 1347).
      3) Click **OK** to return to the **Test Settings** window.
      4) Click **Start Agent**. A window indicates that the Agent is starting.
      5) To simulate a request from Tivoli Enterprise Portal or SOAP for agent data, click **Collect Data**. The Agent Builder runs your command. If you specified a remote system, provide a user ID and password. Even if the return code is not 0, the Agent Builder parses the results of the command in the same way the agent does.
      6) The **Test Settings** window collects and displays any data in the agent's cache since it was last started. The initial names of the attributes are **Attribute_1, Attribute_2**, and so on; however, you can modify the properties of the attributes by clicking the appropriate column heading.
      7) Click **Check Results** to view the return code from the command, the unparsed data, and any error messages that were returned.
      8) The agent can be stopped by clicking **Stop Agent**.
      9) Click **OK** to return to the **Command Information** window.

         If you change the command or the separator, the appropriate command is updated to reflect those changes.

         If this window was opened when you created the script data source, the attributes were added to the new script data source.

         If this window was opened from an existing script data source, then any changes to the attributes are made to the script data source. Any additional attributes are added, but any extra attributes are not removed. These options affect only the attributes that are parsed from the script output. Any derived attributes are not affected. If any of these attributes become invalid based on the attributes they reference, you can update or remove derived attributes manually. The derived attribute formula is displayed and not the actual result value.

   Note: If the attribute group exists, to start a test, complete the following procedure

   a. Select the attribute group on the **Agent Editor Data Sources Definition** page.
   b. Select the script to be tested from the Command List
   c. Click **Test** and follow the procedure at step "9" on page 1257

10. If you skipped testing the command in step ("9" on page 1257), use the following steps:

    a) On the **Command List** page with the completed command information, click **Next**.

b) On the **Attribute Information** page, complete the attribute name and type information by using (Table 264 on page 1173). Select **Add additional attributes** to add further attributes

c) On the **Attribute Information** page, use the **Script Attribute Information** tab to choose a specific data separator for this attribute.

The standard separator `;` is selected by default. You can choose a number of other separators such as a string, a number of characters, a tab, or a space. You can also choose to use a different string separator for the beginning and end of the data. Finally, you can also choose **Remainder of record** to assign the remainder of the record to the attribute. For more information about script parsing and separators, see "Script parsing and separators" on page 1254.

11. Do one of the following steps:
    - If you are using the **Agent** wizard, click **Next**.
    - Click **Finish** to save the data source and open the Agent Editor.
12. You can add attributes and supply the information for them. For more information, see "Creating attributes" on page 1169.

    In addition to the fields applicable to all data sources (described in "Fields and options for defining attributes" on page 1172), the **Data Sources Definition** page for the Script data source has the following options:

    Command List
    Provides access to the commands and scripts to start during data collection.

    Add
    Allows the user to add a command to be started by this attribute group.

    Edit
    Allows the user to edit an existing command entry.

    Remove
    Allows the user to delete an existing command entry.

    Test
    Allows the user to access the test environment for this attribute group.

    Enable data collection using SSH
    Selecting this check box enables SSH for this attribute group. If this check box is not selected, the attribute group runs locally.

    For information about SSH remote connection configuration for script data sources, see "Configuring a Secure Shell (SSH) remote connection" on page 1336.

Monitoring data from Java Database Connectivity (JDBC)

You can define a data source to receive data from a JDBC database. The agent runs an SQL query to collect data from the database. Each column that is returned by the query is an attribute in the resulting data set.

About this task

The JDBC data provider supports the following database servers:

- IBM DB2 9.x and 8.x
- Microsoft SQL Server 2008, 2005, and 2000
- Oracle database 11g and 10g

Agent Builder does not include the JDBC drivers for these databases. The JDBC drivers are a set of JAR files that are provided by the vendor that are necessary to establish a JDBC connection to the database. For convenience, here are links to where those drivers can be downloaded:

- IBM DB2: JDBC drivers are included with the database server installation in a subdirectory named `java` located under the main DB2 installation directory.
- Microsoft SQL Server website at www.microsoft.com

- Oracle database: Oracle Database JDBC (http://www.oracle.com/technetwork/database/features/jdbc/index.html)

Note: An important thing to remember is that the JDBC data provider can remotely monitor your Database servers. A Java runtime environment and JDBC driver JAR files for the database server you are connecting to must be on the system where the agent runs.

The following versions of Java are supported:

- Oracle Corporation Java Version 5 or later
- IBM Corporation Java Version 5 or later

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Data from a server** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **JDBC**.
3. Click **Next**.
4. On the **JDBC Information** area in the **JDBC Information** page, click **Browse** to connect to a database and build your SQL Query.

   Use the JDBC Browser to connect to a database and view its tables so you can build an SQL query that collects the data you need. When you select a table and columns, a query is generated for you and attributes are added for each of the columns returned by the query. You can modify and test the query that is generated to make sure the data that is returned is what you need.

   Note: You can also manually create the JDBC data source without clicking **Browse**. If you want to manually create the data source, specify the query and click **Next**. You must define an attribute for each column returned by the query, in the order that the columns are returned.

   With the JDBC data provider, you can run SQL queries and stored procedures against a database to collect monitoring data. When you specify an SQL query to collect data, you can include a where clause in your SQL statement to filter the data that is returned. The SQL statement can also join data from multiple tables. In addition to SQL select statements, the JDBC data provider can run stored procedures. For information about running stored procedures, see "Stored procedures" on page 1263.
5. The first time the Browser opens, the Java Database Connectivity (JDBC) Browser window indicates that no connections are selected. You must add a connection. Click **Add** and follow the Steps to add a connection.

   If you already defined a connection, that connection is used and you can proceed to Step "6" on page 1260.

   Note: The **Status** field shows the status of the current connection.

   Use the following steps to add a connection:

   a) On the **JDBC Connections** page , click **JDBC Connection**, and click **Next**.

   b) On the **Connection Properties** page, complete the fields as follows:

      Connection Name
      Name of the JDBC connection. Type a unique name for this connection. You use this name to reference the connection in the browser.

      Database Type
      Type of database. Select the database product to which you are connecting. For example, to connect to the IBM DB2 database, select **DB2**.

      User Name
      Must be defined with at least read access to the database, but does not have to be the database administrator

Password
Must be defined with at least read access to the database, but does not have to be the database administrator

Host name
Host name on which the database server is running. With JDBC, you can monitor remote databases so you are not restricted to monitoring databases on the local system.

Port
Port on the host name on which the database server is listening.

Database
Name of the database to which to connect.

Jar Directory
Directory containing the JDBC JAR files used to connect to the database. Type the path name, or click **Browse** to locate the directory.

c) Optional: Select the **Save the password in the Agent Builder workspace** check box if you want to save the password for this connection.

d) Optional: Select the **Set as agent configuration defaults** check box if you want the defaults for this application server type to be copied from these properties.

   If you are building the agent on a system that is similar to your monitored systems, it is advisable to check this box. If you do not check this box, the user who configures the agent sees an empty field. The user must then determine the values for all of the information without default values.

e) Click **Test Connection** to create a connection to the database that uses the configuration parameters you specified.

   A message on the **Connection Properties** page indicates whether the connection succeeds.

f) When you have a working connection, click **Finish**.

6. In the **Java Database Connectivity (JDBC) Browser** window, a connection is made to the configured database. The tables that are contained in the database are shown in the **Database Tables** area. Select a database table to see the columns that are contained in that table in the **Columns in the selected table** area.

   Note:

   a. Click the binoculars icon to search for a table in the **Database Tables** list.
   b. All tables are shown by default. You can filter the tables that are shown by selecting a different filter option. The available filter options are shown in Table 270 on page 1260.

   Table 270. Filter options

Filter option	Description
All	Show all tables
User	Show only user tables
System	Show only system tables
View	Show only database views

   Note: If you want to retrieve specific columns, select only these columns. If you select the table, Agent Builder automatically builds a query that gathers all of the columns from the table and creates attributes for all the columns that are currently in the table.

   You can select columns in the following ways:

   - Select the table and get the default query for all columns.
   - Select columns to get only those columns.

7. Optional: Modify the enumeration values that are set for Error, Missing data, and No value in the **Attribute Information** page.

Modify the values to avoid any overlap with legitimate values that might be returned from database table columns.

8. Optional: Click **Test** on the **Java Database Connectivity (JDBC) Browser** window to test and modify the SQL statement.

   The **Run the SQL statement** window opens.

   a) Enter or modify the SQL statement in the **SQL statement** field.

   b) Click **Run** to run the SQL statement.

      The results are displayed in the **Results** area. Continue to modify and test the statement until you are satisfied with the data that is returned.

   c) Click **OK** to save the statement, create the correct attributes, and return to the **JDBC Information** window.

9. Optional: Click **Test** on the **JDBC Information** window to test the attribute group in a more realistic agent environment. For more information about testing JDBC attribute groups, see "Testing JDBC attribute groups" on page 1265. If you change the JDBC statement during this test, you must also adjust the attributes so that there is one attribute per column returned by the JDBC statement, in the correct order.

10. Optional: You can create a filter to limit the data that is returned by this attribute group by clicking **Advanced**. For more information about filtering data from an attribute group, see "Filtering attribute groups" on page 1178

11. On the **JDBC Information** page, **Operating Systems** section, select the operating systems, and click **Next**. See "Specifying operating systems" on page 1189 for information about which operating systems to select.

    Note: Click **Insert Configuration Property** to select a property to insert. For more information, see ("Customizing agent configuration" on page 1330).

12. On the **Select key attributes** page, select key attributes or indicate that this data source produces only one data row. For more information, see "Selecting key attributes" on page 1151.

13. If you want to test a data source that you previously defined, in the Agent Editor window, select the **Data Sources** tab and select a JDBC data source. In the **JDBC Attribute Group Information** area, click **Test**. For more information about testing, see "Testing JDBC attribute groups" on page 1265.

14. If you want to view the configuration sections that were automatically generated, click the **Insert Configuration Property** tab of the Agent Editor.

    You can change the labels or default values for these properties to match the defaults that the user sees when they initially configure the agent.

15. Optional: Complete the **Attribute Information** page; for details, see "Fields and options for defining attributes" on page 1172. Do this step if you chose to manually create the JDBC data source without clicking Browse in step "4" on page 1259.

    The Agent Builder JDBC data source supports collecting data from most SQL types. The information in Table 271 on page 1261 describes the type of attribute that is created by the JDBC Browser when it detects a column of one of these types. These data types are the supported types for use with a monitoring agent.

Table 271. Supported SQL data types for use with a monitoring agent

SQL data type	IBM Tivoli Monitoring attribute that is created
BIGINT	This data type is a 64-bit gauge value in IBM Tivoli Monitoring. If you select IBM Tivoli Monitoring V6.2 compatibility, it is a 32-bit gauge.

Table 271. Supported SQL data types for use with a monitoring agent (continued)

SQL data type	IBM Tivoli Monitoring attribute that is created
DECIMALDOUBLEFLOATNUMERICREAL	These SQL Types are created as 64-bit gauge attributes in IBM Tivoli Monitoring. If the database metadata contains a scale value, that value is used; otherwise, the scale is set to 1. If you select IBM Tivoli Monitoring V6.2 compatibility, the attribute is a 32-bit gauge.
BITINTEGERSMALLINTTINYINT	The following SQL types are created as 32-bit gauge attributes in IBM Tivoli Monitoring.
BOOLEAN	This value is a 32-bit gauge in IBM Tivoli Monitoring with enumerations for TRUE and FALSE.
TIMESTAMP	Data in columns of this type are converted to a 16-byte IBM Tivoli Monitoring time stamp attribute.
TIMEDATECHARLONGVARCHARVARCHAR	These SQL types are all treated as string attributes by the browser. The column size is used as the attribute size up to 256, which is the default string attribute size for the JDBC browser.

Note: If you collect data from a data type that is not listed, a string attribute is used by default. The agent also tries to collect the data from the database as a string.

Modify the enumeration values that are set for Error, Missing data, and No value in the **Attribute Information** page, if required. Modify the values to avoid any overlap with legitimate values that might be returned from database table columns.

JDBC configuration

When you define a JDBC data source in your agent, some configuration properties are created for you.

If you define a JDBC data source in your agent, the agent must use Java to connect to the JDBC database server. Java configuration properties are added to the agent automatically. The following Java configuration properties are specific to the agent runtime configuration:

- *Java Home*: A fully qualified path that points to the Java installation directory
- *JVM Arguments*: Use this parameter to specify an optional list of arguments to the Java virtual machine.
- *Trace Level*: This parameter defines the amount of information to write to the Java trace log file. The default is to write only Error data to the log file.

Note: Agent Builder does not require the Java properties because it uses its own JVM and logging, which are configured through the `JLog` plug-in.

If you define a JDBC data source in your agent, the following required, common configuration fields are added to the agent automatically:

- *JDBC database type*: Type of database to which you are connecting, IBM DB2, Microsoft SQL Server, or Oracle Database Server.
- *JDBC user name*: User name that is used to authenticate with the database server.
- *JDBC password*: Password that is used to authenticate with the database server.
- *Base paths*: List of directories that are searched for JAR files that are named in the *Class Path* field, or directories that are named in the *JAR directories* field, that are not fully qualified. Directory names are separated by a semi-colon (`;`) on Windows, and by a semi-colon (`;`) or colon (`:`) on UNIX systems.

- *Class path*: Explicitly named JAR files to be searched by the agent. Any files that are not fully qualified are appended to each of the Base Paths until the JAR file is found.
- *JAR directories*: List of directories that are searched for JAR files. Directory names are separated by a semi-colon (`;`) on Windows, and by a semi-colon (`;`) or colon (`:`) on UNIX systems. The JAR files in these directories do not have to be explicitly identified; they are found because they are in one of these directories. Subdirectories of these directories are not searched. Any directories that are not fully qualified are appended to each of the Base Paths until the directory is found.

The runtime configuration also requires that you specify some additional details to connect to the database. You can choose how to specify the remaining configuration items, either as a JDBC URL or as basic configuration properties (the default):

- URL configuration option
  - JDBC connection URL: Vendor-specific connection URL that provides details on which host the database is located and the port number to which to connect. The URL format typically looks as follows:

    ```
    jdbc:identifier://server:port/database
    ```

    see the JDBC driver vendor documentation for the different URL formats.
- JDBC Basic Properties option (default)

  JDBC server name: Host name that the database server is running on.
  JDBC database name: Name of the database on the host where the connection is made.
  JDBC port number: Port number on which the database server is listening.

Note: With the JDBC data provider, you can monitor multiple database types in the same agent by using subnodes. To monitor in this way, you must carefully define the Subnode Configuration Overrides. If you monitor multiple database types, the following configuration settings are likely to be different:

- JDBC database type
- JDBC user name
- JDBC password

If you are using the basic configuration option, you must also define overrides for the following properties on the **Subnode Configuration Overrides** page:

- JDBC server name
- JDBC port number
- JDBC database name

To define the configuration overrides for your subnode, see "Using subnodes" on page 1314 for more details about accessing the **Subnode Configuration Overrides** page. When you configure the agent at run time, all of these properties must be configured for each new subnode instance that is created.

In addition to configuration overrides, your agent must also point to JDBC drivers for each database type that you plan to connect to from your subnodes. The *JAR directories* parameter is the most convenient way to point to your JDBC drivers. List the directories that contain the JDBC drivers by using a semicolon to separate each directory. For example, if you are connecting to DB2 and Oracle databases with the agent, you must specify a *JAR directories* value similar to this example: `C:\Program Files\IBM \SQLLIB\java;C:\oracle\jdbc`.

Stored procedures

Example SQL and DB2 stored procedures that you can use with the JDBC data provider.

The JDBC data provider can process the result sets returned by a stored procedure. String or integer input parameters can be passed to the stored procedure. The following syntax runs a stored procedure:

```
call[:index] procedureName [argument] ...
```

Where:

index
: An optional integer that specifies which result set is to be used by the data provider. This parameter is useful when the stored procedure returns multiple result sets and you want to collect only the values from one of the result sets. If an index is not specified, data from each result set is collected and returned.

procedureName
: The name of the stored procedure that is to be run by the JDBC data provider.

argument
: An input argument to the stored procedure. Multiple arguments must be separated by a space. If the argument contains a space character, enclose the entire argument in double quotation marks. If the argument can be parsed as an integer, it is passed to the stored procedure as an integer argument. Any argument that is enclosed in double quotation marks is passed as a string argument.

SQL Server Samples

`call sp_helpdb`

Runs the procedure `call sp_helpdb` which requires no arguments. Data from all returned result sets are included in the data that is returned by the data provider.

`call:2 sp_helpdb master`

Runs the procedure `sp_helpdb` with the master argument. This argument is a string input argument. Only data from the second result set that is returned by the stored procedure is included in the data that is returned by the data provider.

When the index is not specified, data from all returned results sets is collected. You must ensure that the data returned in these cases is compatible with the attributes you define. Agent Builder creates attributes from the first returned result set, and any further result sets are expected to be compatible with the first one.

DB2 stored procedure

Here is a sample DB2 function that is written in SQL. This function demonstrates how to return results that can be processed by the Agent Builder JDBC data provider:

```
-- Run this script as follows:
-- db2 -td# -vf db2sample.sql

-- Procedure to demonstrate how to return a query from
-- a DB2 stored procedure, which can then be used by
-- an Agent Builder JDBC provider. The stored procedure
-- returns the following columns:
-- Name               Description              Data Type
-- current_timestamp  The current system time  timestamp
-- lock_timeout       The lock timeout         numeric scale 0
-- user               The user for the session String 128 characters long
DROP procedure db2sample#

CREATE PROCEDURE db2sample()
  RESULT SETS 1
  LANGUAGE SQL
BEGIN ATOMIC

  -- Define the SQL for the query
  DECLARE c1 CURSOR WITH HOLD WITH RETURN FOR
  SELECT CURRENT TIMESTAMP as current_timestamp,
CURRENT LOCK TIMEOUT as lock_timeout, CURRENT USER as user
  FROM sysibm.sysdummy1;

  -- Issue the query and return the data
  OPEN c1;
END#
```

This function can be called from Agent Builder by using the same syntax that is defined for other stored procedures. In this case, you define `call db2sample` as your JDBC statement to run this stored procedure.

Oracle stored procedures

Oracle stored procedures do not return result sets. Instead, you must write a function that returns an Oracle reference cursor. Here is a sample Oracle function that is written in PL/SQL that demonstrates how to return results that can be processed by the Agent Builder JDBC data provider:

```
CREATE OR REPLACE FUNCTION ITMTEST
RETURN SYS_REFCURSOR
IS
        v_rc SYS_REFCURSOR;
BEGIN
  OPEN v_rc FOR SELECT * FROM ALL_CLUSTERS;
  RETURN v_rc;
END;
```

This function can be called from Agent Builder by using the same syntax that is defined for other stored procedures. In this case, you define `call ITMTEST` as your JDBC statement to run this stored procedure. Because the Oracle function must return a cursor reference, only one result set can be processed by Oracle functions. This means that the index option is not supported for Oracle because there is no way to return multiple result sets.

Testing JDBC attribute groups

You can test the JDBC attribute group that you created, within Agent Builder.

Procedure

1. You can start the Testing procedure in the following ways:

   - During agent creation click **Test** on the **JDBC Information** page.
   - After agent creation, select an attribute group on the Agent Editor **Data Source Definition** page and click **Test**. For more information about the Agent Editor, see "Using the Agent Editor to modify the agent" on page 1152.

   After you click **Test** in one of the previous two steps, the **Test JDBC Statement** window is displayed.
2. Optional: Before you start testing, you can set environment variables, configuration properties, and Java information.

   For more information, see "Attribute group testing" on page 1347. For more about JDBC configuration properties, see ("JDBC configuration" on page 1262).
3. Click **Start Agent**.

   A window indicates that the Agent is starting.
4. To simulate a request from Tivoli Enterprise Portal or SOAP for agent data, click **Collect Data**.

   The agent queries the database with the specified SQL query. The **Test JDBC Statement** window collects and shows any data in the agent's cache since it was last started.

   Note: The order of the returned data is significant; for example, the data value in the first returned column is always assigned to the first attribute. If you change the JDBC statement, you must add, remove, or reorder the attributes to match the columns returned by the statement.
5. Optional: Click **Check Results** if the returned data is not as you expected.

   The **Data Collection Status** window opens and shows you more information about the data. The data that is collected and displayed by the Data collection Status window is described in "Performance Object Status node" on page 1388
6. Stop the agent by clicking **Stop Agent**.
7. Click **OK** or **Cancel** to exit the **Test JDBC Statement** window. Clicking **OK** saves any changes that you made.

Related concepts

"Testing your agent in Agent Builder" on page 1346

After you use Agent Builder to create an agent, you can test the agent in Agent Builder.

Monitoring system availability by using Ping

You can define a data source to test a list of network devices by using the Internet Control Message Protocol (ICMP) echo ping. The host name or IP address of the devices you want to test are listed in one or more device list files. A separate Ping configuration file specifies the path to each device list file. Then, the name of the Ping configuration file is set in the agent runtime configuration. The results include the status of each network device.

Before you begin

Create device list files and a ping configuration file (see "Configuration files" on page 1266).

About this task

Part of network management involves the ability to determine whether systems respond to an Internet Control Message Protocol (ICMP) ping. Use this data source to monitor basic online or offline status for a set of servers or other critical devices in your environment. Monitoring with ping is simple and low-overhead. To monitor a list of devices, add the Ping data collector to your agent.

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Network management data** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **Ping**.
3. Click **Next**.
4. In the **Operating Systems** area in the **Ping Information** window, select the operating systems.
5. Optional: You can test this attribute group by clicking **Test**. For more information about testing, see "Testing Ping attribute groups" on page 1267
6. Optional: You can create a filter to limit the data that is returned by this attribute group by clicking **Advanced**. For more information about filtering data from an attribute group, see "Filtering attribute groups" on page 1178
7. Do one of the following steps:

   a) If you are using the **Agent** wizard, click **Next**.

   b) Click **Finish** to save the data source and open the Agent Editor.
8. For more information about adding attributes, see ("Creating attributes" on page 1169).

Results

For more information about the attribute group for Ping, see "Ping attribute group" on page 1417.

Configuration files

You provide the agent with the list of devices to ping by using configuration files.

The agent requires two types of configuration files.

Device list file

Includes a list of devices to ping. If you have many devices, you can divide them across multiple device list files. The agent starts a separate thread for each device list file and cycles through the files in parallel. It cycles through each file every 60 seconds or every 30 seconds plus the time it takes to ping the list, whichever is longer.

The syntax of the device list file is as follows:

```
LISTNAME=list_name
device_name or host_name
device_name or host_name
device_name or host_name device_name or host_name
```

Where *list_name* is a description for the devices in that file. If no list name is defined, the name of the device list file is used. The list name does not need to be the first entry in the file. However, if the file has multiple list name definitions, the last definition is used.

There is no limit to the number of devices you can include in a device list file. However, including too many entries defeats the purpose of having a targeted list of critical devices and increases the overall workload. It might be more difficult to retrieve the status of each device within the 60-second monitoring interval.

At the start of each cycle, the agent checks the last modification time of the device list file. If the last modification time of the file is more recent than the last time the agent read the file, the agent rereads the file without requiring a restart.

Ping configuration file

Specifies the location of each device list file. Use the fully qualified path or a path relative to the location of the ping configuration file. The ping configuration file is passed as a runtime configuration parameter to the agent.

Example

In the following example, devices are divided into two files.
The `/data/retailList.txt` file contains the following entries:

```
LISTNAME=Retail
frontend.mycompany.com
productdb.mycompany.com
```

The `/data/manufacturingList.txt` file contains the following entries:

```
LISTNAME=Manufacturing systems
manufloor.mycompany.com
stats.supplier.com
```

The ping file, `/data/pinglists.txt`, contains the following entries:

```
/data/retailList.txt
/data/manufacturingList.txt
```

Network Management configuration property

After a ping data source is added, the configuration is displayed on the **Runtime Configuration Information** page of the Agent Editor.

The **Network Management** configuration section of the **Runtime Configuration Information** page contains the following property:

Table 272. Network Management configuration properties

Name	Valid values	Required	Description
`Ping configuration file`	Path to a file	No. If this file is not provided, the KUMSLIST file is used from the agent bin directory.	The path to the file that contains a list of files, each containing a list of hosts to monitor by using ICMP pings.

Testing Ping attribute groups

You can test the Ping attribute group that you created within Agent Builder.

Procedure

1. You can start the Testing procedure in the following ways:
    - During agent creation click **Test** on the **Ping Information** page.

- After agent creation, select an attribute group on the Agent Editor **Data Source Definition** page and click **Test**. For more information about the Agent Editor, see "Using the Agent Editor to modify the agent" on page 1152.

After you click **Test** in one of the previous two steps, the **Test Settings** window opens.

2. Optional: Before you start testing, you can set environment variables and configuration properties. For more information, see "Attribute group testing" on page 1347.
3. Click **Browse** to select a Ping configuration file. For more about Ping configuration files, see "Configuration files" on page 1266
4. Click **Start Agent**. A window indicates that the Agent is starting.
5. To simulate a request from the monitoring environment for agent data, click **Collect Data**. The agent pings the devices that are specified in the device list file, which is referenced from the Ping configuration file.
6. The **Test Settings** window collects and shows any data in the agent's cache since it was last started.
7. Optional: Click **Check Results** if the returned data is not as you expected.

   The **Data Collection Status** window opens and shows you more information about the data. The data that is collected and shown by the Data collection Status window is described in "Performance Object Status node" on page 1388.
8. Stop the agent by clicking **Stop Agent**.
9. Click **OK** or **Cancel** to exit the **Test Settings** window. Clicking **OK** saves any changes that you made.

Related concepts

"Testing your agent in Agent Builder" on page 1346
After you use Agent Builder to create an agent, you can test the agent in Agent Builder.

Monitoring HTTP availability and response time

You can configure a data source to monitor the availability and response time of selected URLs. Use a configuration file to define a list of URLs. Set the name of the file in the agent runtime configuration. In IBM Tivoli Monitoring, you can also use Take Action commands to add and remove monitored URLs. The status for each URL is added as a line in the resulting data set.

About this task

For each URL you monitor, the results provide general information about the HTTP response to the HTTP request. The results include whether it can be retrieved, how long it takes to retrieve, and the size of the response. If the response content is HTML, information is also provided about the page objects within the URL.

You can monitor URLs that use the HTTP, HTTPS, FTP, and file protocols. You specify the URLs to monitor in the HTTP URLs file, or through Take Action options.

Important: At the time of release, Take Action commands are not available in an IBM Cloud Application Performance Management environment. They are available only in a Tivoli Monitoring environment.

This data source requires a Java runtime environment. The following versions of Java are supported:

- Oracle Corporation Java Version 5 or later
- IBM Corporation Java Version 5 or later

Use the following procedure to create an attribute group to monitor a list of URLs:

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Data from a server** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **HTTP**.
3. Click **Next**.

4. On the **HTTP Information** page , select one or more operating systems in the **Operating Systems** area.
5. Optional: Click **Test** to test this attribute group. For more information about testing, see "Testing HTTP attribute groups" on page 1275
6. Optional: Click **Advanced** to create a filter to limit the data that is returned by this attribute group. For more information about filtering data from an attribute group, see "Filtering attribute groups" on page 1178
7. Do one of the following steps:
   a) If you are using the **Agent** wizard, click **Next**.

   b) Click **Finish** to save the data source and open the Agent Editor.

Results
The HTTP data source creates two attribute groups: Managed URLs and URL Objects. You can add, modify, or delete attributes.

Related tasks

"Creating attributes" on page 1169
You can add new attributes to a data set.

Related reference

"HTTP attribute groups" on page 1420
The two HTTP attribute groups, `Managed URLs` and `URL Objects`, are used to receive information from URLs and the objects within theses URLs.

HTTP tables

Reference information about the default HTTP attribute groups.

The two attribute groups that are created by the HTTP data source are:

`Managed URLs`
The Managed URLs table provides availability and response time data about each URL being monitored.

`URL Objects`
The URL Objects table contains a separate URL entry for each embedded object. For example, the `.gif` and `.jpg` files that might be used in the website that is listed in the Managed URL report.

For information about the syntax that is used in the Managed URLs and URL Objects tables, see ("Specific fields for HTTP attributes" on page 1270).

When you want to monitor the response time and availability of specific objects within a website, review the contents of the URL Objects table. The URL Objects table monitors a specific list of objects that are detected in downloaded HTML files. The following table lists the HTML elements that are searched for objects to monitor and the attributes within these elements that reference the objects:

Table 273. HTML elements searched for objects to monitor

HTML element	Attribute containing object to be monitored
`img`	`src`
`script`	`src`
`embed`	`src`
`object`	`codebase` or `data`
`body`	`background`
`input`	`src`

In the following example HTML extract, the object that is monitored is the image that is referenced by the `src` attribute of the `img`element.

```
<img src="/v8/images/id-w3-sitemark-simple.gif" alt="" width="54" height="33" />
```

A full URL to the image is calculated based on the URL to the source document.

Note: If you do not want to monitor objects that are found in a web page, in the URL Monitoring configuration section, set the **Page object collection** property to **No**.

Specific fields for HTTP attributes

In the **Attribute information** page , there are two fields for HTTP attributes that define how data is collected from the URL. The **Attribute Type** field can be any value from a list that controls the information about the URL that is returned. Some attribute types require a value in the **Type Value** field.

The following table describes all of the attribute types for the Managed URLs attribute group, and the type value when one is required:

Table 274. HTTP Attribute Information - Managed URLs

Attribute type	Description	Type value	Data type that is returned	Differences with FTP and file protocols
`XPath Query`	Runs an XPath query on the content that is returned from a URL connection. The query must be written to return data useful for an attribute, not a list of nodes.	The XPath query to run against the content that is obtained from a URL connection.	The data that is returned can be a string, a numeric, or a timestamp value. If the data is in the XML DateTime format, you can specify timestamp as the attribute type. The agent converts the value to a Candle Timestamp.	None
`Response Time`	The amount of time in milliseconds that it took to download the content from the requested URL.	None	Integer (number of milliseconds)	None
`Response Message`	The HTTP response message that is returned by the server.	None	String	The response message applies only if the URL uses the HTTP or HTTPS protocols.
`Response Code`	The HTTP response code that is returned by the server.	None	Integer	The response code applies only if the URL uses the HTTP or HTTPS protocols. It is always 0 for file or FTP URLs.

Table 274. HTTP Attribute Information - Managed URLs (continued)				
Attribute type	**Description**	**Type value**	**Data type that is returned**	**Differences with FTP and file protocols**
`Response Length`	The size of the content in bytes that is downloaded from the requested URL	None	Integer (size in bytes)	None
`Response Header`	The response header can be used to retrieve a value from one of the URL response header fields. The argument specifies which field is requested.	The response header to collect.	String	Generally FTP and file protocols do not have any headers that can be collected.
`Request URL`	The connection is made to this URL. All of the response keywords provide information about the connection to this URL. The XPath Query can be used to obtain information that is obtained from the content that is returned by accessing this URL.	None	String	None
`Page Objects`	The number of objects that are discovered on the monitored HTML page that are monitored by the URL Objects attribute group.	None	Integer	None
`Total Object Size`	The total size of the objects that is monitored in the URL Objects attribute group for this web page.	None	Integer (in bytes)	None
`Alias`	The user specified alias for this URL.	None	String	None
`User`	The user specified data for this URL.	None	String	None

The following table describes the attribute types for the URL Objects attribute group:

Table 275. HTTP Attribute Information - URL Objects

Attribute type	Description	Type value	Data type that is returned	Differences with FTP and file protocols
URL	The URL that is monitored in the Managed URLs table.	None	String	None
Object Name	The URL for the object that is monitored within the **HTML** page.	None	String	None
Object Size	The size in bytes of the content that is downloaded from the Object Name URL.	None	Numeric	None
Object Response Time	The time in milliseconds it took to download the page object.	None	Numeric	None

Monitoring a URL

You can start monitoring any URL by including it in the URLs file or by using the HTTP URL Add Take Action option.

URLs file

The URLs file specified in configuration can be in any directory. If this file does not exist or is empty, then you can start URL monitoring by using Take Actions. For more information, see "Take Action option" on page 1273. If you already have a Tivoli Universal Agent that uses the Tivoli Universal Agent HTTP Data Provider, you can reuse the KUMPURLS file. When you are configuring the agent, point to your KUMPURLS file.

The following table provides examples of how URLs are entered in the URLs file, depending on the method by which they were added.

Table 276. URLs file entries

URLs	Added by

`www.bbc.co.uk`
`http://weather.com`
`www.ibm.com`
	Manually adding entries to the file. If no protocol is specified, as in the `www.ibm.com` example, http is assumed.

`ftp://userid:password@ftpserver/index.html`	Manually added by using File Transfer Protocol (FTP)
`http://www.ibm.com USER=ibm ALIAS=ibm`	Using the HTTP URL Add Take Action

`file:/tmp/samples.html USER=samples \`
`ALIAS=samples`
	Using a HTTP URL Add Take Action that uses FTP

Table 276. URLs file entries (continued)

URLs	Added by

`http://google.com INTERVAL=60 CACHE=50 \`
`USER=google ALIAS=search`
	Example from the Tivoli Universal Agent KUMPURLS file

When you directly edit the URLs file, your changes are implemented when the agent does its next data collection.

Take Action option

You can also specify URLs to monitor through a Take Action option that is called `HTTP URL Add`.

Restriction: This option is not available in the current release of IBM Cloud Application Performance Management, because you can not start Take Action commands manually.

When this option is selected, a window is displayed where you can specify the following parameters:

URL
: A required parameter that represents the URL itself. You can type this parameter with or without the `http://`or `https://`prefix.

`Alias`
: An optional parameter that you can specify to associate a more meaningful name to a URL. No spaces are allowed in this parameter. If this parameter is not completed, the Alias Name defaults to blank.

`User_Data`
: An optional parameter that you can specify to enter data about the URL. If this parameter is not completed, the `User_Data`defaults to `INITCNFG`.

After you complete the information and close the window, assign the `HTTP URL Add` action to the destination managed system that is associated with the agent. Monitoring begins immediately for the new URL. The URL is also added to the URLs file so that it continues to be monitored across agent restarts.

A corresponding Take Action option is named `HTTP URL Remove`. Use the `HTTP URL Remove` action to immediately stop monitoring for a particular URL. The removed URL is also deleted from the URLs file. The **HTTP URL Remove** window requests only the URL and `User_Data` values. The URL and `User_Data` values must match the values that are seen in the Tivoli Enterprise Portal or the Remove action fails. For example, if you omitted the `http://` from the URL field of the Add action, you must include it in the URL field of the Remove action. If you did not specify `User_Data`, you must specify `INITCNFG` as seen in the Tivoli Enterprise Portal.

If a URL is added manually to the URLs file, you can delete it with the Take Action. If you delete with the Take Action, you must specify the values as seen in the Tivoli Enterprise Portal. For example, if you added `www.ibm.com` to your URLs file, the Tivoli Enterprise Portal displays `http://www.ibm.com` as the URL and `INITCNFG` as the `User_Data`. To remove the URL with the Take Action, you must use the values that are seen in the Tivoli Enterprise Portal.

After you complete the information and close the window, assign the `HTTP URL Remove` action to the destination managed system that is associated with the agent.

Monitor `https://` URLs

The HTTP data source can monitor only secure `https://` URLs that do not require scripted access or interactive prompting.

If the `https://` URL can be retrieved with a standard HTTP Get call, then it can be monitored.

Proxy server

If the system where the agent is running requires a proxy to access the SOAP data provider, you must specify proxy server configuration properties.

For more information, see "Proxy Server configuration" on page 1274.

HTTP configuration

Reference information about HTTP configuration.

After an HTTP data source is added, the configuration is displayed on the **Runtime Configuration** page of the Agent Editor. Configuration sections are added for URL Monitoring, for Proxy Server authentication, and for Java.

URL Monitoring configuration

The URL Monitoring configuration section contains the following properties:

Table 277. URL Monitoring configuration properties

Name	Valid values	Required	Description
HTTP URLs file	Path to a file	Yes	The path to the file that contains a list of URLs.
Page Object Collection	Yes, No The default value is Yes.	No	Whether to download objects that are found in a web page and collect data from them.

Proxy Server configuration

The Proxy Server configuration section contains the following properties:

Table 278. Proxy Server configuration properties

Name	Valid values	Required	Description
Proxy Hostname	String	No	The proxy host name to be used for HTTP connections.
Proxy User Name	String	No	The user name for the proxy server.
Proxy Port	Positive integer The default value is 80.	No	The HTTP port number of the proxy server.
Proxy Password	Password	No	The password for the proxy server.

Note: If the **Proxy Hostname** property is blank, no proxy is used.

Java configuration

If you define an HTTP data source in your agent, the agent must use Java to connect to the HTTP server. Java configuration properties are added to the agent automatically. The following Java configuration properties are specific to the agent runtime configuration. The Agent Builder does not require the Java properties because it uses its own JVM and logging, which are configured through the JLog plug-in):

Table 279. Java configuration properties

Name	Valid values	Required	Description
Java Home	Fully qualified path to a directory	No	A fully qualified path that points to the Java installation directory.

Table 279. Java configuration properties (continued)

Name	Valid values	Required	Description
Trace Level	Choice (The default value is Error)	Yes	Use this property to specify the trace level that is used by the Java providers.
JVM Arguments	String	No	Use this property to specify an optional list of arguments to the Java virtual machine.

Testing HTTP attribute groups

You can test the HTTP attribute group that you created, within Agent Builder.

Procedure

1. Start the Testing procedure in the following ways:
   - During agent creation click **Test** on the **HTTP Information** page.
   - After agent creation, select an attribute group on the Agent Editor **Data Source Definition** page and click **Test**. For more information about the Agent Editor, see "Using the Agent Editor to modify the agent" on page 1152

   After you click **Test** in one of the previous two steps, the **HTTP Test** window is displayed.
2. Click **Browse** to select the HTTP URLs file. For more information about URLs files, see "URLs file" on page 1272.
3. Optional: Set environment variables, configuration properties, and Java information before you start testing.

   For more information, see "Attribute group testing" on page 1347. For more information about HTTP configuration, see "HTTP configuration" on page 1274.
4. Click **Start Agent**.

   A window indicates that the Agent is starting.
5. To simulate a request from Tivoli Enterprise Portal or SOAP for agent data, click **Collect Data**.

   The agent monitors the URLs defined in the HTTP URLs file. The **HTTP Test** window displays any data that is returned.
6. Optional: Click **Check Results** if the returned data is not as you expected.

   The **Data Collection Status** window opens and shows you more information about the data. The data that is collected and displayed by the Data collection Status window is described in "Performance Object Status node" on page 1388
7. Stop the agent by clicking **Stop Agent**.
8. Click **OK** or **Cancel** to exit the **HTTP Test** window. Clicking **OK** saves any changes that you made.

Related concepts

"Testing your agent in Agent Builder" on page 1346

After you use Agent Builder to create an agent, you can test the agent in Agent Builder.

Monitoring data from a SOAP or other HTTP data source

You can define a data source to receive data from an HTTP server (for example, using the SOAP protocol). The data source sends an HTTP request to an URL and parses the response (in XML, HTML, or JSON formats) into the attributes of the resulting data set. You can select the data that is retrieved from the request.

About this task

By using the SOAP data source, you can specify an HTTP URL and send a GET, POST, or PUT request. For POST or PUT requests, you can specify the associated POST data. An XML, HTML, or JSON response is retrieved and parsed, and the data is exposed to the monitoring environment in attributes. You can define the attributes as all of the values within a particular element. Or you can define custom XPath values to specify how to populate individual attributes. You can also combine the two mechanisms.

Use the following procedure to collect and parse XML, HTML, or JSON responses from a URL:

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Data from a server** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **SOAP**.
3. Click **Next**.
4. On the **SOAP Information** page, enter a URL.

   The default value is:

   ```
   http://${KQZ_HTTP_SERVER_NAME}:${KQZ_HTTP_PORT_NUMBER}
   ```

   Note: You can use a configuration variable or multiple configuration variables that resolve to a URL. Click **Insert Configuration Property** to select a property to insert. For more information, see “Customizing agent configuration” on page 1330.
5. Select a request type. The default request type is Get. For Post and Put requests, enter the data to be processed.

   Note: For Post and Put requests, the **Insert Configuration Property** is enabled. Click **Insert Configuration Property** to include a configuration variable in the data to be processed. For more information, see (“Customizing agent configuration” on page 1330).
6. Click **Browse**

   Note: If after you enter a URL and select a request type, you do not want to use the SOAP browser to build the definition, enter a **Row Selection XPath**. You enter the **Row Selection XPath** in the **SOAP Information** window. Next, define all of the attributes for the attribute group.
7. In the **SOAP Browser** window, do the following steps:

   a) Enter a URL and select a request type if you did not already do so.

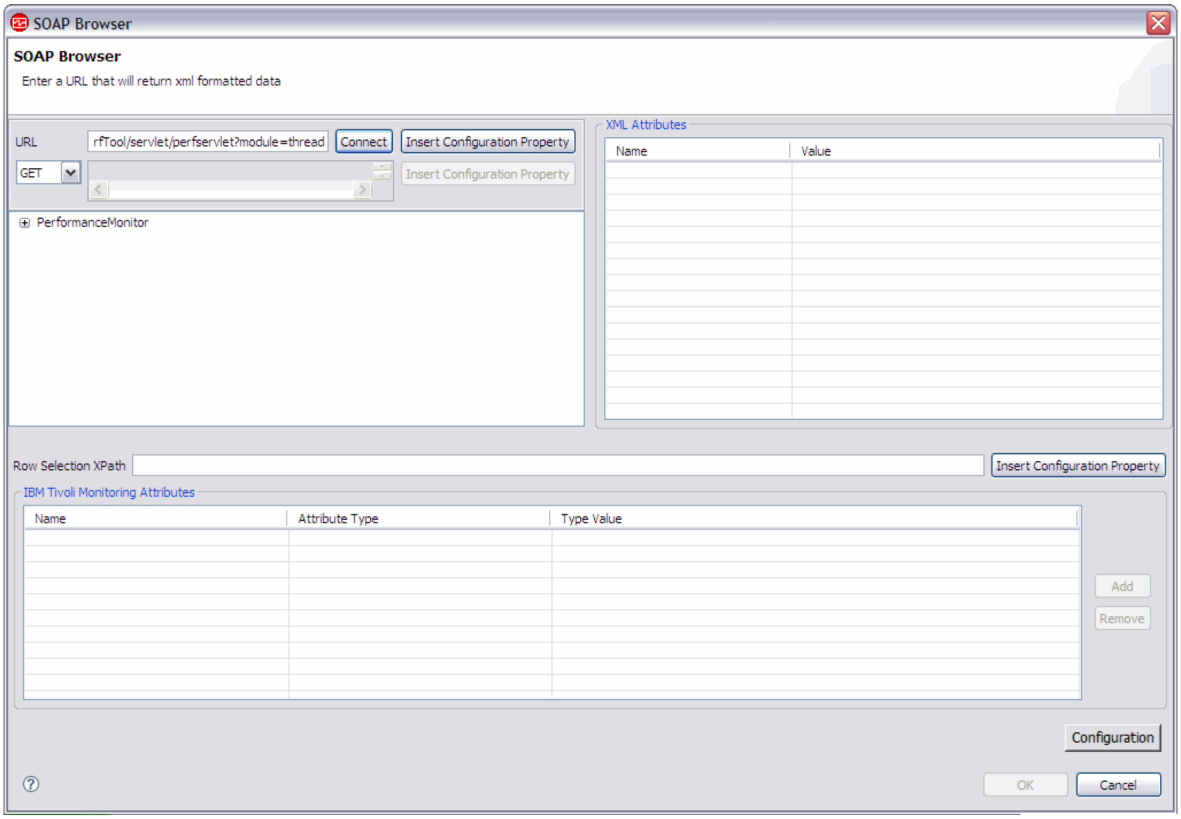
   b) Click **Configuration** to set any configuration properties that are referenced in the URL or other fields.

   c) Click **Connect** to obtain data from the SOAP provider.

   When you connect to the URL, a list of XML elements for this URL is shown in a Document Object Model (DOM) tree. An HTML or JSON response is converted to XML and displayed as a DOM tree. For details about conversion of a JSON response to XML, see “XML representation of JSON data” on page 1279. In the WebSphere Application Server example in (Figure 44 on page 1277), the following URL was entered:

   ```
   http://nc053011.tivlab.raleigh.ibm.com:9080/wasPerfTool/servlet
   /perfservlet?module= \threadPoolModule
   ```

   The PerformanceMonitor XML element is shown. This element is the top-level XML element in the XML document that is returned by the SOAP provider.



SOAP Browser

Enter a URL that will return xml formatted data

URL: rfTool/servlet/perfservlet?module=thread

GET

Row Selection XPath:

SOAP Browser
Connect
Insert Configuration Property
Insert Configuration Property
PerformanceMonitor
XML Attributes
Name
Value
Insert Configuration Property
IBM Tivoli Monitoring Attributes
Name
Attribute Type
Type Value
Add
Remove
Configuration
OK
Cancel

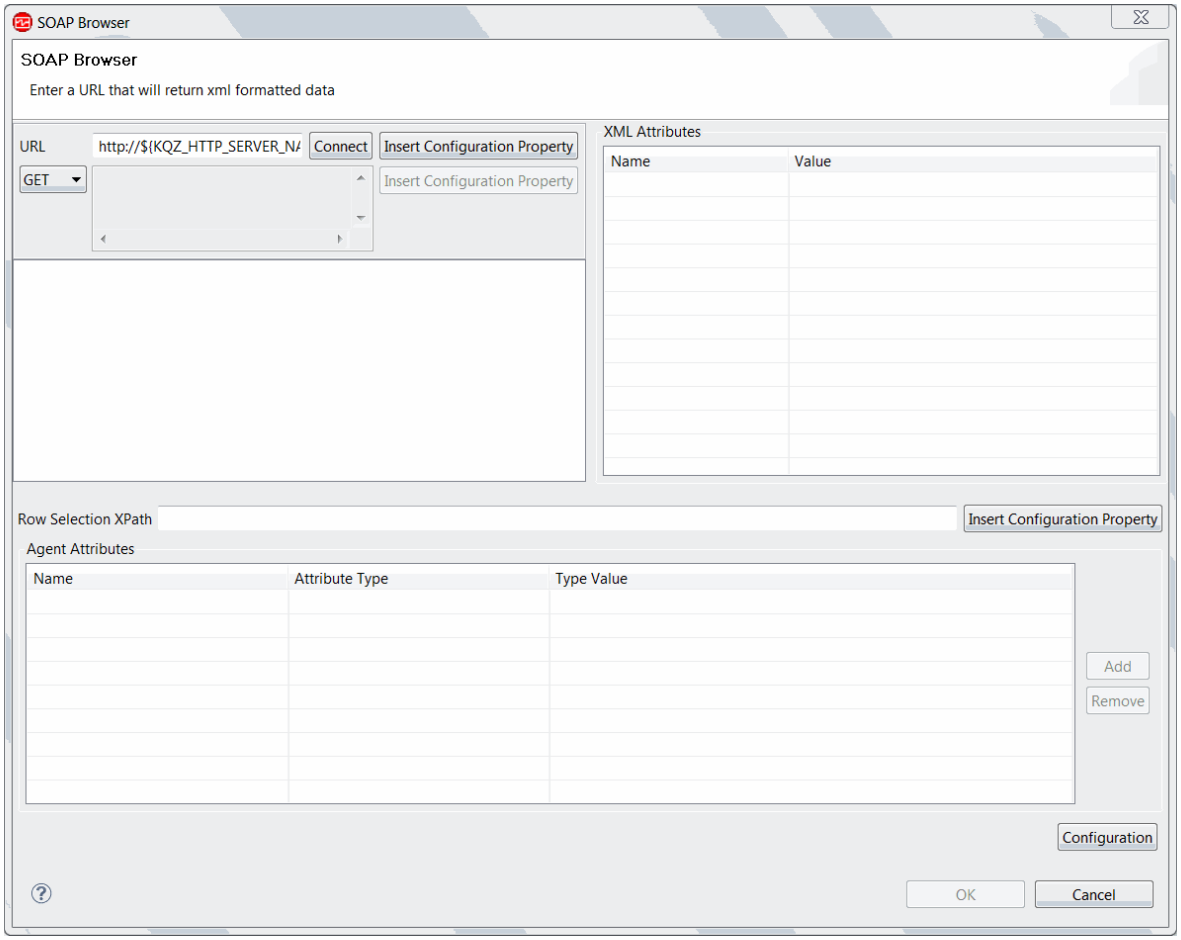
*Figure 44. **SOAP Browser** window*

d) In the DOM tree, find and select the XML node that you want to set as the **Row Selection XPath**.

In the WebSphere Application Server example in (Figure 45 on page 1278), the `PerformanceMonitor/Node/Server/Stat/Stat/Stat` node is selected. This node represents a row of data in the attribute group. When you select a node in the DOM tree and click **Add**, you get all of the attributes and elements defined on that node of the tree. (You click **Add** in the **Agent Attributes** area).

When a node is selected, the **XML Attributes** area shows any XML attributes defined for the selected node. Select an XML attribute and click **Add** to include this attribute in the list of Agent Attributes.

Note: If more than one row of data is expected, the XPath must map to a node set. Where the Row Selection XPath returns a node that is set with only one item, the attribute group contains only one row.


SOAP Browser

Enter a URL that will return xml formatted data

URL: http://${KQZ_HTTP_SERVER_N/

GET

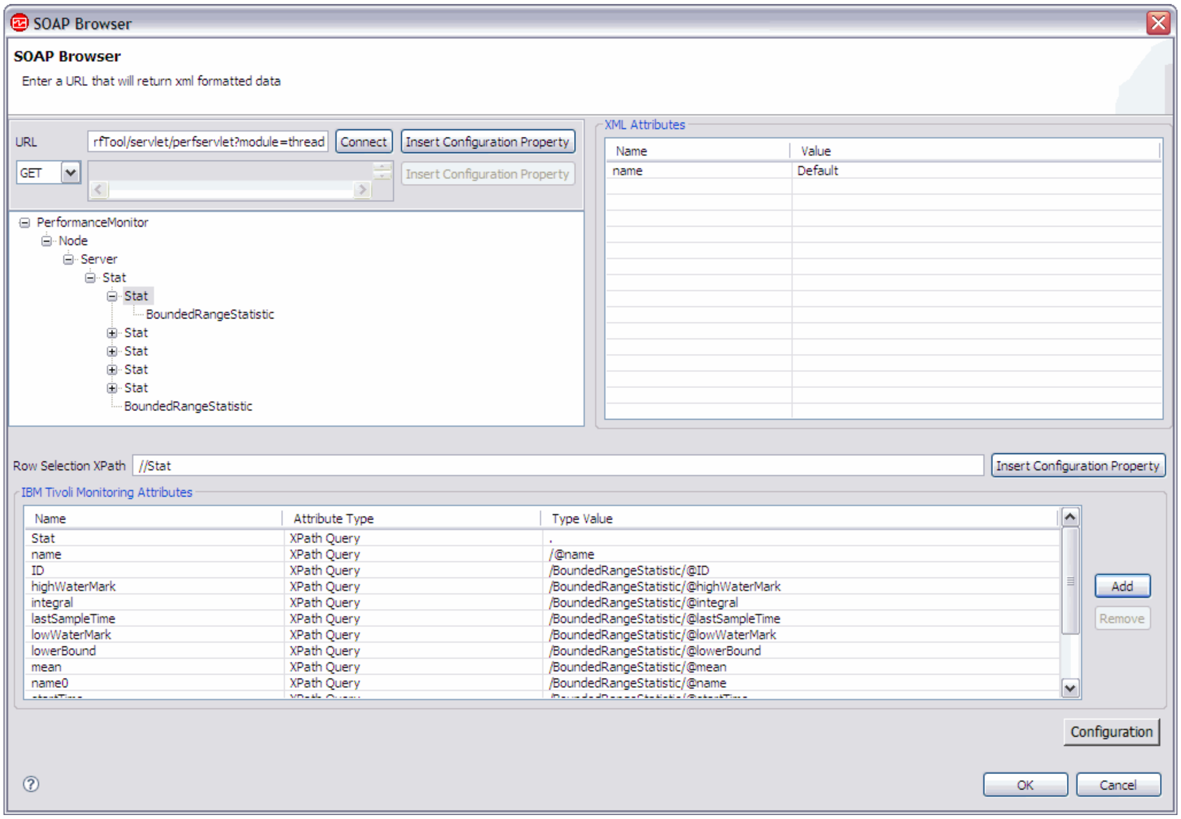
Figure 45. ***SOAP Browser*** *window*

e) Click **Add** in the Agent Attributes area.

The list of agent attributes is shown and the **Row Selection XPath** field is filled.

The XPath for each agent attribute is used to map XML nodes or elements to agent attributes. In the WebSphere Application Server example in theFigure 46 on page 1279, the first attribute in the list of agent attributes, `Stat`, is not of use and would be removed.

You can edit the name and XPath for an agent attribute in the **Type Value** field. For more information about using XPaths, see "XPath options" on page 1281



URL: rfTool/servlet/perfservlet?module=thread

GET

Name	Value
name	Default

Row Selection XPath: //Stat

Name	Attribute Type	Type Value
Stat	XPath Query	.
name	XPath Query	/@name
ID	XPath Query	/BoundedRangeStatistic/@ID
highWaterMark	XPath Query	/BoundedRangeStatistic/@highWaterMark
integral	XPath Query	/BoundedRangeStatistic/@integral
lastSampleTime	XPath Query	/BoundedRangeStatistic/@lastSampleTime
lowWaterMark	XPath Query	/BoundedRangeStatistic/@lowWaterMark
lowerBound	XPath Query	/BoundedRangeStatistic/@lowerBound
mean	XPath Query	/BoundedRangeStatistic/@mean
name0	XPath Query	/BoundedRangeStatistic/@name

*Figure 46. **SOAP Browser** window*

f) In the **SOAP Browser** window, click **OK** to save your changes and return to the **SOAP Information** window.

8. In the **SOAP Information** window, click **Next**.
9. If you did not use **Browse** earlier and you entered the **URL** and **Row Selection XPath** in the **SOAP Information** window, the **Attribute Information** page is shown. Specify the information for the first attribute on the **Attribute Information** page, and click **Finish**. You can then specify more attributes by using the Agent Editor. For more information about creating attributes, see ("Creating attributes" on page 1169).
10. If you used the **Browse** function in step "6" on page 1276, the **Select key attributes** page is shown. On the **Select key attributes** page, select key attributes or indicate that this data source produces only one data row. For more information, see "Selecting key attributes" on page 1151.
11. Optional: You can test this attribute group by clicking **Test**. For more information about testing, see "Testing SOAP attribute groups" on page 1283
12. Optional: You can create a filter to limit the data that is returned by this attribute group by clicking **Advanced**. For more information about filtering data from an attribute group, see "Filtering attribute groups" on page 1178
13. Do one of the following steps:

    a) If you are using the **Agent** wizard, click **Next**.

    b) Click **Finish** to save the data source and open the Agent Editor.

XML representation of JSON data

If the HTTP request returns JSON data, the data provider converts the data to XML.

The data provider converts the name of a JSON attribute to the element name. For a JSON attribute of a simple type, it converts the value to text data within the element. Embedded JSON objects are converted to embedded XML elements. Any subordinate attributes are converted to subordinate elements.

The root XML element is `JSON_document`.

If a JSON attribute name contains characters that are invalid in an element name, the data provider modifies it to produce a valid element name. The data provider also adds a `JSON_name` attribute to the element. The value of the attribute is the original JSON attribute name.

For every element of a JSON array, the data provider creates a `JSON_`*`xxx`*`_array_element` XML element, where *xxx* is the name of the array. The value of the array element is converted into text within the XML element. A `JSON_index` attribute is added to each XML element; the value of the attribute is the index of the array element within the array.

The data provider adds the following attributes to every element:

- `JSON_level`: the level of the node within the JSON file. The root of the tree, represented by the `JSON_document` tag, is level 1.
- `JSON_type`: the type of the JSON node (object, array, string, or number).

Specific fields for SOAP attributes
In the **Attribute Information** window, there are two fields for SOAP attributes that define how data is collected from the SOAP response.

The **Attribute Type** field can be any value from a list that controls the information about the response that is returned. Some attribute types require a value in the **Type Value** field. The default attribute type is XPath Query, which runs an XPath query against the SOAP server response content. The type value is the XPath query that is run. The following table describes all of the attribute types and the type value when one is required:

Table 280. SOAP Attribute Information

Attribute type	Description	Type value	Returned data type	Differences with FTP and file protocols
`XPath Query`	Runs an XPath query on the content that is returned from a URL connection. The query must be written to return data useful for an attribute, not a list of nodes.	The XPath query to run against the content that is obtained from a URL connection. If a row selection query was defined, this XPath query must be relative to the row selection query.	The data that is returned can be a string, a numeric, or a timestamp value. The Agent Builder browser for SOAP generally detects the correct data type for the attribute from the data that is being browsed. If the data is in XML DateTime format, you can specify timestamp as the attribute type and the agent converts the value to a Candle Timestamp.	None
`Response Time`	The amount of time in milliseconds that it took to download the content from the requested URL.	None	Integer (number of milliseconds)	None

Table 280. SOAP Attribute Information (continued)				
Attribute type	**Description**	**Type value**	**Returned data type**	**Differences with FTP and file protocols**
`Response Message`	The HTTP response message that is returned by the server.	None	String	The response message applies only if the URL uses the HTTP or HTTPS protocols.
`Response Code`	The HTTP response code that is returned by the server.	None	Integer	The response code applies only if the URL uses the HTTP or HTTPS protocols. It is always 0 for file or FTP URLs.
`Response Length`	The size of the content in bytes that was downloaded from the requested URL	None	Integer (size in bytes)	None
`Response Header`	The response header can be used to retrieve a value from one of the URL response header fields. The argument specifies which field is requested.	The response header field to collect.	String	Generally FTP and file protocols do not have any headers that can be collected.
`Request URL`	The connection was made to this URL. All of the response keywords provide information about the connection to this URL. The XPath Query can be used to obtain information that is obtained from the content that is returned by accessing this URL.	None	String	None

XPath options

Using XML Path Language, you can select nodes from an XML document. A few of the possible uses of XPaths for the SOAP data sources include:

- Using predicates in the XPath to identify the XML elements that correspond to rows of data in the IBM Tivoli Monitoring attribute group. You can use predicates in the XPath that maps XML elements or attributes to Tivoli Monitoring attributes, as in the following example:

  ```
  Stat[@name="URLs"]/CountStatistic[@name="URIRequestCount"]/@count
  ```

  Where there are multiple location steps in the XPath, each location step can contain one or more predicates. The predicates can be complex and contain boolean values or formula operators. For example:

  ```
  //PerformanceMonitor/Node/Server[@name="server1"]/Stat/Stat/Stat[@name=
  "Servlets"]/Stat
  ```

- Including node set functions in the XPath, if a row contains multiple XML elements of the same type. And if the position of an XML element in the node list determines the Tivoli Monitoring attribute the element maps to. Examples of node set functions are, `position()`, `first()`, `last()`, and `count()`.
- Doing simple data transformation, such as substring. If you specify the following substring:

  ```
  substring(myXMLElement,1,3)
  ```

  the XPath returns the first three characters of the XML element, myXMLElement.

You can specify elements outside the context of the Row Selection XPath by using two periods, `(..`, as in the following example:

```
/../OrganizationDescription/OrganizationIdentifier
```

SOAP configuration
After a SOAP data source is added, the configuration is displayed on the **Runtime Configuration** page of the Agent Editor.

Configuration sections are added for HTTP Server, for Proxy Server, and for Java. For information about Proxy server configuration, see ("Proxy Server configuration" on page 1274). For information about Java configuration, see "Java configuration" on page 1274.

HTTP Server

The HTTP Server configuration section contains the following properties:

Table 281. HTTP Server configuration properties

Name	Valid values	Required	Description
HTTP user name	String	No	The HTTP user
HTTP password	Password	No	The HTTP server password
HTTP server name	String (The default value is localhost)	No	The host or IP address of the HTTP server
HTTP port number	Numeric (The default value is 80)	No	The host or IP address of the HTTP server
Certificate validation enabled	True, False (The default value is True)	Yes	Disabling certificate validation is potentially insecure
HTTP trust store file	Path to a file	No	The HTTP trust store file

Table 281. HTTP Server configuration properties (continued)

Name	Valid values	Required	Description
HTTP trust store password	The HTTP trust store password	No	The HTTP trust store password

Proxy server

If the system where the agent is running requires a proxy to access the SOAP data provider, you must specify proxy server configuration properties. For more information, see "Proxy Server configuration" on page 1274.

Testing SOAP attribute groups

You can test the SOAP attribute group that you created, within Agent Builder

Procedure

1. You can start the Testing procedure in the following ways:
   - During agent creation click **Test** on the **SOAP Information** page.
   - After agent creation, select an attribute group on the Agent Editor **Data Source Definition** page and click **Test**. For more information about the Agent Editor, see "Using the Agent Editor to modify the agent" on page 1152

   After you click **Test** in one of the previous two steps, the **Test SOAP Collection** window is displayed.
2. Optional: Before you start testing, you can set environment variables, configuration properties, and Java information.

   For more information, see "Attribute group testing" on page 1347. For more information about SOAP configuration, see "SOAP configuration" on page 1282.
3. Change the URL, Row Selection XPath, and request type.
4. Click **Start Agent**.

   A window indicates that the Agent is starting.
5. To simulate a request from Tivoli Enterprise Portal or SOAP for agent data, click **Collect Data**. This action populates the Results table and you can preview how the data is parsed and shown in columns in the Tivoli Enterprise Portal.

   In the Results area, you can change the attribute definitions and reload the data to see how your changes affect the attribute group. You can right-click in a column results area to display options to edit the attribute. The attribute edit options are:
   - **Edit Attribute**
   - **Hide Attribute**
   - **Insert Attribute Before**
   - **Insert Attribute After**
   - **Remove**
   - **Remove Subsequent Attributes**
   - **Remove All**
6. Optional: Click **Check Results** if the returned data is not as you expected.

   The **Data Collection Status** window opens and shows you more information about the data. The data that is collected and shown by the **Data Collection Status** window is described in "Performance Object Status node" on page 1388.
7. Stop the agent by clicking **Stop Agent**.
8. Click **OK** or **Cancel** to exit the **Test SOAP Collection** window. Clicking **OK** saves any changes that you made.

Related concepts

“Testing your agent in Agent Builder” on page 1346
After you use Agent Builder to create an agent, you can test the agent in Agent Builder.

Monitoring data by using a socket

You can define a data source to collect data from an external application by using a TCP socket. The application must initiate the TCP connection to the agent and send data in a structured XML format. Depending on the application, the data source can produce a data set with a single row, multiple rows, or event data.

About this task

Use the socket data source to provide data to the agent from an external application, running on the same system as the agent. The external application can send data to the agent anytime it wants to. For example, you can develop a command-line interface that allows a user to post data to an attribute group when it is run. Another option is to modify a monitored application to send updates to the agent. The agent does not start or stop the application that is sending data to the socket; this action is controlled by the user.

There are some limitations with the socket data source:

- By default only connections to the local host (127.0.0.1) are possible. For more information about configuring your agent to accept connections from a remote host, see “Remote socket port connection” on page 1291.
- There is no mechanism in the socket API for the client to determine what subnodes are available. The client can send data for a specific subnode, but it must already know the subnode name.

Use the following procedure to create an attribute group to collect data by using a Transmission Control Protocol socket (TCP) socket.

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Custom programs** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **Socket**.
3. Click **Next**.
4. On the **Socket Information** page, enter an Attribute group name.
5. Enter a help text for the attribute group.
6. Select whether the attribute group **Produces a single data row**, **Can produce more than one data row**, or **Produces events**. For more information, see “Sending data” on page 1286.
7. In the Socket Information section, select a **Code page**. For more information, see “Character sets” on page 1289.
8. Optional: Click **Advanced** to modify the advanced properties for the attribute group. The **Advanced** option is active when you select that the attribute group **Can produce more than one data row**, or **Produces events**.
9. Click **Next**.
10. On the **Attribute Information** page, specify the first attribute for the attribute group. For more information about creating attributes, see “Creating attributes” on page 1169.
11. Click **Next**.
12. Optional: On the **Global Socket Data Source Information** page, **Error Codes** section, you can define the error codes that the socket client can send when it cannot collect data. For more information, see (“Sending errors instead of data” on page 1287). To define an error code, use the following steps:

    a) In the **Error Codes** section, click **Add**. An error code has a limit of 256 characters. Only ASCII letters, digits, and underscores are allowed. No spaces are allowed.

b) In the **Socket Error Code Definition** window, enter a display value that is shown in the **Performance Object Status** attribute group.

c) Enter an internal value. The internal value must be an integer from 1,000 to 2,147,483,647.

d) You must define a message text for each error. You can use message text that was entered previously by selecting it from the list. Click **OK** to return to the **Global Socket Data Source Information** page. The message text is used in the agent log file.

If no suitable message text is available, click **Browse** to set up the message text. The Messages (list) window opens. The message window lists messages that are defined in the agent. Until you define messages, the list remains blank. You can use **Edit** to alter a defined message and **Remove** to delete one or more messages that you defined.

e) In the Messages (list) window, click **Add** to see a **Message Definition** window. In the **Message Definition** window type, the text that describes the meaning of the new message and select the message type.

Note: The message identifier is automatically generated for you.

f) Click **OK**.

g) The Messages (list) window opens, with the new message. To verify the message and return to the **Global Socket Data Source Information** page, click **OK**.

13. Optional: In the **Supplemental Files** section of the **Global Socket Data Source Information** page, you can add files that are packaged with the agent. These files are copied to the agent system when the agent is installed.

The **File Type** column describes how each file is expected to be used. Three possible uses are described in the following table:

Table 282. File types for supplemental files

File Type	Description
Executable	Select this option if you want to include an executable file with the agent. The agent does not use these files.
Library	Select this option if you to include a library with the agent. The agent does not use these files.
Java resource	Select this option to include Java resources with the agent. The agent does not use these files.

For information about where the Supplemental Files are installed with your agent, see ("New files on your system" on page 1362).

Click **Edit** to edit the imported file. For more information, see ("Editing a command file definition" on page 1253).

14. Optional: You can test this attribute group by clicking **Test**. For more information about testing, see "Testing socket attribute groups" on page 1292
15. Optional: If the data source is sampled, you can create a filter to limit the data that is returned by this attribute group by clicking **Advanced**. The data source is sampled when you did not select "Produces events" on the **Socket Information** page. For more information about filtering data from an attribute group, see "Filtering attribute groups" on page 1178
16. Do one of the following steps:

a) If you are using the **Agent** wizard, click **Next**.

b) Click **Finish** to save the data source and open the Agent Editor.

Select the operating systems on which the agent listens to data from socket clients in the **Operating Systems** section of the **Socket Provider Settings** page. To open the page, click **Socket Provider Settings** in the outline view or click **Global Settings** in the Agent Editor on any socket attribute group page.

Note: Error codes and supplemental files can be updated in the **Error Codes** and **Supplemental Files** sections of the **Socket Provider Settings** page.

Sending socket information to the agent
When your agent contains one or more socket attribute groups, the agent opens a socket and listens for data from clients.

The application that sends socket data to the agent connects to a port that is defined in the agent. The port is either the value that is set by an agent configuration property or an ephemeral port that is allocated automatically by TCP/IP. For more information about socket ports and configuration, see "Socket configuration" on page 1290.

The data that is received must follow a structured XML format. The following XML information flows are possible by using the socket data source:

- Send one or more rows of data to the agent for a sampled attribute group
- Send a row of data to the agent for an attribute group that Produces events
- Send an error code to the agent instead of data.
- Send a task prefix registration to the agent
- Receive a task request from the agent
- Send a task response to the agent

Sending data

An attribute group is defined to receive sampled data or event data. When you create the attribute group, you specify an option that indicates whether the data to be received:

- Produces a single data row
- Produce more than one data row
- Produces events

If you select **Produces a single data row** or **Can produce more than one data row**, that is a sampled attribute group. If you select **Produces events**, then your attribute group sends an event to the monitoring environment each time that a row is received.

When you view sampled data in the Tivoli Enterprise Portal or IBM Cloud Application Performance Management console, you see the latest set of collected rows. The data that is displayed for an event attribute group is the contents of a local cache that is maintained by the agent. For event data, the agent adds the new entry to the cache until the size is reached when the oldest one is deleted. For sampled data, the agent replaces the contents of the cache every time you send data.

If you select **Produces events** or **Produces a single data row**, you must send only one row of data to the agent for that attribute group in each message. You can send as many events as you want, send each event in a separate message.

Normally sampled data is collected by the agent on request, but the socket client provides updated samples on its own schedule. You can update a sampled attribute group (single row or multiple row) as often as you require. When the data is requested by Tivoli Monitoring or IBM Cloud Application Performance Management, the agent provides the latest data.

If there are missing rows of data for the socket attribute group in the Tivoli Enterprise Portal or IBM Cloud Application Performance Management console, check the errors in the log file. Also, if the data in the attribute group is not as expected, check the errors in the log file. The socket data source attempts to process whatever it can from the input. For example, if the client sends three well-formed rows and one that is not valid (for example, malformed XML), you see:

- Three rows of data in the attribute group
- An error is logged for the malformed row in the agent's log file
- Since valid rows were returned, the Performance Object Status shows a status of `NO_ERROR`

For both event and sampled data, the data is sent to the agent as a single XML data flow from the socket client. Data that is sent from a socket client must always be terminated with a newline character: `'\n'`. The agent reads data until it sees the newline character and then an attempt is made to process what was received. Any data received that cannot be processed is discarded. The following is a sample of how you would send two rows of data to the agent for an attribute group named `abc`:

```
<socketData><attrGroup name="abc"><in><a v="1"/><a v="no"/><a v="5"/></in><in> \
<a v="3"/><a v="yes"/><a v="5"/></in></attrGroup></socketData>\n
```

This sample sends two rows of data to the agent where each row contains three attributes. The order of the attributes is important and must follow the order that is defined in your attribute group. The only exception to this is that the derived attributes must be skipped, regardless of where they are in your attribute group.

If the attribute group is defined in a subnode, then the subnode instance ID must be identified when data is sent to the agent. The subnode instance ID is identified by using the subnode attribute in the `socketData` element. A convention must be adopted for configuring subnode instance IDs for use by the socket client since the client cannot query instance IDs or configuration properties. Data sent to a subnode which is not configured is ignored.

Here is a sample:

```
<socketData subnode="app1"><attrGroup name="abc"><in><a v="1"/><a v="no"/><a v="5"/>
</in><in> \<a v="3"/><a v="yes"/><a v="5"/></in></attrGroup></socketData>\n
```

In this sample, the data is sent to the subnode with an instance ID equal to `"app1"`. `"app1"` is not the managed system name, but the instance identifier that is specified when the subnode instance is configured.

The following XML elements make up the socket data:

socketData
: The root element. It has one optional attribute that is called `subnode` that specifies the subnode instance ID.

attrGroup
: This element identifies the attribute group that the socket data is for. The `name` attribute is required and is used to specify the attribute group name.

in
: This element is required to identify a new row of data. All of the attribute values for a row of data must be children of the same `in` element.

a
: The a element identifies an attribute value. The v attribute is required and is used to specify the attribute value.

Sending errors instead of data

Sometimes the application that posts socket data might not be able to collect the data necessary for an attribute group. In this case, instead of sending data to the agent, an error code can be returned. The error code gives you a way to tell the monitoring environment about your problem. An example error is:

```
<socketData><attrGroup name="abc"/><error rc="1000"/></attrGroup></socketData>\n
```

The error code must be defined in the agent in a list that is common to all of the socket attribute groups. When the agent receives an error code, the defined error message is logged in the agent log file. In addition, the attribute group named Performance Object Status has an Error Code attribute is updated with the Error Code Type. The Error Code Type is defined for the error code you send.

For the previous example, you must define the Error Code Value of 1000 in the agent. See the following sample error code definition:

Table 283. Sample error code

Error Code Value	Error Code Type	Message
1000	APP_NOT_RUNNING	The application is not running

When the error code is sent, a message similar to the following is logged in the agent log file:

```
(4D7FA153.0000-5:customproviderserver.cpp,1799,"processRC") Received error code 1000
 from client. \Message: K1C0001E The application is not running
```

If you select Performance Object Status query from the Tivoli Enterprise Portal, the **Error Code** column for the row **abc** attribute group shows the value `APP_NOT_RUNNING` in that table.

Sending an error to a sampled attribute group clears any data that was previously received for that attribute group. Sending data to the attribute group causes the error code to no longer be displayed in the Performance Object Status attribute group. You can also send an error code of 0 to clear the error code from that table.

Sending an error to an attribute group that produces events does not clear the cache of events that were previously sent.

Handling `take action` requests

The socket client can register to receive `take action` requests from the agent when the action command matches a certain prefix. Any action that does not match is handled by the agent. The prefix must not conflict with actions that the agent is expected to handle, so use the agent product code as the prefix. `Take actions` provided with the Agent Builder are named after the data source that the `take action` uses. For example, the `JMX_INVOKE take action` operates on the JMX data source. Another example is the `SSHEXEC take action` which uses the SSH script data provider. Since these actions do not use the product code, the product code is a safe prefix to use as the `take action` prefix.

The socket client must be long running and leave the socket open. It must send a registration request for the prefix and listen for requests from the socket. The agent ensures that a timeout does not occur on the socket of a long-running client, even if no data is flowing. The following is a sample registration request:

```
<taskPrefix value="K42"/>\n
```

In this sample, any `take action` command that is received by the agent that begins with `"K42"` is forwarded to the socket client that initiated the registration. The following shows a sample `take action` request that the socket client might receive:

```
<taskRequest id="1"><task command="K42 refresh" user="sysadmin"/></taskRequest>\n
```

The `id` is a unique identifier that the agent uses to track requests that are sent to clients. When the socket client responds to the task, it must provide this identifier in the `id` attribute of the `taskResponse` element.

The socket client must process the action and send a response. A sample response is:

```
 <taskResponse id="1" rc="1"/>\n
```

If the action completes successfully, an `rc` attribute value of `0` is returned. The value of `rc` must be an integer, where any value other than 0 is considered a failure. The task return code value is logged to the agent log file and shown in the Take Action Status query that is included with the agent. The dialog that is displayed on the Tivoli Enterprise Portal after an action is run does not show the return code. That dialog indicates whether the `take action` command returned success or failure. The agent log or Take Action Status query must be viewed to determine the actual return code if a failure occurred.

It is the agent developer's responsibility to document, create, and import any actions that are supported by the socket clients that are used with an agent. If users send unsupported actions to the socket client, the client must be developed to handle those scenarios in an appropriate manner. If users define more actions that start with the registered prefix, they are passed to the client. The client must be developed to handle those scenarios in an appropriate manner.

There is a timeout that controls how long the agent waits for a response from the socket client. The setting is an environment variable that is defined in the agent that called `CDP_DP_ACTION_TIMEOUT` and the default value is 20 seconds.

Note: The error code messages that are defined for socket data source attribute groups are not used for `take actions`. You can return the same return code values. However, the agent does not log the message that is defined or affect the `Error Code` field in the Performance Object Status attribute group.

Encoding of socket data
The socket client encodes data that is sent to the agent.

It is important to be aware of how your socket client is encoding data that is being sent to the agent.

Special characters

Data sent to the agent must not contain any newline characters except at the end of each event or data sample. Newline characters that occur inside of attribute values must be replaced with a different character or encoded as shown in (Table 284 on page 1289). You must also be careful not to break the XML syntax with your attribute values. The following table shows the characters that occur in the attribute values that you encode:

Table 284. Characters to encode in attribute values

Character	Header
&	`&amp;`
<	`&lt;`
>	`&gt;`
“	`&quot;`
'	`&apos;`
`\n`	`&#10;`

Note: The agent uses the newline character to separate responses received from a client. Unexpected newline characters prevent data from being parsed correctly.

The agent does not contain a full-featured XML parser so you must not use special encoding for characters not in (Table 284 on page 1289). For example, do not encode `&#162;` or `&cent` in place of a cent sign `¢`.

Character sets

In addition to encoding special characters, the agent must know what code page was used to encode your data. Define each socket attribute group to indicate whether you are sending the data to the agent as **UTF-8** data or as **Local code page**. Be aware of how your client is sending data. If you use a client that is written in Java, specify **UTF-8** as the encoding on the writer you use to send data to the agent. Specify **UTF-8** as the **Code Page** for your attribute group. **Local code page** means the local code page of the agent. If the data is sent over a remote socket, it must conform to the local code page of the agent or use UTF-8.

Numeric Data

Be aware of how you are formatting your numeric attribute values. The numeric values that you send to the agent must not contain any special characters. One example is the thousands separator character. Other examples are currency symbols or characters that describe the units of the value. If the agent encounters a problem when it is parsing numeric data, it logs an error that indicates the issue. The

Performance Object Status Error Code is not set when an attribute fails to parse. The following is an example error message from the agent log:

```
(4D3F1FD6.0021-9:utilities.cpp,205,"parseNumericString") Invalid characters :00:04 \
found getting numeric value from 00:00:04, returning 0.000000
```

Note: For information about how a time stamp attribute must be formatted, see ("Time stamp" on page 1175).

Socket errors
Errors are written to the agent log file for problems that occur with data received from a socket client.

Other errors that are logged are `take actions` that return a value other than `0`. Error values that are sent by the socket client are logged along with the message associated with the error code.

The Performance Object Status for the attribute group is set when the socket client sends an error return code to the agent. Some other values can be seen in addition to the ones defined by the agent. The following table describes other "Error Code" values you are likely to encounter with socket attribute groups:

Table 285. Performance Object Status values

Error Code	Description
`NO_ERROR`	No error occurred. Indicates that there are no problems with the attribute group. Problems with a row of sampled data do not cause the state to change from `NO_ERROR`. You must validate the number of rows that are shown and the attribute values even when you see `NO_ERROR` as the error code.
`NO_INSTANCES_RETURNED`	A socket client sent no rows of data for a sampled attribute group. Not an error. It indicates that there are no instances of the resources that are being monitored by this attribute group.
`XML_PARSE_ERROR`	The agent failed to parse data that is received from the client. See the agent log for more details.
`OBJECT_CURRENTLY_UNAVAILABLE`	The client sent the agent an error code that was not defined in the global list of error codes.
`GENERAL_ERROR`	A problem occurred collecting data from the client, usually because the client did not reply to the request within the timeout interval. See the agent trace log for more details. The client can also specify GENERAL_ERROR as an error code, but it is better if a more detailed error code is defined.

Socket configuration
After you add a socket data source to your agent, you can configure the agent to accept data from a specified socket port.

About this task

After you add a Socket data source, the configuration is displayed on the **Runtime Configuration** page of the Agent Editor. The Socket configuration section contains the following property:

Table 286. Socket configuration property

Name	Valid values	Required	Description
Port number	0 or any positive integer The default value is 0	Yes	The port that the agent uses to listen on for data from socket clients. A value of 0 indicates that an ephemeral port is to be used.

The agent writes the value of the port that is being used to a file. Socket clients that run on the agent computer can later read this file to determine which port to connect to. The file that the port is written to is named `*kxx_instanceName*_cps.properties`, where: *kxx* is the three character product code of the agent and *instanceName* is the agent instance name for a multiple instance agent. If the agent is not a multiple instance agent, this part of the name is not included so the file name is `*kxx*_cp.properties`.

In Windows, the file is written to the `%CANDLE_HOME%\TMAITM6` directory for 32-bit installations or `%CANDLE_HOME%\TMAITM6_x64` for 64-bit installations. In UNIX, the file is written to `/tmp`.

Procedure

1. Optional: Set the environment variable `CDP_DP_HOSTNAME` to the host name or IP address of your network interface, if your system has multiple interfaces:
   a) Go to the Agent Editor **Agent Information** view and select **Environment Variables**.
   b) Click **Add** and select `CDP_DP_HOSTNAME` from the list of environment variables by using the `Name` field.
   c) Set the host name or IP address in the `Value` field.
2. Start your agent.

   When the agent is started, it binds to the interface that is defined by the `CDP_DP_HOSTNAME` environment variable. If `CDP_DP_HOSTNAME` is not set, the agent binds to the default host name.

   If you want the agent to bind to a defined port instead of an ephemeral port, you can set the configuration property **Port Number** (`CP_PORT`).

   To set the port number configuration property, use the following steps:

   a) Go to the Agent Editor **Runtime Configuration** view.
   b) In the **Runtime Configuration Information** pane select **Configuration for Socket** > **Socket** > **Port Number**
   c) Enter a port number value in **Default value**.

      If you do not enter a value, a value of 0 is used. A value of 0 indicates that an ephemeral port is used.

Remote socket port connection

You can configure your agent to accept data from a remote socket port. The agent must run on a system that has a network interface connection to a remote system.

Procedure

1. Set the value of the environment variable `CDP_DP_ALLOW_REMOTE` to YES by completing the following steps.
   a) Go to the Agent Editor **Agent Information** page and select **Environment Variables**.
   b) Click **Add** and select `CDP_DP_ALLOW_REMOTE` from the list of environment variables by using the **Name** field.
   c) Set the **Value** field to YES.
2. Follow the procedure that is detailed in "Socket configuration" on page 1290.

Restriction:

- The data that is sent between the socket application and the agent:
  - Must conform to the XML syntax defined for a socket data provider. For more information, see "Encoding of socket data" on page 1289.
  - Must be encoded in UTF-8.
  - Is sent in clear text (unencrypted). If the data contains sensitive information, the communication must be secured through an SSH tunnel or other mechanism outside the agent.
- The agent processes data that is received from any remote hosts, so the environment must be secured with appropriate firewall or network traffic filters.

Results

You can run code that implements a socket data provider on any system which can connect to the system where the agent is running.

Sample script for socket

This samples script demonstrates how a socket client might be written.

Perl sample

The following sample Perl script connects to a socket and sends data. This sample was written for an agent that runs on UNIX, with the product code `k00` and an attribute group called `SocketData`.

```
#!/usr/bin/perl -w
# SocketTest.pl
# A simple Agent Builder Socket client using IO:Socket
#------------------------

use strict;
use IO::Socket;

# Initialize socket connection to the agent
#-----------------------
my $host = '127.0.0.1';
my $port = 0;
# This sample is for an agent with the k00 product code.  The product code is
# used in the following line to find the file containing the port number to use.
open PORTFILE, "/tmp/k00_cps.properties" || die "Port file not found $!\n";
while (<PORTFILE>) {
    if (/^CP_PORT=([0-9]+)/) {
        $port = $1;
    }
}

if ($port == 0) {
    die "Could not find port to use to connect to agent.\n";
}

my $sock = new IO::Socket::INET( PeerAddr => $host, PeerPort => $port,
Proto => 'tcp'); $sock or die "no socket :$!";

# The following call sends 2 rows of data to the agent.  Each row contains 1
# String attribute and 3 numeric attributes.
syswrite $sock, "<socketData><attrGroup name=\"SocketData\"><in><a v=\"A message
from perl\"/>  \<a v=\"1\"/><a v=\"2\"/><a v=\"123\"/></in><in><a v=\"More from
perl\"/><a v=\"456\"/>  \<a v=\"123\"/><a v=\"789\"/></in></attrGroup>
</socketData>\n";

close $sock;
```

Testing socket attribute groups

You can test the socket attribute group that you created, within Agent Builder.

Before you begin

To test the attribute group, you need a socket client to send data. An example socket client that is written with perl script can be seen at "Sample script for socket" on page 1292

Restriction: Unlike most other attribute groups, you cannot test the socket attribute group while it is being created. You can test the attribute group when you complete its creation.

Procedure

1. Select an attribute group on the Agent Editor **Data Source Definition** page after agent creation and click **Test**. For more information about the Agent Editor, see "Using the Agent Editor to modify the agent" on page 1152.

   After you click **Test** in one of the previous two steps, the **Test Socket Client** window is displayed.
2. Optional: Set environment variables and configuration properties before you start testing.

   For more information, see "Attribute group testing" on page 1347.
3. Click **Start Agent**. A window indicates that the Agent is starting.
4. When the agent starts, it listens for socket data according to its configuration.
5. To test your agent 's data collection, you now generate socket data that matches the agents configuration.

   You can generate socket data by using a socket client.

   When the agent receives socket data that matches its configuration, it adds the data to its internal cache.
6. To simulate a request from Tivoli Enterprise Portal for agent data, click **Collect Data**.

   The **Test Socket Client** window collects and displays any data in the agent's cache since it was last started.
7. Click **Check Results** if something does not seem to be working as expected.

   The **Data Collection Status** window opens and shows you more information about the data. The data that is collected and displayed by the Data collection Status window is described in "Performance Object Status node" on page 1388
8. Stop the agent by clicking **Stop Agent**.
9. Click **OK** or **Cancel** to exit the **Test Socket Client** window. Clicking **OK** saves any changes that you made.

Related concepts

"Testing your agent in Agent Builder" on page 1346
After you use Agent Builder to create an agent, you can test the agent in Agent Builder.

Use the Java API to monitor data

You can define a data source to use the Java API to interact with a long-running application on the Java platform. The agent starts the application at startup and interacts with it periodically. When you build the agent, Agent Builder creates the source code for the application. You must customize the code to gather the correct data. Depending on the code, the data source can produce multiple data set that can contain a single row, multiple rows, or event data.

About this task

Use the Java API data source and the Java programming language to collect data that cannot be collected by using other Agent Builder data sources. The agent starts the Java application and sends a shutdown request when it is time to shutdown. The Java application must exit only when it is requested to do so.

An agent that contains Java API attribute groups interfaces with the Java application process. The Java application uses the Java Provider Client API to interface with the agent. For information about the API, see the Javadoc on the Tivoli Monitoring Knowledge Center. Using the Java API you can:

- Connect to the agent process and register for attribute groups that are supported by the Java application
- Receive and reply to a request for sampled data
- Send data asynchronously for an attribute group that produces events

- Send an error for an attribute group where data collection is failing
- Support attribute groups in subnodes with configured subnode instances
- Receive and reply to a "Take Action" request

Use the following procedure to create an attribute group which collects data in a Java application and sends it using the Java API. The procedure shows how to create a sample Java application to use as a starting point for your Java application.

Procedure

1. On the **Agent Initial Data Source** page or the **Data Source Location** page, click **Custom programs** in the **Monitoring Data Categories** area.
2. In the **Data Sources** area, click **Java API**.
3. Click **Next**.
4. On the **Java API Information** page, enter an Attribute group name.
5. Enter a help text for the attribute group.
6. Select whether the attribute group **Produces a single data row**, **Can produce more than one data row**, or **Produces events**. This choice affects the sample Java application that is created at the end of the wizard. For more information, see "Sending data" on page 1286.
7. Optional: Click **Advanced** to modify the advanced properties for the attribute group. **Advanced** is available when you select that the attribute group **Can produce more than one data row**, or **Produces events**.
8. Click **Next**.
9. On the **Attribute Information** page, specify the first attribute for the attribute group. For more information about creating attributes, see ("Creating attributes" on page 1169).
10. Select **Add additional attributes** and click **Next** to add other attributes to the agent. References to the attributes are incorporated into the sample Java application that is created at the end of the wizard.
11. Click **Next**.
12. On the **Global Java API Data Source Information** page, enter a Class name and a JAR file name.

    The class name is a fully qualified class name whose main method is called when Java is started. The sample Java application is created with the main Java method in this class.

    The JAR file is the archive that contains the Java classes that comprise the Java application. The JAR file is packaged and installed with the agent.
13. Optional: Define the error codes that the Java application can send, on the **Global Java API Data Source Information** page, **Error Codes** section. These error codes are sent by the Java application when it cannot collect data.

    Restriction: An error code has a limit of 256 characters. Only ASCII letters, digits, and underscores are allowed. No spaces are allowed.

    a) Click **Add** in the Error Codes section.

    b) In the **Java API Error Code Definition** window, enter a display value.

    c) Enter an internal value. The internal value must be an integer from 1,000 to 2,147,483,647.

    d) Define a message text for each error. You can use message text that was entered previously by selecting it from the list. Click **OK** to return to the **Global Java API Data Source Information** page.

    The message is logged in the agent log file.

    e) If no suitable message text is available, click **Browse** to set up the message text.

    The Messages (list) window is displayed. The message window lists messages that are defined in the agent. Until you define messages, the list remains blank. You can use **Edit** to alter a defined message and **Remove** to delete one or more messages that you defined.

f) In the Messages (list) window, click **Add** to see a **Message Definition** window. In the **Message Definition** window, you can type the text that describes the meaning of the new message and select the message type.

   Note: The message identifier is automatically generated for you.

g) Click **OK**.

h) The Messages (list) window is displayed with the new message. To verify the message and return to the **Global Java API Data Source Information** page, click **OK**.

14. Optional: In the **Supplemental Files** section of the **Global Java API Data Source Information** page, you can add files that are packaged with the agent and copied to the agent system on agent installation. The Java provider client API JAR file is not listed here; it is automatically copied to the agent system. The **File Type** column describes how each file is expected to be used. Three possible uses are described in the following table (Table 287 on page 1295). Click **Edit** to edit the imported file. For more information, see ("Editing a command file definition" on page 1253).

Table 287. File types for supplemental files

File type	**Description**
Executable	Select this option if you want to include an executable file with the agent. The agent does not use this file, but it is in the path for the Java application to use.
Library	Select this option if you to include a library with the agent. The agent does not use this file, but it is in the library path for the Java application to use.
Java resource	Select this option to include Java resources with the agent. The agent does not use this file, but it is in the class path for the Java application to use.

Note: When a Java resource supplemental file is added to the Agent Builder, the file is automatically added to the project class path. The Java compiler uses the supplemental file to resolve any references that your code has, to classes in the resource.

For information about where the Supplemental Files are installed with your agent, see ("New files on your system" on page 1362).

15. Optional: Create a filter to limit the data that is returned by this attribute group, if the data is sampled. Create a filter by clicking **Advanced**.

    Note: The data is sampled if you did not select **Produces events** on the **Java API Information** page.

    For more information about filtering data from an attribute group, see "Filtering attribute groups" on page 1178.

16. Optional: Add configuration properties to the subnode.

    If you are adding this data source to a subnode, the **Subnode Configuration Overrides** page is shown so you can add configuration properties to the subnode. At least one configuration property is needed under the subnode for the sample Java application to be created. At least one configuration property is needed because the sample uses a configuration property to distinguish one subnode instance from another.

17. Do one of the following steps:

    a) If you are using the **Agent** wizard, click **Next**. Complete the wizard as required.

    b) Otherwise, click **Finish** to save the data source and open the Agent Editor. Then, in the main menu, select **File** > **Save**.

At this point, Agent Builder creates the source code for the monitoring application. The code is located in the `src` subdirectory of the project directory. Edit this code to create your monitoring application.

What to do next

Select the correct operating systems on the **Java API Settings** page. Make this selection if this attribute group and the Java application, run on operating systems different from the operating systems that are defined for the agent. To open the page, click **Java API Settings** in the outline view or click **Global Settings** in the Agent Editor on any Java API attribute group page.

Note: Error codes and supplemental files can be updated later in the **Error Codes** and **Supplemental Files** sections of the **Java API Settings** page.

Running the Java application

Information about the initialization of the Java application and its dependencies

Initializing the Java application

The agent starts the Java application while the agent is starting and initializing. Configuration settings are used to control which Java run time is used to start the process. Java virtual machine arguments and the Java logging level can also be specified in the configuration. For more information about Java API configuration, see "Java API configuration" on page 1305. The Java process inherits the environment variables that are defined for the agent. Runtime configuration settings are also placed in the environment and can be queried by using API calls.

The Java application must be a long-running process. It must not terminate unless it receives a shutdown request from the API. If the Java application does terminate after it is registered with the agent, the agent will attempt to restart the Java application up to three times. If data collection is successfully resumed, this restart count is reset. The agent logs an error when a Java application terminates and when a restart is initiated.

Note: If the Java application terminates before attribute group registration is completed, no restart is attempted.

Dependencies

A Java application must use a Java runtime environment. The following versions of Java are supported:

- Oracle Corporation Java Version 5 or later
- IBM Corporation Java Version 5 or later

Java must already be installed on the agent system when the agent is configured and started. The JAR file that contains the API used to communicate with the agent is included with the agent `runtime` and included in the `classpath` of the JVM. Any additional JAR files that are needed by your Java application must be defined as Supplemental Files to the Java API attribute groups. Any supplemental files that have a *File Type* of *Java resource* are automatically added to the base `classpath` of the Java application, along with the Java API JAR file.

Any JAR files that are necessary for the runtime operation of the Java application that are not included with the agent, must be included in the *Classpath for external jars* configuration setting.

Generated sample Java application

A reference that describes the code the Agent Builder generates and the code you must add or replace for the resources you want to monitor.

When you create an agent with one or more Java API data sources, the Agent Builder generates Java application source code. The code is generated in the agent project and follows the structure of your agent. You must add your own Java code to the generated application. Your code collects data for sampled attribute groups, handles events to be posted to event-based attribute groups, reports errors if problems are encountered, and runs tasks. The generated application supplies the agent with data, but it is sample data, to be replaced with data obtained from the resources you want to monitor.

A sample agent is assumed that has the following characteristics:

- Product code: K91
- Java API Main class: `agent.client.MainClass`
- Agent data source structure as shown in (Figure 47 on page 1297):

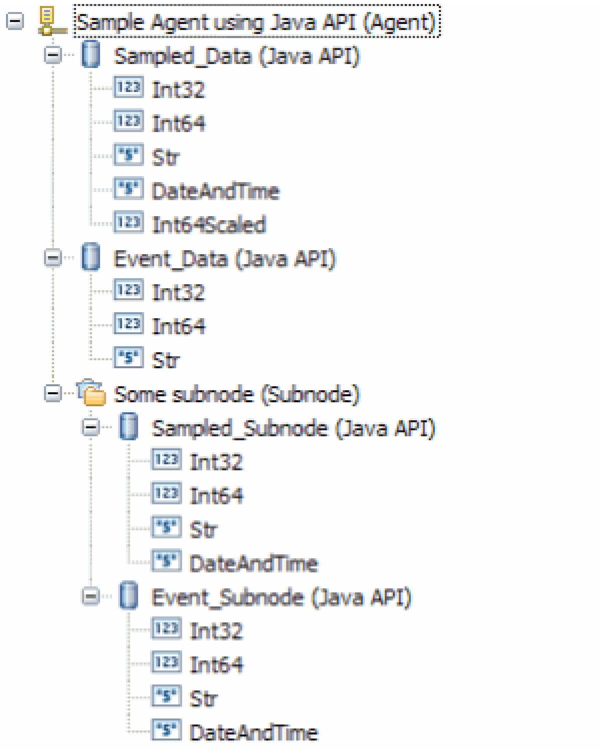


Figure 47. Sample agent structure

- *Some subnode* configuration property: `K91_INSTANCE_KEY`

Class structure

The generated Java application separates, to a great degree, code that interfaces with the agent from code that interfaces with the resources you are monitoring. It contains files that you modify, and files that you do not modify.

The following Java classes are created by the Agent Builder:

`MainClass` (`agent.client` package)
The class that you specified on the **Global Java API Data Source Information** page. This class contains a main method and a method that handles *take action* requests. This class inherits from the helper class described next. You must modify this class to interface with resources you want to monitor and the actions you want to take.

`MainClassBase` (`agent.client` package)
A helper class which initializes the connection to the server, registers attribute groups, and waits for requests from the server. Do not modify this class.

`Sampled_Data`, `Sampled_Subnode`, `Event_Data`, and `Event_Subnode` classes (`agent.client.attributeGroups` package)
There is one class for each Java API attribute group which handles data collection requests for the attribute group or generates events for the attribute group. These classes each inherit from one of the helper classes described next. You must modify these classes to gather data from the resources you want to monitor.

Sampled_DataBase, Sampled_SubnodeBase, Event_DataBase, and Event_SubnodeBase classes (agent.client.attributeGroups package)
Helper classes, one for each Java API attribute group, which define the structure of the attributes of the group in an internal class. Do not modify these classes.

ICustomAttributeGroup interface (agent.client.attributeGroups package)
An interface that defines public methods in each attribute group class. Do not modify this interface.

The classes which you can modify are never overwritten by the Agent Builder. The Agent Builder creates them only if they do not exist.

The helper classes and the interface are overwritten each time the Agent Builder is saved. As you modify and save the agent, the helper classes are updated to reflect any structural changes to the Java API attribute groups. The interface and helper classes contain a warning in the header that reminds you not to modify the file.

Initialization and cleanup

The main method in MainClass is called when the agent is started. It creates a MainClass instance and then enters the long-running method to receive and handle agent requests.

Most of the initialization and cleanup code must be added to MainClass. In the constructor, add initialization that is needed to create or access your resources. You might want to open connections to remote resources, create handles, or initialize data structures.

Before the agent terminates, the stopDataCollection method is called. If you want to close connections or cleanup before the Java application ends, add that code to the stopDataCollection method.

If initialization is needed only for a particular attribute group, that initialization can be added to the constructor of the attribute group class. Similarly, if any cleanup is needed only for a particular attribute group, that cleanup code can be added to the stopDataCollection method of the attribute group.

Any code in the Java application can use the logger object to write log entries. (The main helper class creates a protected logger object in its constructor. The attribute group helper objects create a protected reference to that logger in their constructors). The logger object uses the Java trace log utility. Errors and detailed trace information can be obtained from the trace log that is created by the logger. The trace information is important for troubleshooting problems with the provider.

When stopDataCollection is called, if you pass the cleanup work to another thread, wait for that thread to finish before you return from the stopDataCollection method. Otherwise, the cleanup work can be abruptly terminated when the process ends because the main thread completed.

One of the agent configuration settings is for the Java trace level. The following table shows the values that you can set in the JAVA_TRACE_LEVEL configuration property. If the API created the logger for you, the table shows the Level that is used by the logger.

Table 288. Java trace level options

Configured trace level	Java logging trace level	Description
Off	OFF	No logging is done.
Error	SEVERE	Trace problems that occurred in the Java application.
Warning	WARNING	Trace errors and potential errors.
Information	INFORMATION	Trace important information about the Java application.
Minimum Debug	FINE	Trace high-level details necessary to analyze the behavior of the Java application.

Table 288. Java trace level options (continued)

Configured trace level	Java logging trace level	Description
Medium Debug	FINER	Trace details about the program flow of the Java application.
Maximum Debug	FINEST	Trace all details about the Java application.
All	ALL	Trace all messages.

The name of the log file that is created by the Java application in this example is `k91_trace0.log`. If the agent is a multiple instance agent, the instance name is included in the log file name.

Note: Do not write messages to standard error or to standard out. On Windows systems, these messages are lost. On UNIX and Linux systems, this data is written to a file that does not wrap.

Collecting sampled attribute group data

The class for a sampled attribute group (one that collects one or more data rows) contains a `collectData` method, for example, `Sampled_Data.collectData`. This method is called whenever data is requested by the agent.

The helper class of the attribute group defines an inner class that is called Attributes. This class has one field for each attribute that is defined in your attribute group. Derived attributes are not included since they are calculated by the agent. The data types of attribute fields are Java equivalents to the Tivoli Monitoring attribute types, as shown in (Table 289 on page 1299).

Table 289. The data types of attribute fields and their IBM Tivoli Monitoring attribute type equivalents

Tivoli Monitoring type	Data type of attribute field
String	String
Numeric, 32 bit, no decimal adjustment	int
Numeric, 64 bit, no decimal adjustment	long
Numeric, non-zero decimal adjustment	double
Time stamp	Calendar

The `collectData` method must:

1. Collect the appropriate data from the resource that is being monitored.
2. Create an Attributes object.
3. Add the data to the fields of the Attributes object.
4. Call the `Attributes.setAttributeValues` method to copy the data to an internal buffer.
5. Repeat steps 1 - 4 as necessary for each data row. (You can skip steps 1 - 4 altogether and return no rows. In this case, the Error Code column of the Performance Object Status table has a value of NO_INSTANCES_RETURNED. For more information about error codes, see ("Error codes" on page 1302).
6. Call `AgentConnection.sendData`to send the data to the agent, or call `sendError` to discard data that is copied from calls to `setAttributeValues`and send an error code instead.

You must collect the data from your resource (Step 1), replacing the sample data that is used in the generated application.

To populate the Attributes object, you can pass the data in using the Attributes constructor (as is done in the generated application). Alternatively use the zero-argument constructor to create an Attributes object and then assign the fields of the attributes object to the attribute values you collected. Fields have the same name as the attributes, though they start with a lowercase letter.

Collecting sampled data for a subnode

If a sampled attribute group is in a subnode, there are presumably multiple resources that you are monitoring (a different one for each subnode). You must determine which resource to collect data from. There must be one or more configuration properties that identify which resource is being monitored.

For this example, it is assumed that one configuration property, K91_INSTANCE_KEY, contains a value that identifies the resource from which data must be collected.

Use the following steps to find the correct resource:

1. Get the instance ID of all configured subnodes by calling AgentConnection.getConfiguredSubnodeInstanceIDs. Each subnode that is configured has a unique instance ID.
2. For each instance ID, get the K91_INSTANCE_KEY configuration property by calling AgentConnection.getSubnodeConfigurationProperty.
3. Find the resource that is represented by the value in K91_INSTANCE_KEY.

These steps might be done in the collectData method before the series of steps that are detailed in ("Collecting sampled attribute group data" on page 1299).

Alternatively, you might want to do these steps in the attribute group class constructor and establish a direct mapping from instance ID to resource. An example attribute group class constructor is the Sampled_Subnode constructor. This procedure also gives you the opportunity to create handles or open connections that might be used through the life of the agent. Creating handles or open connections can make access to your resources more efficient.

The generated code creates sample resource objects of type MonitoredEntity in the constructor, and adds them to a configurationLookup map. You must remove the MonitoredEntity inner class, and replace the MonitoredEntity objects with objects that access your own resources. If you choose to do the entire lookup procedure in the collectData method, you can remove the configurationLookup map from the class.

If you choose to use the constructor, to map the subnode instance ID to your resource, the steps in the collectData method are:

1. Retrieve the instance ID of the subnode from the request parameter, by calling Request.getSubnodeInstanceID.
2. Retrieve the resource object from the map that is created in the constructor.
3. Perform the series of steps that are detailed in "Collecting sampled attribute group data" on page 1299 to send data to the agent.

An arbitrary subnode property is chosen in the Agent Builder example, in this case K91_INSTANCE_KEY. If not the correct property, or more than one property is needed to identify the correct resource, you must choose the properties to identify the resource.

Sending events

For attribute groups that generate events, there is no periodic call to a collectDatamethod. Events are sent by your application as your resource posts them.

As an example of producing events, the generated code for an event-based attribute group creates and starts a thread which runs from an internal class named SampleEventClass. The event-based attribute group that is used in the example is the Event_Dataclass. The thread periodically wakes up and sends an event. If you want to periodically poll your resource for events, you can use the structure of the Event_Data class as it was generated:

1. From the Event_Data constructor, create and start a thread.
2. In the run method of the thread, loop until the agent terminates.
3. Sleep for a time before you check for events. You might want to change the polling interval of 5000 milliseconds to a number that makes sense for your agent.

4. Determine whether one or more events occurred. The generated application does not check, but always posts a single event.
5. For each event that must be posted, get the event data to be posted.
6. Create and populate the Attributes object (like the `collectData` method did for a sampled attribute group).
7. Call the `Attributes.sendEventData` method. Events consist of a single row, so only a single event can be sent at a time.

Alternatively, if you are working with a Java API that reports events from its own thread, you can initialize that thread in the `Event_Data` constructor. You can also register your own event-handling object with the event-handling mechanism of your resource. In your event handler, use the following steps:

1. Get the event data to be posted.
2. Create and populate the Attributes object.
3. Call the `Attributes.sendEventData` method.

In this case, you do not have to create your own thread in the `Event_Data` class nor would you need the `SampleEventClass` class.

Sending events in a subnode

When an event is detected for a subnode attribute group, the Java application must post the event to the correct subnode.

For this example, it is assumed that one configuration property, K91_INSTANCE_KEY, contains a value that identifies an instance of a resource which can produce events. It is also assumed that the value of the K91_INSTANCE_KEY property is retrieved along with data to be posted in the event. To do retrieve the property and data, the Java application does the following steps:

1. Gets the event data to be posted, along with the “instance key”.
2. Creates and populates the Attributes object.
3. Gets a list of all configured subnode instance IDs by calling `AgentConnection.getConfiguredSubnodeInstanceIDs`.
4. For each subnode instance, fetches the value of K91_INSTANCE_KEY by calling `AgentConnection.getSubnodeConfigurationProperty`.
5. When the value of K91_INSTANCE_KEY is found which matches the value that is obtained with the event data, remembers the corresponding subnode instance ID.
6. Calls `Attributes.sendSubnodeEventData`, passing the remembered subnode instance ID.

The generated application does not do the lookup described in steps 4 and 5, but instead posts an event to the attribute group of every subnode. This behavior is probably not the correct one for a production agent.

`Take action` commands

`Take action` commands are defined either in the Tivoli Enterprise Portal or by using the `tacmd createaction` command. The actions can be imported into the agent's Agent Builder project so that they are created when the agent is installed. For more information about importing `take action` commands, see (“Importing application support files” on page 1371).

The generated Java application registers for any actions that begin with the product code of the agent, for example, `K91Refresh`. This registration is done in the main helper class (`MainClassBase`) from the `registerActionPrefix` method. If you want to register other prefixes, or not register for actions at all, override the `registerActionPrefix` in (`MainClassBase`).

When the agent wants to run an action which starts with a prefix that your agent registered, the `MainClass.takeAction` method is called. You add code to call `Request.getAction()`, do the appropriate action, and then call `AgentConnection.sendActionReturnCode` to send the return code from your action. A return code of 0 means the action is successful, any other return code means the action failed.

Handling exceptions

The collectData and takeAction methods can throw any Java exception, so you can allow your collection code to throw exceptions without catching them. The handleException method (for collectData) or handleActionExceptionmethod (for takeAction) is called when the helper class gets the exception.

For collectData exceptions, you must call AgentConnection.sendError when an exception occurs or when there is a problem in data collection. The generated application passes an error code of GENERAL_ERROR. However, you must replace this error code with one defined by your agent that best describes the problem that was encountered. For more information about adding error codes, see Step (“13” on page 1294).

For takeAction exceptions, you must call AgentConnection.sendActionReturnCode with a non-zero return code.

Some of the AgentConnection methods throw exceptions that are derived from com.ibm.tivoli.monitoring.agentFactory.customProvider.CpciException. The handleException method is not called if a CpciException is thrown during the collecting of data as the helper class handles the exception.

Note: If you choose to catch your exceptions inside the collectData method rather than using the handleException method, ensure any CpciException is rethrown. You ensure CpciException is rethrown so it can be handled by the base class.

Error codes

A typical response to an exception or other resource error is to send an error code to the agent by calling the AgentConnection.sendError method. An error for an event-based attribute group can be sent at any time. An error for a sampled attribute group can be sent only in response to a collect data request, and in place of a sendData call.

If you send an error to the agent, the following happens:

1. An error message is logged in the agent trace log. This error message includes the error code and the message that is defined for that error code.
2. There is a Performance Object Status query that can be viewed to obtain status information about your attribute groups. The Error Code column is set to the Error Code type defined for the error you sent. The error status clears after data is successfully received by the agent for the attribute group. If you reply to a collect data request with a sendData call but you included no data rows, you get NO_INSTANCES_RETURNED in the Error Code column.

The following table describes some error codes that are internal to the agent that you can expect to see in certain situations:

Table 290. Internal error codes for the agent

Error Code	Description
NO_ERROR	There are no problems with the attribute group currently.
NO_INSTANCES_RETURNED	The Java application responded to a data collection request but provided no data. Not providing data is not an error. It generally indicates that there are no instances of the resource that are being monitored by the attribute group.

Table 290. Internal error codes for the agent (continued)

Error Code	Description
OBJECT_NOT_FOUND	The agent tried to collect data for an attribute group that is not registered through the client API. This error can mean that the application failed to start or did not initiate the attribute group registration when the agent tried to collect data.
OBJECT_CURRENTLY_UNAVAILABLE	The application sent the agent an error code that is not defined in the global list of error codes.
GENERAL_ERROR	A problem occurred collecting data from the application, usually because the application did not reply to the request within the timeout interval. See the agent trace log for more details. The application can also specify GENERAL_ERROR as an error code, but it is better if a more detailed error code is defined.

Changes to the agent

Certain changes to the agent require you to make corresponding changes to the Java application. If the structural changes are complex, you can delete any or all of the Java source files before you save the agent. You can also delete the files if you want to start over without the customizations you made,

The following table describes required modifications to the Java application source files after certain changes are made in the Agent Builder when the agent is saved.

Table 291. Changes to an agent that require modifications to the Java source

Agent change	What Agent Builder does	Manual changes that are needed in the Java source
Change of the main class package name	• Generates all classes in the new package structure. • Removes all helper classes from the old package.	• Port main and attribute group class content from the classes in the old package to the classes in the new package. • Remove the classes from the old package after migration is complete.
Change of the main class name	• Creates new main classes. • Removes old main helper class.	• Port main class content to the new class. • Update references to the class name from the attribute group classes.
Addition of a Java API attribute group	• Creates classes for the new attribute group. • Adds registration for the new attribute group in the main helper class.	Overwrite sample code with custom logic in the attribute group class.

Table 291. Changes to an agent that require modifications to the Java source (continued)

Agent change	What Agent Builder does	Manual changes that are needed in the Java source
Removal of a Java API attribute group	Removes registration from the main helper class.	• Remove the attribute group class or port customized logic to some other class. • Remove the attribute group helper class.
Renaming of a Java API attribute group	• Creates classes for the new name of the attribute group. • Updates registration for the renamed attribute group in the main helper class.	• Port customized logic in the attribute group class with the old name to the attribute group class with the new name. • Remove the attribute group class with the old name. • Remove the attribute group helper class with the old name.
Addition of an attribute to a Java API attribute group	Updates the Attributes inner class in the attribute group helper class.	Collect data for the new attribute in the attribute group class.
Removal of an attribute from a Java API attribute group	Updates the Attributes class in the attribute group helper class.	Remove data collection for the former attribute in the attribute group class.
Renaming of an attribute in a Java API attribute group	Updates the attribute name in the Attributes class in the attribute group helper class.	Update any references to the attribute name in the Attributes class (often there are no references because the Attributes constructor, with positional arguments, is used).
Reordering of attributes in a Java API attribute group	Updates the attribute order in the Attributes class in the attribute group helper class.	Update the argument order in any calls to the Attributes constructor.

Some of the changes that are mentioned in the previous table can be streamlined if you use the Eclipse Refactor - Rename action. Use this action on all the affected names (including helper class names) before you save the changed agent.

Use of the Java API

The Java API is used throughout the generated Java application to communicate with the agent. Often your only direct interaction with the Java API is to modify a parameter of an existing method call. For example, changing a posted error code from `GENERAL_ERROR` to an error code defined in your agent.

If you want to do more extensive coding with the Java API, you can view Javadoc from the Eclipse text editor. You can view Javadoc while you edit the Java code by doing the following steps:

1. Highlight a package, class, or method name from the API.
2. Press **F1** to open the Eclipse Help view.
3. Select the Javadoc link.

You can also see a brief description from the Javadoc by hovering over a class or method name. Javadoc for the API can also be found on the Tivoli Monitoring Knowledge Center, see Javadoc.

The classes for the Java API are in `cpci.jar`. The `cpci.jar` file is automatically added to the Java Build Path of the project when an agent which contains a Java API attribute group is created. The file is also added when an agent that contains a Java API attribute group is imported. The file is also added when a Java API attribute group is added to an existing agent. The`cpci.jar` is also automatically packaged with each agent that contains a Java API attribute group and added to the CLASSPATH of the Java application.

Java API configuration

When you define a Java API data source in your agent, some configuration properties are created for you.

If you define a Java API data source in your agent, the agent must use Java to connect to the Java API server. Java configuration properties are added to the agent automatically. The following Java configuration properties are specific to the agent runtime configuration:

Table 292. Java configuration properties

Name	Valid values	Required	Description
Java home	Fully qualified path to a directory	No	A fully qualified path that points to the Java installation directory.
Java trace level	Choice	Yes	Use this property to specify the trace level that is used by the Java providers.
JVM arguments	String	No	Use this property to specify an optional list of arguments to the Java virtual machine.
Class path for external jars	String	No	Path containing any JAR files that are not included with the agent, but are necessary for the runtime client operation.

These configuration variables are available on the **Runtime Configuration Information** page of the Agent Editor under **Configuration for Java Virtual Machine (JVM)**, and **Configuration for Java API**.

Testing Java application attribute groups

You can test the Java application attribute group that you created, within Agent Builder.

Before you begin

Restriction: Unlike most other attribute groups, you cannot test the Java application attribute group while it is being created. You can test the attribute group when it is added to the agent and the agent is saved. Saving the agent causes the Java code to be generated for the attribute group.

Procedure

1. Select an attribute group on the **Agent Editor Data Source Definition** page after agent creation and click **Test** .

   For more information about the Agent Editor, see “Using the Agent Editor to modify the agent” on page 1152

   After you click **Test** in one of the previous two steps, the **Test Java Client** window is displayed.
2. Optional: Set environment variables, configuration properties, and Java information before you start testing. For more information, see “Attribute group testing” on page 1347. For more about default Java runtime configuration properties, see “Java API configuration” on page 1305.

3. Click **Start Agent**. A window indicates that the Agent is starting.
4. To simulate a request from Tivoli Enterprise Portal or SOAP for agent data, click **Collect Data**.

   The agent monitors the Java Client for data. The **Test Java Client** window displays any data that is returned.
5. Optional: Click **Check Results** if the returned data is not as you expected.

   The **Data Collection Status** window opens and shows you more information about the data. The data that is collected and displayed by the Data collection Status window is described in "Performance Object Status node" on page 1388
6. Stop the agent by clicking **Stop Agent**.
7. Click **OK** or **Cancel** to exit the **Test Java Client** window. Clicking **OK** saves any changes that you made.

Related concepts

"Testing your agent in Agent Builder" on page 1346
After you use Agent Builder to create an agent, you can test the agent in Agent Builder.

Creating data sets from existing sources

When at least one data set exists, you can create a new data set using data from an existing data set.

The option to create a new data set is available on the **Agent Initial Data Source** page and on the **Data Source Location** page. You can create a data set by using existing data sources in the following ways:

1. Joining data from two existing data sets (attribute groups). For more information, see "Joining two attribute groups" on page 1306.
2. Filtering data from an existing data set (attribute group). For more information, see "Creating a filtered attribute group" on page 1311.

Tip: The option to join two data sets is available only after two or more data sets are created.

Joining two attribute groups

Create an attribute group from two other attribute groups.

About this task

Joining attribute groups is most useful when the agent collects data from two different types of data sources. For example, the agent might collect data WMI and PerfMon, or SNMP and script data sources. Each set of attributes might be more useful when used together in one Tivoli Enterprise Portal view.

For example, assume that your attribute groups are defined as follows:

```
First_Attribute_Group
    index integer
    trafficRate    integer
    errorCount    integer
```

```
Second_Attribute_Group
    index2    integer
    name string
    traffic string
```

One definition provides you with counters (like Perfmon) and the other provides you with identification information. Neither attribute group is useful to you by itself. However, if you can combine both attribute groups by using the index to match the appropriate rows from each, you have a more useful attribute group. You can use the combined attribute group to display the name, type, and metrics together.

This same mechanism can be used to add tags to information collected through normal attribute groups. The information can then be more easily correlated in an event system when a problem is detected. For example, a company wants to manage all its servers by collecting common data and by using common situations to monitor the health of the servers. It also wants to be able to identify the servers with more information that tells it what application is running on a particular server. It wants to have control of the values that are used on each server, but it does not want to create different agents for each application. It can accomplish this control by creating an additional attribute group in its single agent as follows:

```
Application_Information
    application_type     integer
    application_name    string
    application_group    string
```

This attribute group would be defined as a script attribute group that gathers its values from agent configuration. You can specify different values for each agent instance and use one agent to manage all of their systems. This attribute group would then be joined to all the source attribute groups where this application information might be needed. The information is then available in the Tivoli Enterprise Portal, situations, events, and warehoused data.

When you join two attribute groups, a third attribute group is created. This attribute group contains all the attributes that are contained within the source attribute groups.

The results of a join operation vary depending on the number of rows that each source attribute group supports. If both attribute groups are defined to return only a single row of data, then the resulting joined attribute group has one row of data. The single row contains all attributes from both source attribute groups.

Table 293. source attribute group one (single row)

Attribute1	**Attribute2**	**Attribute3**
16	some text	35

Table 294. source attribute group 2 (single row)

Attribute4	**Attribute5**	**Attribute6**	**Attribute7**
5001	more data	56	35

Table 295. Resulting join

Attribute1	**Attribute2**	**Attribute3**	**Attribute4**	**Attribute5**	**Attribute6**	**Attribute7**
16	some text	35	5001	more data	56	35

Suppose that one source attribute group is defined to return only one row (single-row) while the other can return more than one row (multi-row). The resulting joined attribute group contains the same number of rows as the multi-row source attribute group. The data from the single-row attribute group is added to each row of the multi-row attribute group.

Table 296. source attribute group one (single row)

Attribute1	**Attribute2**	**Attribute3**
16	some text	35

Table 297. source attribute group two (more than one row)

Attribute4	**Attribute5**	**Attribute6**	**Attribute7**
user1	path1	56	35
user2	path2	27	54
user3	path3	44	32

Table 298. Resulting join

Attribute1	Attribute2	Attribute3	Attribute4	Attribute5	Attribute6	Attribute7
16	some text	35	user1	path1	56	35
16	some text	35	user2	path2	27	54
16	some text	35	user3	path3	44	32

Finally, assume that both source attribute groups are defined to return more than one row. You must identify an attribute from each of the source attribute groups on which to join. The resulting attribute group contains data rows where the attribute value in the first attribute group matches the attribute value from the second attribute group.

Table 299. source attribute group one (more than 1 row)

Attribute1	Attribute2	Attribute3
16	some text	35
27	more text	54
39	another string	66

Table 300. source attribute group 2 (more than 1 row)

Attribute4	Attribute5	Attribute6	Attribute7
user1	path1	56	35
user2	path2	27	54
user3	path3	44	32

Table 301. Resulting join (joining on Attribute3 and Attribute7)

Attribute1	Attribute2	Attribute3	Attribute4	Attribute5	Attribute6	Attribute7
16	some text	35	user1	path1	56	35
27	more text	54	user2	path2	27	54

With Agent Builder, you can also join user-defined attribute groups to the Availability attribute group if there are any availability filters defined in your agent. For more information about the data that is contained in the Availability attribute group, see ("Availability node" on page 1383).

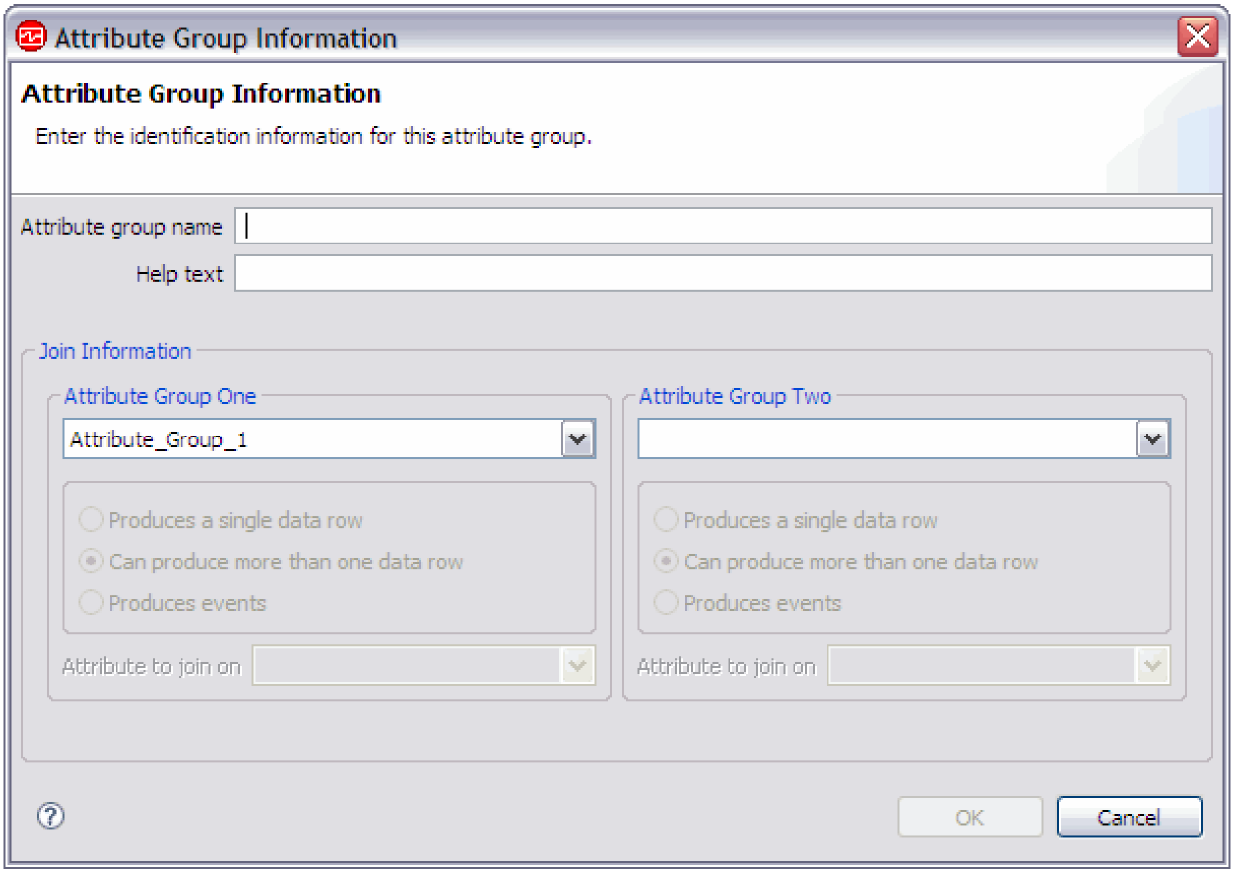
You can create this type of attribute group by accessing the menu on the data sources tree by right-clicking and then selecting **Join Attribute Groups**.

Procedure

1. On the **Data Source Definition** page, right-click one of the attribute groups you would like to join and select **Join Attribute Groups**.

   This option is only visible if there are at least two attribute groups defined. Having an availability filter defined counts as having an attribute group defined.

   The **Attribute Group Information** page is displayed.



Attribute Group Information

Enter the identification information for this attribute group.

Attribute group name

Help text

Join Information

Attribute Group One: Attribute_Group_1

○ Produces a single data row
● Can produce more than one data row
○ Produces events

Attribute to join on

Attribute Group Two

○ Produces a single data row
● Can produce more than one data row
○ Produces events

Attribute to join on

Figure 48. ***Attribute Group Information*** *pageAttribute Group Information window*

2. In the **Join Information** area, select the two attribute groups you would like to join. Select the attribute groups by choosing from the groups available in the **Attribute Group One** and **Attribute Group Two** lists.

   For each attribute group, either **Produces a single data row** or **Can produce more than one data row** is selected for you. This selection is locked and depends on how the source attribute groups were originally defined.

   Note: There are restrictions on which attribute groups can be joined:

   - You cannot join an attribute group in one subnode type to an attribute group in another subnode type.
   - You can join only an event attribute group to a single row non-event attribute group.

   a) Select the attribute that you want to join on for each attribute group when both attribute groups show **Can produce more than one data row**, under **Attribute to join on**.

   The **Attribute group name** and **Help** fields are filled for you using information from the chosen attribute groups. If you want to, you can change these entries.

3. Click **OK**.

Results

The joined attribute group that you created is added to the **Attribute Group Information** area of the **Data Source Definition** page

Manipulating attributes in joined attribute groups

Using attributes in joined attribute groups can impose rules on how those attributes are manipulated.

Deleting an attribute group

An attribute group cannot be deleted if it is referenced in a joined attribute group unless the joined attribute group is also being deleted.

Deleting an attribute

An attribute cannot be deleted if its parent attribute group is referenced in a joined attribute group and one of these statements is true:

- The attribute is defined as a join attribute in the joined attribute group.
- The attribute is used in any derived attribute in the joined attribute group.

Joined attributes cannot be deleted. Only derived attributes, if any are added, can be deleted from the joined attribute group.

Reordering attributes

The order of the joined attributes is fixed by the order of the source attributes. The joined attribute list cannot be reordered. Only derived attributes, if any, can be reordered.

When the version of an agent, is committed, source, and derived attributes cannot be reordered or removed. Attributes added in a new version of the agent, whether source or derived attributes, will come after all committed attributes. For more information, see "Committing a version of the agent" on page 1166.

Adding an attribute

New joined attributes cannot be explicitly added. Only derived attributes can be explicitly created.

Removing availability filters

The last availability filter cannot be removed if the Availability attribute group is referenced in a joined attribute group.

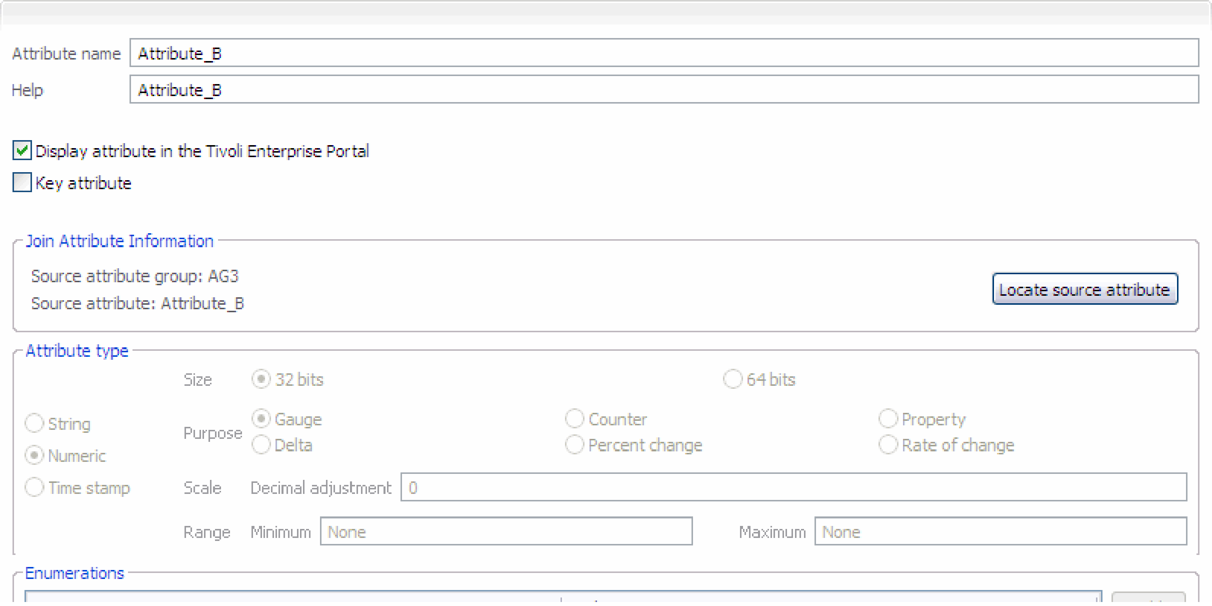
Joined attributes

Manipulate information that relates to joined attributes

Procedure

- Change the attribute name and help text of the joined attribute can be changed so that they are different from the source attribute:
  a) Select the attribute in the joined attribute group in the **Attribute Group Information** pane of the **Data Source Definition** page.
  b) Enter the new name and help text.
- The joined attribute can be shown or not shown on the Tivoli Enterprise Portal by selecting or clearing the **Display attribute in the Tivoli Enterprise Portal** check box. The check box is in the **Joined Attribute Information** section of the **Data Source Definition** page. This choice is irrespective of whether the source attribute is shown on the Tivoli(r) Enterprise Portal.
- Any attribute or combination of attributes (that are shown on the Tivoli Enterprise Portal) can be marked as key attributes by selecting the **Key attribute** check box. This choice is independent of whether the attributes are key attributes in the source attribute groups. The choice is also independent of whether the source attributes are shown on the Tivoli(r) Enterprise Portal.

- Attribute type information for joined attributes is taken from the source attributes and cannot be changed in the joined attribute. In the **Joined Attribute Group Information** section of the agent editor (Figure 49 on page 1311), click **Locate source attribute** to go to the source attribute.



Attribute name: Attribute_B

Help: Attribute_B

☑ Display attribute in the Tivoli Enterprise Portal

□ Key attribute

Join Attribute Information

Source attribute group: AG3

Source attribute: Attribute_B

Attribute type

Size: ● 32 bits ○ 64 bits

○ String ● Numeric ○ Time stamp

Purpose: ● Gauge ○ Delta ○ Counter ○ Percent change ○ Property ○ Rate of change

Scale: Decimal adjustment 0

Range: Minimum None Maximum None

Figure 49. Locating source attribute information

Any changes to the source attribute groups are reflected in the joined attributes. If the source attribute groups change, those attributes are automatically updated under the joined attribute group. This automatic update also occurs if a different attribute group is set as the source attribute group. Changes to a source attribute type are copied to the joined attribute. Changes to a source attribute name or help text are copied to the joined attribute. However, such source attribute changes are not copied after you change the name or help text of a joined attribute.

Creating a filtered attribute group

Create a filtered attribute group (data set) by filtering rows of data from an existing attribute group. If an existing data set returns multiple rows, you can create a filtered group returning one row for use with IBM Cloud Application Performance Management.

About this task

A filtered attribute group has the same columns as the source attribute group, but can exclude some of the rows. It uses a selection formula to determine which rows to include.

To provide status and summary information for Cloud APM, you need to use a data set that returns a single row. For details, see "Preparing the agent for Cloud APM" on page 1342. If the source information is in a data set that returns multiple rows, you can create a filtered attribute group that returns a single row.

For example, the process, Windows service, and command return code data sources provide information as rows in the single Availability data set. You can create a filtered attribute group, using the NAME field in the selection formula. The group includes status for the necessary application. Define it as returning one row. Then you can use this attribute group as the summary data set for Cloud APM.

A filtered attribute group is also useful when a base data source query returns data that you prefer to divide into separate groups. Examples of such data sources are Windows Performance Monitor, SNMP and WMI.

For example, assume that a data source can return the following data:

```
Name    Type  Size  Used  Free
Memory  MEM      8     4     4
```

```
Disk1   DISK   300   200   100
Disk2   DISK   500   100   400
```

This is a table that reports on the storage that exists on the system and it includes both memory and disk space. You might prefer to break down the table into memory and disk as separate tables. You can break down the table by creating two base attribute groups. Each of these base attribute groups collects the same data and filters out the rows you do not want. However, that is not the most efficient way to do things. Instead, you define one base attribute group that returns both the memory and disk usage data together. Next, define two filtered attribute groups. Each uses the same base table as its source. One includes a filter where `Type=="MEM"` and the other includes a filter where `Type=="DISK"`.

In the example, for the filtered attribute group where `Type=="MEM"`, the returned data is:

```
Name    Type  Size   Used  Free
Memory  MEM      8      4     4
```

and where `Type=="DISK"`, the returned data is:

```
Name    Type  Size  Used  Free
Disk1   DISK  300   200   100
Disk2   DISK  500   100   400
```

Note: Attributes groups whose data is event-based cannot be used to create filtered attribute groups. Only attribute groups whose data is sampled can be used.

Procedure

1. Click **Existing data sources** in the **Monitoring Data Categories** area on the **Agent Initial Data Source** page or the **Data Source Location** page

   Note:

   - You reach the **Agent Initial Data Source** page by using the new agent wizard. For more information, see "Creating an agent" on page 1149.
   - You can reach the **Data Source Location** page by right-clicking an agent in the **Data Source Definition** page of the **Agent Editor** and selecting **Add Data Source**.
2. Select **Filter an attribute group's data rows** in the **Data Sources** area.
3. Click **Next**

   The **Filter Information** page is displayed.
4. Select a **Source attribute group** from the list.
5. Enter a **Selection formula** to filter the data from the attribute group you selected.
   For example, in the **Filter Information** page that is shown earlier, the selection formula filters data rows where the Type attribute is equal to `"DISK"`. Data rows whose Type attribute does not match `"DISK"` are discarded. The selection formula that you enter must evaluate to a Boolean result, true, or false.

   Note: In the **Filter Information** page, you can click **Edit** to enter or modify the formula by using the Formula Editor. For more information about the Formula Editor, see "Formula Editor" on page 1178.
6. Click **Next**.
7. Select **Produces a single data row** or **Can produce more than one data row**.
   a) If you selected **Can produce more than one data row**, select a key attribute or attributes from the list.
8. Click **Finish**.

Creating a navigator group

In an IBM Tivoli Monitoring environment, use Navigator groups to group several related data sources (attribute groups) together so that workspaces can be created that show views that combine the data sources. You can create a navigator group while you create an agent by using the New Agent wizard at the base agent level. You can also create a navigator group while you define a subnode by using the New Agent Component wizard.

About this task

For example, you might be able to collect file system data from more than one data source. It can be useful to create one workspace that shows views of all file system data from those different data sources.

Navigator groups are also a good way to hide data sources on the Tivoli Enterprise Portal. You might decide that metrics collected from two data sources are most useful if the data sources are joined to create a new combined data source. You want to see only the combined data in the Joined data source. You can create a navigator group that contains all three data sources and create a workspace that contains views to display only the combined data source. The two original data sources are effectively hidden from view in the Tivoli Enterprise Portal. See "Creating data sets from existing sources" on page 1306 for information about joining data sources.

Note: When you group data sources in a navigator group, Tivoli Monitoring does not associate a query with the navigator group. It is assumed that you define a default workspace for the navigator group to display the data sources in a useful format.

A navigator group can be defined in the base agent or in a subnode. A navigator group cannot contain another Navigator group.

Navigator groups have no effect in an IBM Cloud Application Performance Management environment.

Procedure

1. Take one of the following steps:
   - When creating a new agent using the **Agent** wizard, on the **Agent Initial Data Source** page, click **Data source groupings** in the **Monitoring Data Categories** area.
   - With an existing agent, take the following steps in the Agent Editor:
     a. Click the **Data Sources** tab to open the **Data Source Definition** page.
     b. Select the agent and click **Add to selected**.
     c. On the **Data Source Location** page, in the **Monitoring Data Categories** area, click **Data source groupings**.
2. In the **Data Sources** area, click **A navigator group**.
3. Click **Next**.
4. On the **Navigator Group Information** page, type the navigator group name and the text for the Help you want associated with the name, and click **Next**.

   Note: Agent Builder automatically creates navigator groups in certain situations. The following navigator group name is reserved:

   - Availability

   .
5. On the **First Navigator Group Data Source** page, select the first source of monitoring data for the new navigator group. Click a category in the **Monitoring Data Categories** list and a data source in the **Data Sources** list. Then, click **Next**.

   Tip: You can create the data source as usual. Alternatively, click **Existing data sources** and choose to move one or more data sources that you already created into the navigator group.
6. If you want to create a data source within a navigator group, on the **Data Source Definition** page, select the navigator group, and click **Add to Selected**.
7. If you want to move existing data sources into the navigator group, on the **Data Source Definition** page, select the navigator group, and click **Add to Selected** and on the **Navigator Group Data Source** page, select **Existing data sources**. In the **Currently Defined Data Sources** page, select the data sources.

8. If you want to remove a data source from a navigator group, do one of the following steps on the **Data Source Definition** page:
   - Select the data source, and drag it to the root of the data sources tree.
   - Select the data source, and clicking **Remove**.
9. If you want to create a navigator group, do one of the following steps on the **Data Source Definition** page:
   - Click **Add to Agent**.
   - Select a subnode and click **Add to Selected**.

Using subnodes

You can use subnodes to monitor multiple application components from a single agent instance.

You can build a single agent that accomplishes the following tasks by using subnodes:

- Monitors each instance of a software server that is running on a system instead of having to use separate instances of the agent, one per software server instance.
- Monitors several different remote systems instead of having to use separate instances of the agent, one for each remote system.
- Monitors several different types of resources from one agent instead of having to build and deploy several different agents.
- In IBM Tivoli Monitoring, displays an additional level in the Tivoli Enterprise Portal physical Navigation tree that allows further grouping and customization. Moreover, you can define Managed System Groups for another level of granularity with situations.
- In IBM Cloud Application Performance Management, provides several different resources, displaying different summary and detail dashboards. Subnode resources can be displayed as peers or subcomponents of the agent resource. You can include these resources in applications independently.

You can create subnode types in Agent Builder. Each type must correspond to a different type of resource that an agent can monitor. Add data sources and data sets to the subnode type for a particular monitored resource.

When you deploy the agent on a monitored host and configure it, you can create one or more instances of each subnode type. Each instance of a subnode must correspond to an instance of a server, a remote system, or whatever resource the subnode type was designed to monitor. All subnode instances of a single subnode type have attribute groups and workspaces that have an identical form. However, each subnode instance has data that comes from the particular resource that is being monitored.

When you configure the agent on the monitored host, you can determine the number of subnode instances. Some configuration data can apply to the agent as a whole, but other configuration data applies to a single subnode instance. Configure each subnode instance differently from the other subnode instances so that they do not monitor the exact same resource and display the exact same data.

In an IBM Tivoli Monitoring environment, a subnode instance is displayed within the agent in the Navigation Physical view in the Tivoli Enterprise Portal. Workspaces display the data that is produced by a subnode instance and situations can be distributed to one or more instances of a subnode. A managed system list is automatically created that contains all instances of the subnode, just like the Managed System List that is created for an agent.

In an IBM Cloud Application Performance Management environment, you can display both agent and subnode instances as monitored resources. Each subnode instance becomes a separate resource. For details, see "Subnodes in IBM Cloud Application Performance Management" on page 1319.

Because agents built with Agent Builder create the subnode instances that are based on configuration values, these subnodes have the same life span as the agent. There is still just one heartbeat that is done for the agent, not a separate heartbeat for each subnode. Therefore, by using subnodes you can significantly increase the potential scale of the monitoring environment. The alternative is to use multiple agent instances, which can limit the potential scale of the IBM Tivoli Monitoring or IBM Cloud Application Performance Management environment.

Adding or removing a subnode requires reconfiguring the agent. To reconfigure the agent, you need to stop and restart it, involving all subnodes. You can define the agent as a multi-instance agent; in this case, you can start and stop a single instance, and leave the other instances running.

Along with data sets in subnodes, an agent can define agent-level data sets that are located outside of a subnode.

In the Tivoli Enterprise Portal Navigator tree, a subnode type is displayed under the agent name, and subnode instances are displayed under a subnode type. Subnodes are identified by a Managed System Name (MSN) just like agents, for example `94:Hill.cmn`.

For example, in the Navigator tree in Figure 50 on page 1315, **Watching Over Our Friends** is an agent with three resources (**Boarders**, **Common Areas**, and **Kennel Runs**) and two subnode types (**Common Area** and **Kennel Run**). Two of these resources have subnode types that are defined for them (**Common Area** and **Kennel Run**). A subnode is not required for the third resource (**Boarder**), which is represented by a single row in a table at the base agent level. The Common Area subnode type has three subnode instances: `94:Hill:cmn`, `94:Meadow:cmn`, and `94:Tree:cmn` representing three common areas in the kennel. The Kennel Run subnode type has four subnode instances: `94:system1:run`, `94:system2:run`, `94:system4:run`, and `94:system5:run` representing four kennel runs.

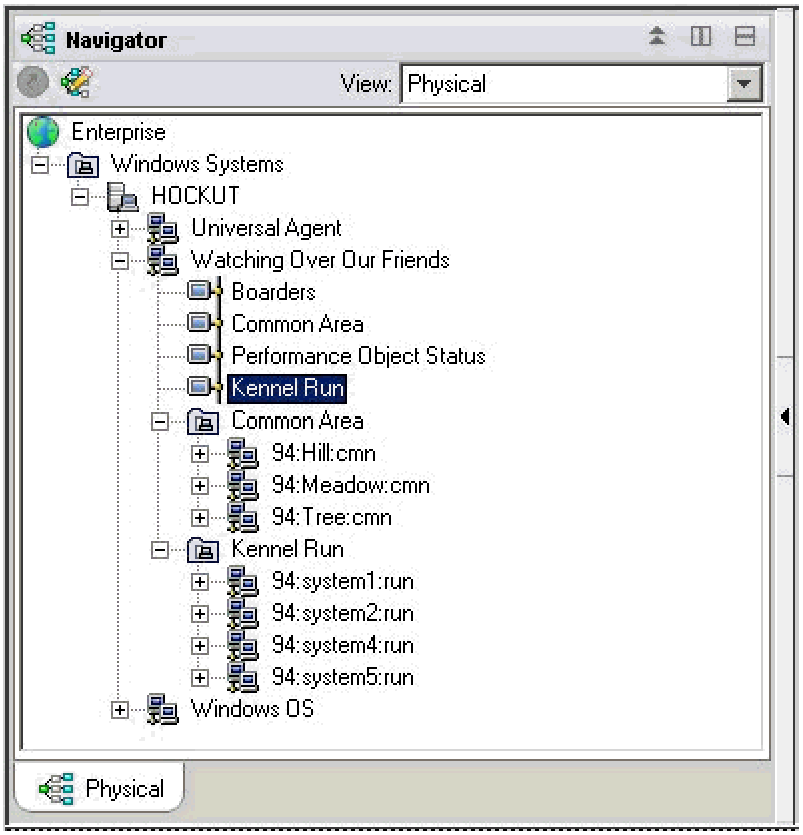


Figure 50. Subnodes in the Navigator tree

There are two ways that a single agent can use subnodes:

- The agent can have different subnodes of the same type.
- The agent can have subnodes of different types.

Subnodes for the same data from different sources

You can use subnodes of the same type to represent multiple instances of a monitored resource type. Each subnode of the same type includes the same attribute groups and the correct values for the specific monitored resource instance. The number of subnodes varies based on agent configuration. The example in Figure 51 on page 1316 shows the monitoring of different systems.

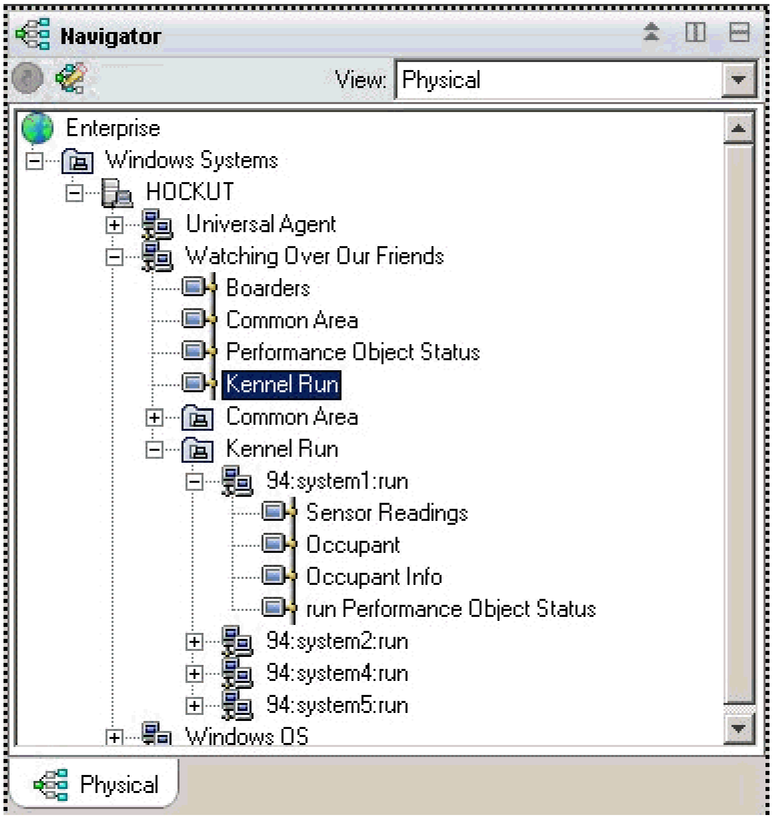


Figure 51. Subnodes monitoring different systems

Subnodes for multiple types of data

When one agent monitors multiple types of monitored resources, you can create a subnode type for each of the resource types. Each subnode includes the information that is defined in that subnode type. The following example shows two subnode types. Each type is monitoring a different type of resource, with different types of data available for each resource:

- Common Area
- Kennel Run

The agent in Figure 52 on page 1317 runs one copy of each subnode type. A particular agent might create any subset of the defined agents. Subnodes can be used to mimic Tivoli Monitoring V5 profiles.

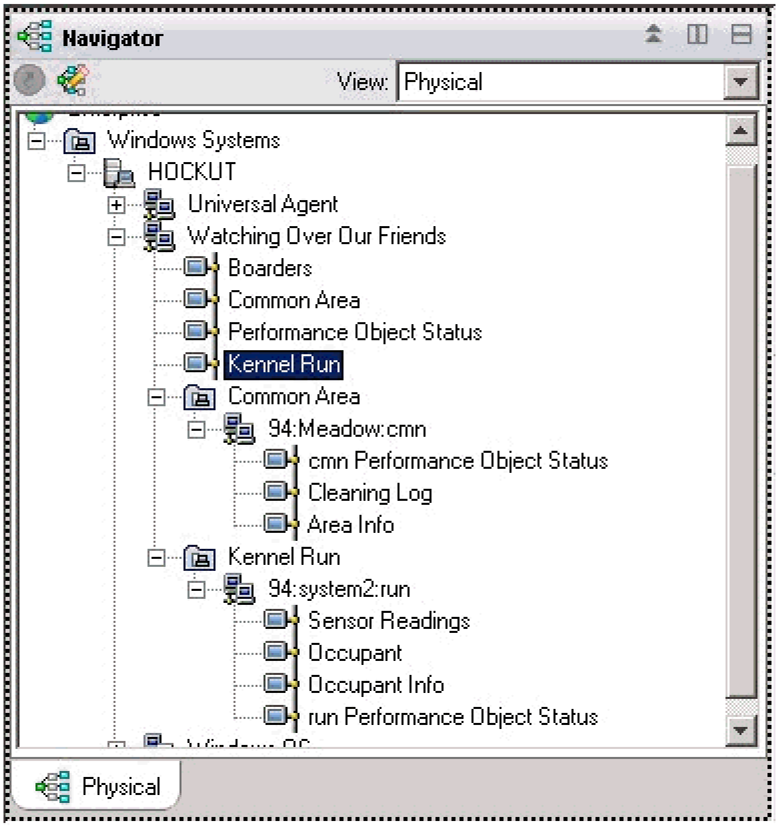


Figure 52. Subnode types in Navigator tree

Both ways of using subnodes can be used in the same agent, where each type can have more than one subnode instance.

Figure 52 on page 1317shows two types of subnodes that monitor two types of resources: Common Areas and Kennel Runs. In addition, there are several subnodes that are defined for each type. There are three subnodes of type Common Area; these subnodes have the following IDs: Meadow, Hill, and Tree. There are also four subnodes of type Kennel (each collecting data from a different system that is dedicated to a Kennel Run); these subnodes have the following IDs: system1, system2, system4, and system5.

Note: The first 24 characters of subnode IDs must be unique for all instances of the subnode type in the IBM Tivoli Monitoring installation.

Data Providers in subnodes

A subnode can contain any mixture of data from the different data provider types. Most current Agent Builder data providers can be used in a subnode, including the following data providers:

- WMI
- Perfmon
- Windows Event Log
- SNMP
- SNMP Events
- JMX
- ICMP ping
- Script
- Log

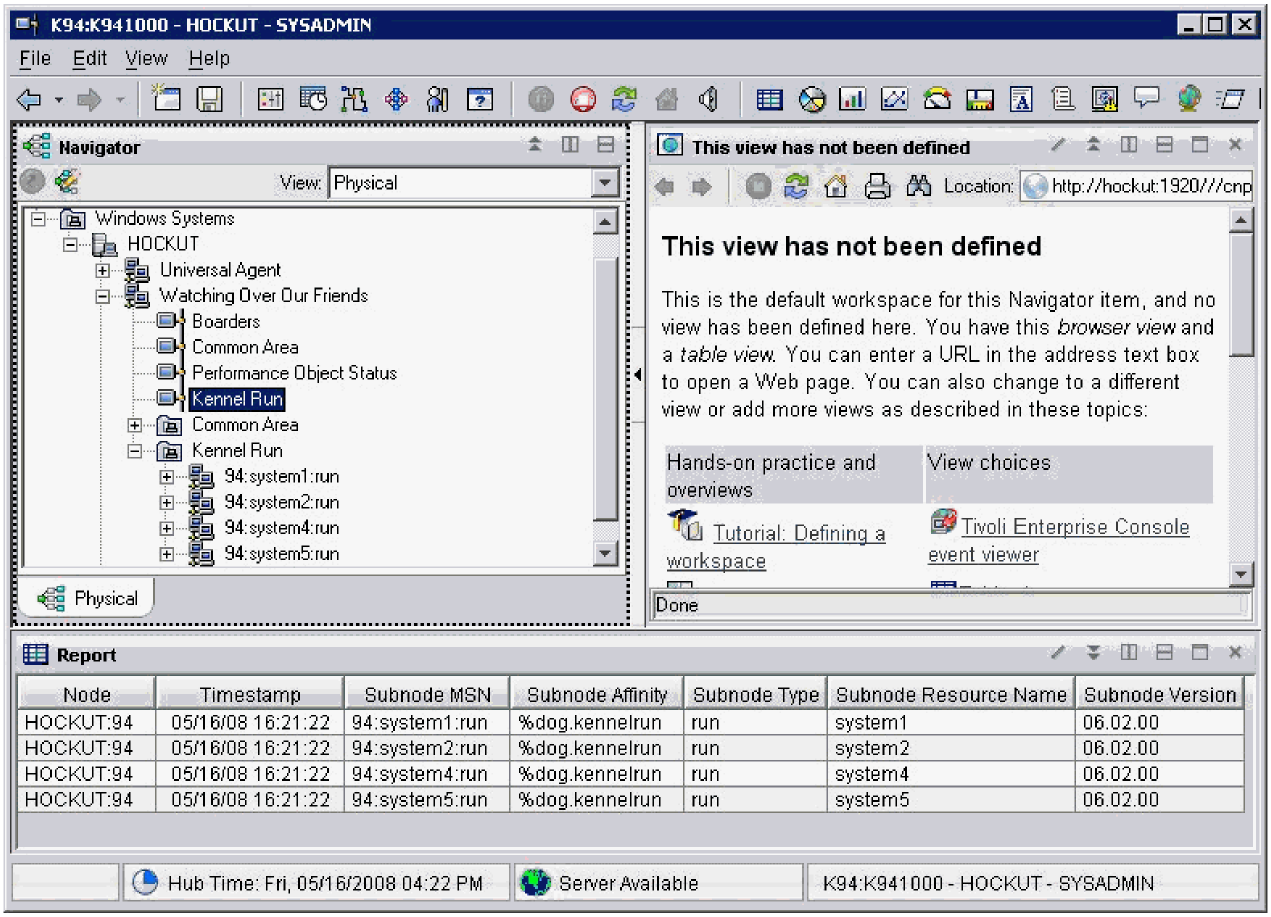
- CIM
- JDBC
- HTTP
- SOAP
- Socket
- Java API

A subnode can also contain a joined attribute group that combines data from two other attribute groups from the same subnode or from agent-level attribute groups.

Status of subnodes

There are two ways to determine status for a subnode agent. The first way is to look at the data that is displayed in the Performance Object Status attribute group. This attribute group displays the status for each of the other attribute groups at the same level in the agent. The Performance Object Status attribute group at the agent level displays the collection status for the other attribute groups at the agent level. The Performance Object Status attribute group in each subnode displays the collection status for the attribute groups in that subnode.

Agent Builder also creates one attribute group for each subnode type, which displays one row for each configured subnode of that type. In the example in (Figure 53 on page 1318), four subnodes are running to collect data.



This view has not been defined

This is the default workspace for this Navigator item, and no view has been defined here. You have this *browser view* and a *table view*. You can enter a URL in the address text box to open a Web page. You can also change to a different view or add more views as described in these topics:

Hands-on practice and overviews	View choices
Tutorial: Defining a workspace	Tivoli Enterprise Console event viewer

Node	Timestamp	Subnode MSN	Subnode Affinity	Subnode Type	Subnode Resource Name	Subnode Version
HOCKUT:94	05/16/08 16:21:22	94:system1:run	%dog.kennelrun	run	system1	06.02.00
HOCKUT:94	05/16/08 16:21:22	94:system2:run	%dog.kennelrun	run	system2	06.02.00
HOCKUT:94	05/16/08 16:21:22	94:system4:run	%dog.kennelrun	run	system4	06.02.00
HOCKUT:94	05/16/08 16:21:22	94:system5:run	%dog.kennelrun	run	system5	06.02.00

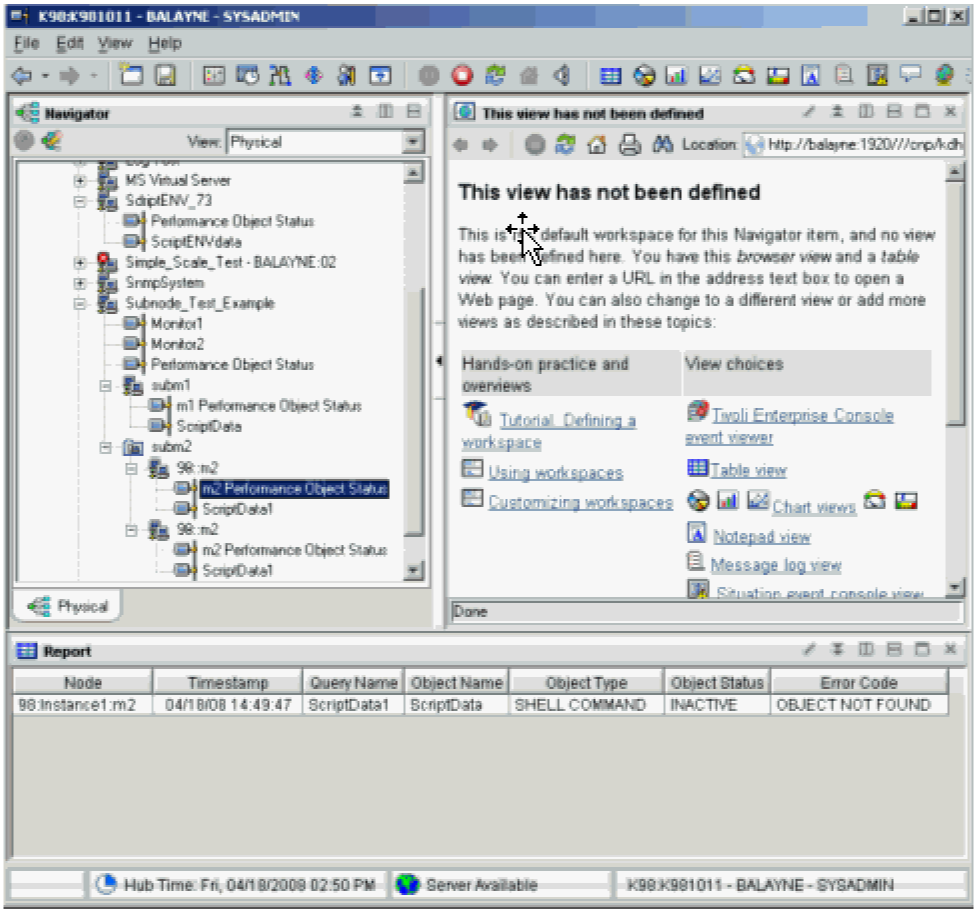
Hub Time: Fri, 05/16/2008 04:22 PM | Server Available | K94:K941000 - HOCKUT - SYSADMIN

Figure 53. Monitoring multiple subnode instances of the same subnode type

In the IBM Tivoli Monitoring environment, the **Performance Object Status** subnode contains data visible in the Navigator tree and can have situations that monitor the status of the other data collections.

In the IBM Cloud Application Performance Management environment, you can create thresholds to monitor **Performance Object Status** data.

The example in Figure 54 on page 1319 shows a case where the data collection failed (the script shell command was not found). Typically, any value other than NO_ERROR indicates that there is a problem. For each of the data collectors that are defined in the subnode, there is one row in the table.


K98:K981011 - BALAYNE - SYSADMIN
File
Edit
View
Help
Navigator
View: Physical
MS Virtual Server
ScriptENV_73
Performance Object Status
ScriptENVdata
Simple_Scale_Test - BALAYNE:02
SnmpSystem
Subnode_Test_Example
Monitor1
Monitor2
Performance Object Status
subm1
m1 Performance Object Status
ScriptData
subm2
98::m2
m2 Performance Object Status
ScriptData1
98::m2
m2 Performance Object Status
ScriptData1
Physical
This view has not been defined
Location: http://balayne:1920///cnp/kdh
Hands-on practice and overviews
Tutorial: Defining a workspace
Using workspaces
Customizing workspaces
View choices
Tivoli Enterprise Console event viewer
Table view
Chart views
Notepad view
Message log view
Situation event console view
Done
Report
Hub Time: Fri, 04/18/2008 02:50 PM
Server Available
K98:K981011 - BALAYNE - SYSADMIN

This view has not been defined

This is the default workspace for this Navigator item, and no view has been defined here. You have this *browser* view and a *table* view. You can enter a URL in the address text box to open a Web page. You can also change to a different view or add more views as described in these topics:

Node	Timestamp	Query Name	Object Name	Object Type	Object Status	Error Code
98:Instance1:m2	04/18/08 14:49:47	ScriptData1	ScriptData	SHELL COMMAND	INACTIVE	OBJECT NOT FOUND

Figure 54. Example: data collection in a subnode

Subnodes in IBM Cloud Application Performance Management

In IBM Cloud Application Performance Management, you can define either the agent instance or a subnode instance or both as monitored resources, and each resource corresponds to a summary dashboard.

Subnode dashboards can not display agent-level data. To display agent-level data in this environment, define a summary dashboard for the agent.

Depending on the settings you select, agent and subnode resources can appear at the same level, with no hierarchical distinction, or subnode resources can be listed as children to agent resources.

For instuctions about configuring agent and subnode resources, see "Preparing the agent for Cloud APM" on page 1342.

Creating subnodes

You can create a subnode when creating or editing an agent.

Procedure

1. Take one of the following steps:

- When creating a new agent using the **Agent** wizard, on the **Agent Initial Data Source** page, click **Data source groupings** in the **Monitoring Data Categories** area.
- With an existing agent, take the following steps in the Agent Editor:

   a. Click the **Data Sources** tab to open the **Data Source Definition** page.

   b. Select the agent and click **Add to selected**.

   c. On the **Data Source Location** page, in the **Monitoring Data Categories** area, click **Data source groupings**.

2. In the **Data Sources** area, click **A Subnode Definition**
3. Click **Next**.
4. Complete the **Subnode Information** page as follows to define the new subnode:

   a) In the **Name** field, type the name of the subnode you are creating.

   b) In the **Type** field, enter 1-3 characters (by using numbers, letters, or both) to identify the type of the subnode you are creating.

   c) In the **Description** field, type a description for the subnode you are creating.

   d) Click the **Show nodes attribute group for this type of subnode** check box to hide or display the availability attribute group. For more details about this attribute group, see "Availability node" on page 1383.

   e) Click **Next**.

5. Complete the **Initial Subnode Data Source** page to select a data source as the first item in the new subnode. Click a category in the **Monitoring Data Categories** list and a data source in the **Data Sources** list. Then, click **Next**.

   Tip: You can create the data source as usual. Alternatively, you can move one or more data sources that you already created into the navigator group. To move data sources, click **Existing data sources** and, in the **Currently Defined Data Sources** page, select the data sources.

   Important: You can not include process, Windows service, or command return code data sources in a subnode. As a workaround, you can write a script that determines the necessary process or service information and use a script output data source.

6. If your agent contains custom configuration properties or if the selected data source requires configuration, use the **Subnode Configuration Overrides** page to choose the configuration properties.

   In the **Subnode Configuration Overrides** page, choose the configuration properties that you want for the subnode at the agent level. Then, choose the configuration properties that you want to vary for each subnode.

   Use **Move**, **Copy**, and **Remove** to specify the configuration properties as described in "Configuring a subnode" on page 1321.

7. Click **Next**.

   The **Data Source Definition** page is displayed.

Subnode configuration

When a subnode type is defined, a single configuration section is defined specifically for that subnode.

There are several ways that a subnode configuration section differs from other configuration sections:

- The set of properties in a subnode section can be duplicated, so there are multiple sets of properties. Each set of properties forms its own section. The layout of all sections is identical, but different values can be entered in each section.

  In contrast, the properties in other sections (which are referred to as agent-level sections) are shown only one time during runtime configuration. They do not form subsections and cannot be duplicated or removed.

  See "Subnode configuration example" on page 1324 for GUI and command-line examples of configuring subnodes.

- For each copy of a subnode section that is created at runtime configuration, the agent creates a separate subnode instance. All of those subnode instances are of the same type.
- The property names in subnode sections can be duplicates of property names in agent-level sections. When duplicate names occur, the subnode property value overrides the agent-level property value.
- In IBM Tivoli Monitoring V6.2.1 and later, a subnode section can have default property values that apply to all instances of subnodes of that type. This feature makes it possible to have a three-level lookup of a single property value as follows:
  1. The agent obtains the property value from the subnode instance subsection.
  2. If no value is configured at the subnode instance level, the property value is obtained from the subnode default level.
  3. If no value is configured at either of those two levels, then the property value is obtained from an agent-level section.

See "Subnode configuration example" on page 1324 for GUI and command-line examples of configuring subnodes.

Configuring a subnode

Use the **Subnode Configuration Overrides** page to configure a subnode data source.

Before you begin

Use the steps in "Creating subnodes" on page 1319 to create a subnode.

About this task

When you add a data source to a subnode, the **Subnode Configuration Overrides** page is presented if the data source requires configuration. It shows custom configuration properties and any other configuration properties that are applicable to the subnode type.

Procedure

- In the **Subnode Configuration Overrides** window, choose the configuration properties that you want for the subnode at the agent level. Also, choose the configuration properties that you want to vary for each subnode.
- Use **Copy >>** to copy configuration properties so that they are both at the agent level and the subnode level.

  The agent looks for a value first at the subnode level, and if it does not find a value, it looks at the agent level. If a property at both levels is a required property, it is required only at the agent level, it is optional at the subnode level.
- Use **Move >>** to move properties from the agent level to the subnode level. **Move>>** is not available for properties that are required by an agent-level data source or by a subnode of a different type.
- Use **Remove** to remove one of the two lists. Properties can be removed only if they are listed at both the agent-level and the subnode level. This function cannot be used to remove a property completely.
- Use **<< Copy** to copy a property from the subnode level to the agent-level.
- Use **<< Move** to move a property from the subnode to the agent-level.

What to do next

You can change the configuration for an existing subnode by using the Agent Editor.

Subnode configuration overrides

Use Subnode Configuration Overrides to override agent configuration properties with subnode-specific properties.

The procedure in "Configuring a subnode" on page 1321 describes how to manage subnode configuration for automatically generated properties. Managing custom configuration properties is similar. Any custom configuration properties that are defined are displayed in the **Subnode Configuration Overrides** window.

When you copy or move a custom property from the subnode level to the agent-level, you are prompted for the section to place the property in. You can select an existing custom section, or enter the name of a new custom section.

Selecting subnode configuration properties

Without subnodes, all instances of a data source type share the configuration parameters. For example, all SNMP attribute groups connect to the same host by using the same community name. With subnodes, each instance of a subnode can connect to a different host if the SNMP_HOST property is placed at the subnode level.

Selecting properties to be overridden at the subnode level is an important consideration when you are developing an agent. If too many properties are selected, the subnode configuration section becomes cluttered and difficult to manage. If too few properties are selected, then the agent functions might be limited when someone wants to vary a property from one subnode to the next.

The following properties cannot be copied to the subnode level. (All attribute groups in all subnodes and in the base agent must use the same SNMP version and JMX connection type):

- SNMP version
- JMX MBean server connection type
- Java home
- Java trace level
- JVM arguments
- Class path for external JAR files
- Socket data source port number
- JMX or JDBC class path settings

Advanced subnode configuration

Use advanced subnode configuration to override an agent configuration property in a subnode.

About this task

There is an option in IBM Tivoli Monitoring V6.2.1 and later agents that you can enable to override properties from any agent-level configuration section in a subnode instance. On the **Subnode Configuration Overrides** page, there is a check box labeled **Allow any configuration property to be overridden in any subnode**. For more information, see ("Subnode configuration overrides" on page 1322). For this option to be enabled, you must select **6.2.1** as the **Minimum ITM version** when you name your agent ("Naming and configuring the agent" on page 1149). If you choose this option, each subnode instance can override any property from any agent-level configuration section. But this property can be overridden only from the GUI and not from the **itmcmd** command line.

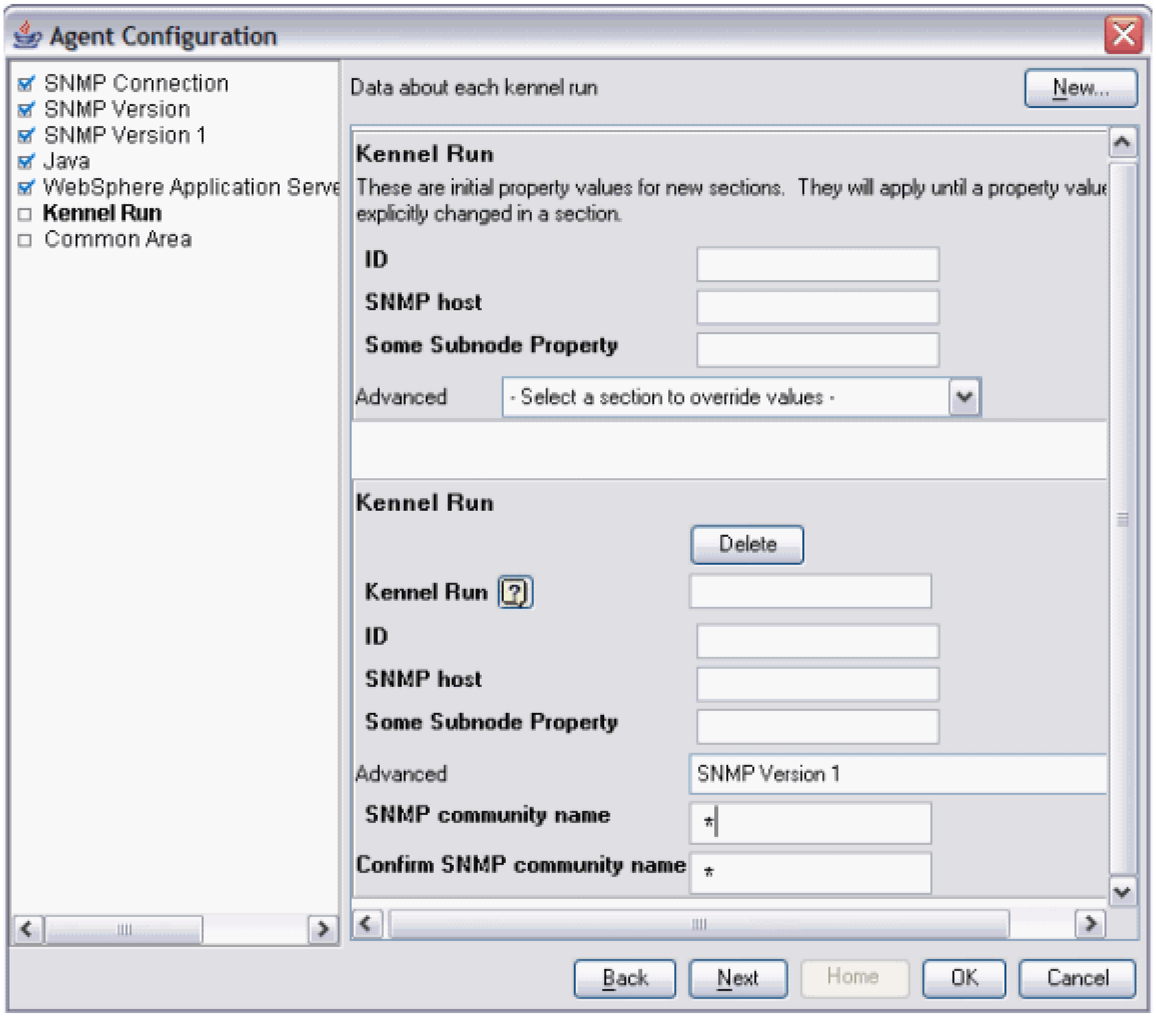
Procedure

The **Allow any configuration property to be overridden in any subnode** option causes an **Advanced** field that contains a list to be displayed on each subnode configuration panel. The initial selection in the **Advanced** field provides the brief directions: **Select a section to override values**.

- When you click the list, you see a list of all the non-subnode sections that contain configuration properties.

- Select a section.

  The properties from that section are temporarily added to the subnode panel. The value of any property that you change is added to the set of properties that are defined for the subnode. A data source in the subnode looks for property values in the subnode before it looks in the agent-level sections. .



Agent Configuration

☑ SNMP Connection
☑ SNMP Version
☑ SNMP Version 1
☑ Java
☑ WebSphere Application Serve
□ **Kennel Run**
□ Common Area

Data about each kennel run

Kennel Run

These are initial property values for new sections. They will apply until a property value explicitly changed in a section.

ID

SNMP host

Some Subnode Property

Advanced: - Select a section to override values -

Kennel Run

Kennel Run

ID

SNMP host

Some Subnode Property

Advanced: SNMP Version 1

SNMP community name: *

Confirm SNMP community name: *

Figure 55. SNMP Version 1 Properties expanded

The following further information applies to overriding properties from agent-level sections:

- Properties that were copied to the subnode section are not shown when the agent-level section is selected in the **Advanced** list. For example, in Figure 55 on page 1323, **SNMP host** is not displayed after the **Advanced** list because it was copied to the subnode properties and is already displayed.
- Sections that contain no properties to override do not have a selection in the **Advanced** list.
- Overridden values that you enter for one section are retained even if you select a different section to display different properties.
- Select **Allow any configuration property to be overridden in any subnode** to enable this feature in your agent.

Configuring a subnode from the command line

In the IBM Tivoli Monitoring environment, you can also configure a subnode by using the command line.

Before you begin

For more information about subnode configuration, see "Subnode configuration" on page 1320

About this task

Procedure

- To configure a subnode instance from the command line, use the following command:

  ```
  tacmd configureSystem -m HOSTNAME:00 -p
  section_name:subnode_instance_id.property_name=value
  ```

  Where:

  section_name
  Same as the subnode type

  subnode_instance_id
  ID for the subnode that is defined during configuration.

  property_name
  Name of the configuration property

  value
  Value for the property

Subnode configuration example
How to configure a sample agent with one defined subnode.

Example:

This example shows how to configure a sample agent that has one subnode named Example Subnode of type `exs` and the following three configuration properties:

- `Agent Cfg` (actual property name is `K00_AGENT_CFG`) is defined only at the agent level.
- `Subnode Cfg` (actual property name is `K00_SUBNODE_CFG`) is defined only in the example subnode.
- `Overridable Cfg` (actual property name is `K00_OVERRIDABLE_CFG`) is defined at the agent level and was copied to the example subnode.

(Figure 56 on page 1325) shows these configuration properties on the **Runtime Configuration Information** page of the Agent Editor.

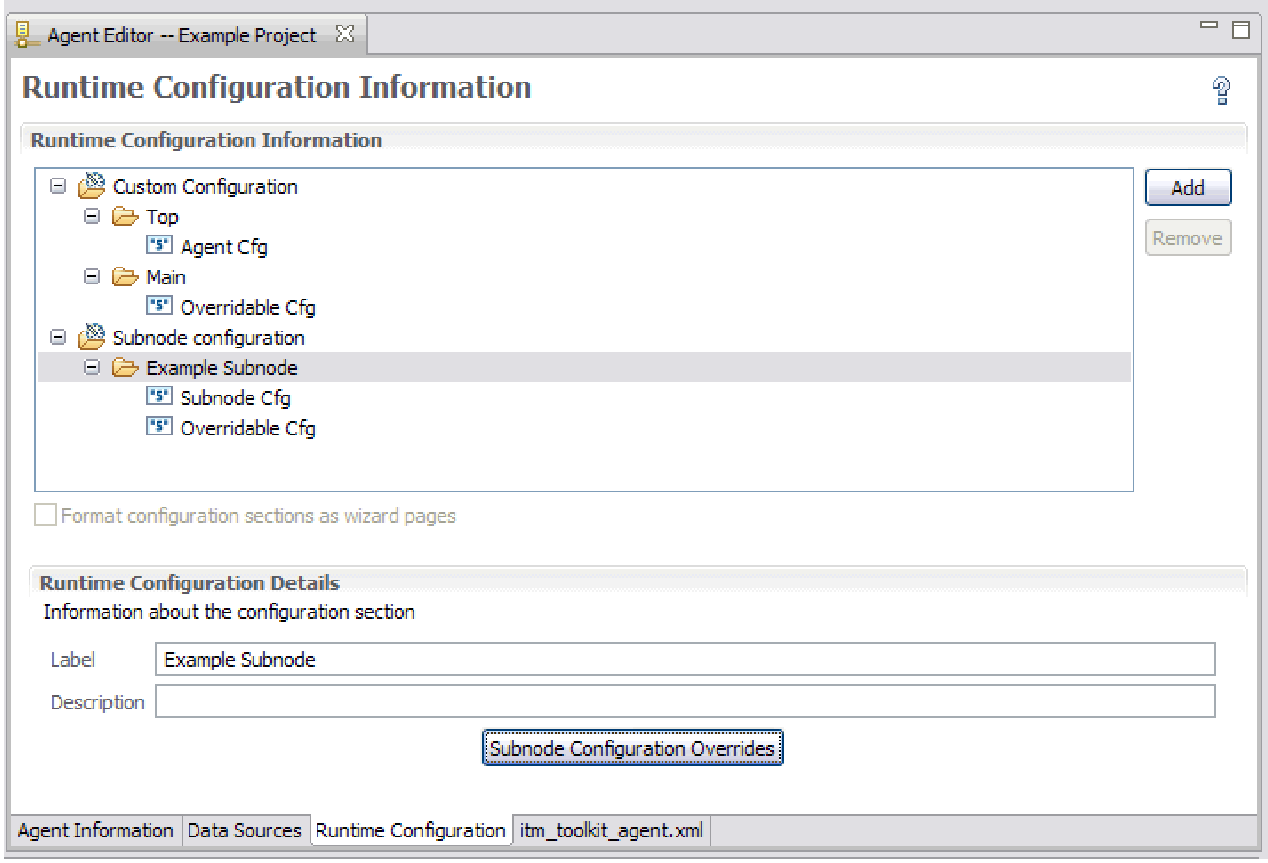
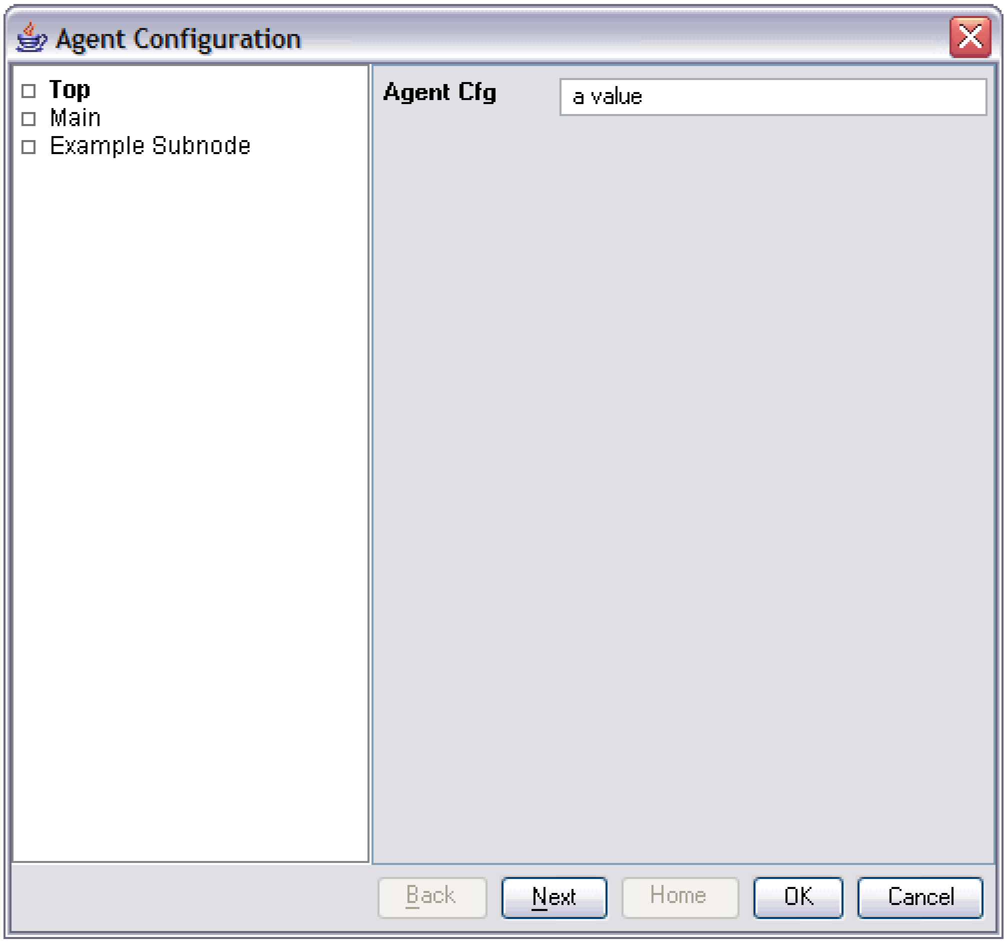

Agent Editor -- Example Project
Runtime Configuration Information
Runtime Configuration Information
Custom Configuration
Top
Agent Cfg
Main
Overridable Cfg
Subnode configuration
Example Subnode
Subnode Cfg
Overridable Cfg
Add
Remove
Format configuration sections as wizard pages
Runtime Configuration Details
Information about the configuration section
Label
Example Subnode
Description
Subnode Configuration Overrides
Agent Information
Data Sources
Runtime Configuration
itm_toolkit_agent.xml

Figure 56. Configuration property definitions in the Agent Builder

When this example agent is configured, the first page that is displayed is the **Top** section, which contains the **Agent Cfg** property as shown in (Figure 57 on page 1326). Because this property is an agent-level property, it is shown one time during agent configuration. Any instance of the Example Subnode can see this property value, but all instances see the same value.



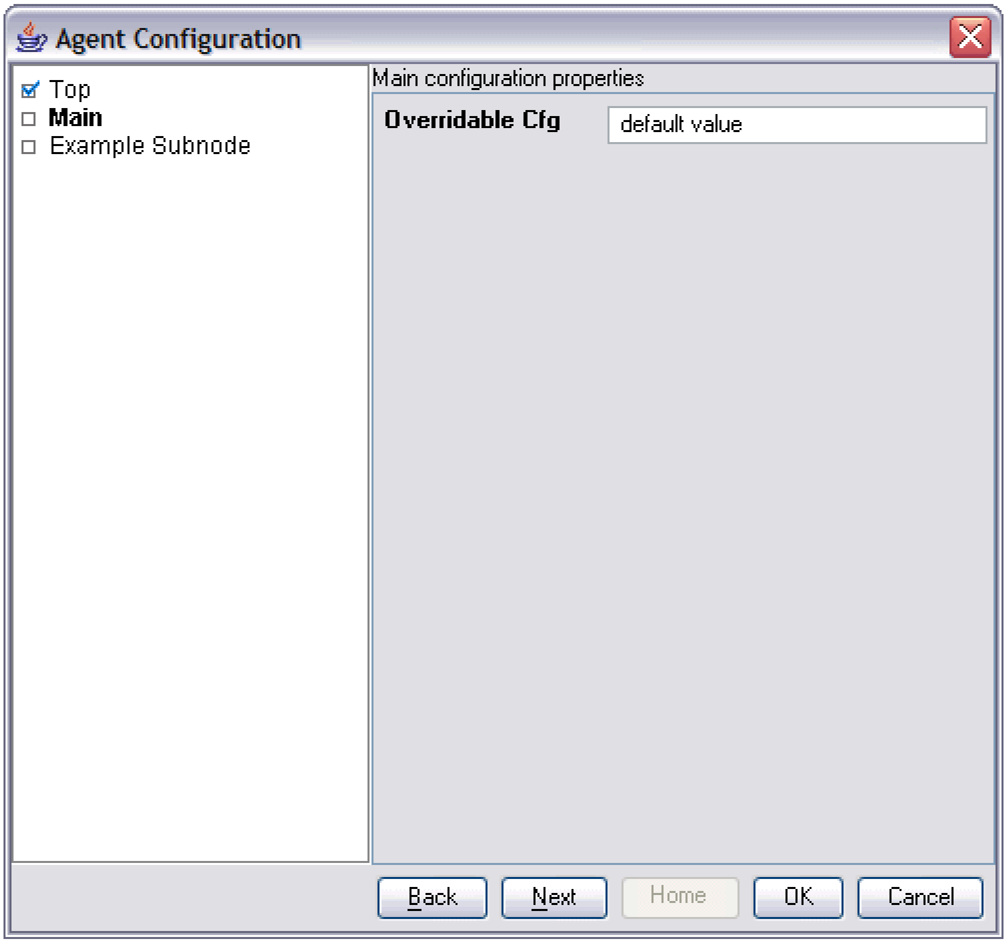
Agent Cfg: a value

*Figure 57. **Top** section with agent-level configuration for the **Agent Cfg** property*

If you are configuring from the Tivoli Enterprise Monitoring Server command line, the **Agent Cfg** property can be set by using the following command:

```
tacmd configureSystem -m HOSTNAME:00 -p "TOP.K00_AGENT_CFG=a value"
```

The next section that is displayed is the **Main** section as shown in Figure 58 on page 1327. It is also an agent-level section and contains the agent-level **Overridable Cfg** property. This property differs from the **Agent Cfg** property because this property was copied to the Example Subnode in the Agent Builder. This means that a default value for the property can be entered on the **Main** page. However, any Example Subnode instance can override the value that is entered here with a different value.



Main configuration properties

Overridable Cfg: default value

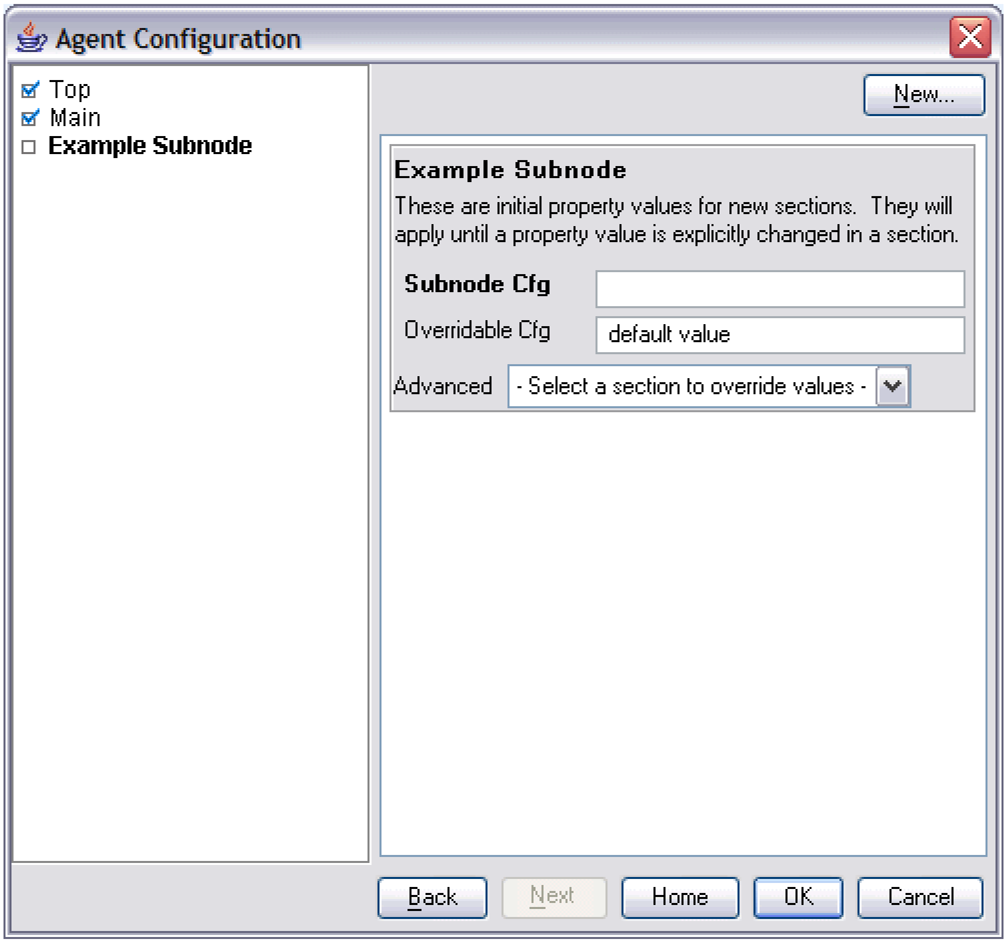
Figure 58. ***Main*** *section with the agent-wide default value for the* ***Overridable Cfg*** *property*

If you are configuring from the Tivoli Enterprise Monitoring Server command line, this property can be set by using the following command:

```
tacmd configureSystem -m HOSTNAME:00 -p "MAIN.K00_OVERRIDABLE_CFG=default value"
```

You can place both of these properties in the same agent-level section. You can decide how many custom agent-level sections to create and how to distribute custom properties among them.

The next section that is displayed is the **Example Subnode** section as shown in Figure 59 on page 1328. Because this agent is being configured for the first time, there are no defined subnode instances and no subnode instance subsections are shown. The initial property values subsection is shown, although it is optional and some subnode types might not show it. Because the initial property values subsection is shown, default values can be entered for any of the configuration properties. The **Overridable Cfg** property already has a default value that was obtained from the agent-level property of the same name.



Example Subnode

These are initial property values for new sections. They will apply until a property value is explicitly changed in a section.

Subnode Cfg:

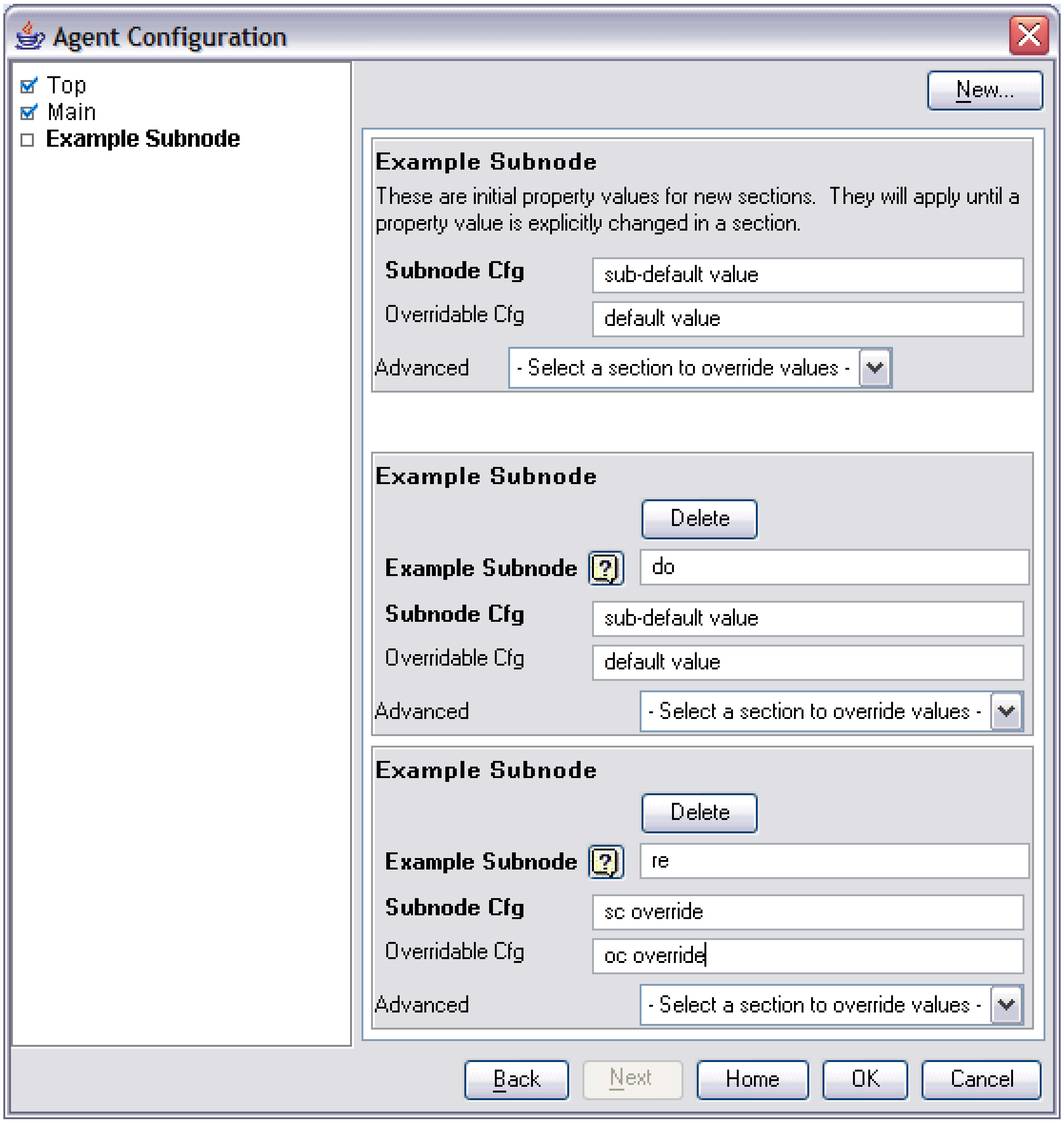
Overridable Cfg: default value

Advanced: - Select a section to override values -

*Figure 59. **Example Subnode** section page with no subnode*

Subnode instances are defined by doing the following actions on the empty **Example Subnode** section page (Figure 60 on page 1329):

1. In the initial **Example Subnode** section, in the **Subnode Cfg** field, type the following default string for the property: `sub-default value`.
2. Click **New**. An **Example Subnode** subsection is displayed after the initial properties subsection.
3. In the **Example Subnode** field, type the following subnode instance ID: do.
4. Click **New**. A second **Example Subnode** subsection is shown after the first.
5. In the second **Example Subnode** field, type the following subnode instance ID: `re`.
6. In the **Subnode Cfg** field, type the following value for the **Subnode Cfg** property: `sc override`.
7. In the **Overridable Cfg** field, type the following value for the **Overridable Cfg** property: `oc override`.



Example Subnode

These are initial property values for new sections. They will apply until a property value is explicitly changed in a section.

Subnode Cfg: sub-default value
Overridable Cfg: default value
Advanced: - Select a section to override values -

Example Subnode

Delete

Example Subnode: do
Subnode Cfg: sub-default value
Overridable Cfg: default value
Advanced: - Select a section to override values -

Example Subnode

Delete

Example Subnode: re
Subnode Cfg: sc override
Overridable Cfg: oc override
Advanced: - Select a section to override values -

Figure 60. ***Example Subnode*** *section page with two subnode instances defined*

The two new subsections cause the agent to create two subnode instances when it is started. Because the properties of the **do** subnode subsection were not changed, the default property values are used by that subnode instance. Since different values were entered for the properties in the **re** subsection, the **re** subnode instance uses those values that were typed.

You can set a default value from the Tivoli Enterprise Monitoring Server command line with the following command:

```
tacmd configureSystem -m HOSTNAME:00 -p "exs.K00_SUBNODE_CFG=sub-default value"
```

The format for setting subnode default values is exactly like the format for setting agent-level properties, except that the section name identifies a subnode section.

You can create the subnode instances from the Tivoli Enterprise Monitoring Server command line with the following command:

```
tacmd configureSystem -m HOSTNAME:00 -p "exs:do.K00_OVERRIDABLE_CFG=default value" \
 "exs:re.K00_SUBNODE_CFG=sc override" "exs:re.K00_OVERRIDABLE_CFG=oc override"
```

The subnode instance ID is inserted between the section name and property name. When you use the command line to create a subnode instance, at least one property must be specified, even if all the properties use default values. Otherwise, default values are not required to be specified on the command line when you define subnode instances.

All of the agent configuration properties can be set in a single command. The following command is equivalent to all of the preceding individual commands:

```
tacmd configureSystem -m HOSTNAME:00 -p "TOP.K00_AGENT_CFG=a value" \
 "MAIN.K00_OVERRIDABLE_CFG=default value" \
 "exs.K00_SUBNODE_CFG=sub-default value" \
 "exs:do.K00_OVERRIDABLE_CFG=default value" \
 "exs:re.K00_SUBNODE_CFG=sc override" "exs:re.K00_OVERRIDABLE_CFG=oc override"
```

Subnodes and Windows data sources

Choose to include Windows Remote Connection properties in the agent or not.

About this task

If an agent has Windows data sources at the agent level and not in subnodes, including Windows Remote Connection configuration properties in the agent are optional. Windows data sources are Windows Event Log, Windows Management Instrumentation, Windows Performance Monitor. If configuration properties are not included, these data sources monitor the local Windows system by default and need no configuration. By default, no Windows data sources are included in any subnode.

To choose whether to include Windows Remote Connection properties in the agent, do the following steps:

Procedure

1. On the **Windows Management Instrumentation (WMI) Information** page, click **Global Options** when data source properties are displayed. Select **Global Options** either while you are creating the data source or from the Agent Editor **Data Sources** page.
2. In the **Global Windows Data Source Options** window, select **Include Windows Remote Connection configuration** if you want to include these properties in the agent.

Subnodes and Script data sources

Subnode instance configuration properties are accessed in subnode scripts just as they are in agent-level scripts.

Scripts have access to all agent-level configuration properties and all subnode instance configuration properties. If an agent-level property is overridden at the subnode level, the script has access only to the subnode level property value.

Customizing agent configuration

Customize the configuration of process, log file, and script data sources.

Before you begin

If you are adding SNMP, JMX, CIM, JDBC, HTTP, and SOAP data sources to your agent, configure these data sources as described in the following sections:

- “Monitoring data from a Simple Network Management Protocol (SNMP) server” on page 1204
- “Monitoring Java Management Extensions (JMX) MBeans” on page 1214
- “Monitoring data from a Common Information Model (CIM)” on page 1233
- “Monitoring data from Java Database Connectivity (JDBC)” on page 1258
- “Monitoring HTTP availability and response time” on page 1268
- “Monitoring data from a SOAP or other HTTP data source” on page 1275

About this task

Use this task to customize the configuration of process, log file, and script data sources so an agent can access the application that it is monitoring.

All agents must be configured before they can be started. All agents must have basic configuration information such as the method of connecting to the Tivoli Enterprise Monitoring Server. Many times, an agent must have more configuration information so it has access to information specific to the system on which it is running. For example, if you must know the installation location of a software product, add configuration properties to prompt for this information. Another example of information you might prompt for is the user ID and password to access an interface,

Custom configuration is defined by the agent developer. It is not required for all agents, but can be used in the following areas of data collection:

- Matching an argument in a Process Monitor
- Matching the command line in a Process Monitor
- Forming a log file path or name
- Defining an environment variable in a script

Note: Certain data sources such as JMX and SNMP add this configuration automatically.

Note: When data source specific configuration is added automatically by the Agent Builder, this configuration is added in English only.

If during data source definition your agent requires system-specific information for an area of data collection, **Insert Property** or **Insert Configuration Property** is shown.

For example, when you create an attribute group that monitors a log file, **Insert Configuration Property** is shown.

Procedure

1. Click **Insert Configuration Property** to display the **Configuration Properties** window,
2. In the **Configuration Properties** window, click a property, and click **Add**.

   Note: Initially there are no configuration properties that are defined for the agent.
3. In the **Runtime Configuration Property** window, complete the following fields:

   a) In the **Section** area, complete the following fields:

      Label
      Text that describes the properties

      Description
      (optional) Description of the properties

   b) In the **Property** area, complete the following fields:

      Label
      Text that is displayed in the agent configuration panel that identifies the information you must enter.

      Environment variable
      The environment variable is displayed in the **Environment variable** field and is updated as you type in the label field. The Agent Builder automatically constructs the name of the environment variable from the product code and the label. If you want to change the environment variable independently from the label, you can clear **Match Label**.

      Description
      (optional) Description of the property that is being defined.

      Type
      Type of information that is collected, one of the following options:

String
For any alphabetic information that must be collected (for example, installation locations, user names, and host names).

Password
For any information that must be encrypted when stored. In addition to providing encryption of the data, the data that is typed into the text box is obscured by asterisks. In addition, you are required to type this information twice to validate the data.

Numeric
For any numeric information (for example, port numbers).

Choice
For a list of specific values. This option enables the Choices table. You can define specific values by clicking **Add**. The values that are entered are displayed in the agent configuration panel as a group of selections, you can make only one selection from the group.

Read Only Text
Displays text when you configure the agent, but no information is collected.

Separator
Displays a horizontal separator, but no information is collected.

File Browser
Collects a string, which is a file name. Click **Browse** to browse the file system for the wanted file.

Default value
(Optional) Specify the value that is shown in the configuration panel at run time when the agent is configured for the first time. If you want a default value for UNIX/Linux that is different from a default value for Windows, click **Multiple Values**.

In the **Configuration Property Default Values** window , specify the default values that you want for Windows systems and for UNIX and Linux systems.

Note: Support for multiple default values is a feature that is only supported in IBM Tivoli Monitoring V6.2.1 and higher. If your agent is compatible with IBM Tivoli Monitoring V6.2, a prompt warns you about this requirement and you can cancel or continue with V6.2.1 compatibility enabled.

Required
Check this field if the user must enter a value when the agent is configured. Clear this field if it is optional for the user to enter a value.

c) To add a choice, click **Add**

4. In the **Configuration Property Value** window, complete the **Label** and **Value** fields.

   The Label is displayed as one of the choices. If this choice is taken, the value becomes the property value.
5. Click **OK**.

   The new configuration section and property are displayed in the **Configuration Properties** window under **Custom Configuration**.
6. Optional: To add another property to an existing section, select the section or an existing property in the section and click **Add**. You make the selection in the runtime configuration tree of the **Configuration Properties** window.
7. Complete the fields for the new property. (Complete the same fields as in step "3" on page 1331).
8. Click **OK**. The property that you most recently added is selected.
9. Keep the selection or select the property that you want to insert into the log file name.
10. Click **OK**. The property is inserted into the log file name.

    You can then continue through the wizard to complete defining your log file attribute group.

Note: Even though a configuration property is defined in the context of a log file name, it can be used in other locations. For instance, another location that accepts a configuration property is a script data source. This flexibility means that you can access the value for the configuration element **File Information** with the script variable *$K00_APPLICATION_LOG_FILE* if the product code is K00. You can also use the Windows batch file variable *%K00_APPLICATION_LOG_FILE%*.

Changing configuration properties by using the Agent Editor

Use the Agent Editor to change configuration properties of your agent.

About this task

This task provides information about viewing, adding, and changing configuration properties by using the Agent Editor.

Procedure

1. Click the **Runtime Configuration** tab.
2. Select a configuration section, and click **Add**.

   Add works just like it does in "Customizing agent configuration" on page 1330. There is no **Edit** selection because a configuration section or property is edited when it is selected.
3. Select a configuration property to display the **Runtime Configuration Details** area.
4. In the **Runtime Configuration Details** area, edit the fields to configure the property.

Configuring a Windows remote connection

Information about Configuring a Windows remote connection

About this task

Windows Management Instrumentation (WMI), Windows Performance Monitor (Perfmon), and Windows Event Log data sources can monitor data on the system where the agent is installed. These data sources can also monitor data on remote Windows systems. These three data source types are known as Windows data sources. If these Windows data sources are monitoring data remotely, they all share Windows Remote Connection configuration properties for the agent level where they are defined.

If you define a Windows data source in the base level of your agent, Windows Remote Connection configuration properties are not added to the agent automatically. They are not added to maintain compatibility with earlier versions of agents that might use the Windows data provider before remote monitoring was enabled. The Windows data source in your agent monitors data on the local Windows system where the agent is installed.

If you define a Windows data source in a subnode in your agent, Windows Remote Connection configuration properties are added to the agent automatically. The Windows data source must support Windows Remote Connection if it is in a subnode. You cannot clear the option until all windows data sources are removed from all subnodes in the agent. Each instance of a subnode might be configured to monitor a different remote Windows system. All Windows data sources in the subnode share Windows Remote Connection configuration properties.

To configure a base agent to remotely monitor a single remote Windows system, use the following procedure.

Procedure

1. In the Agent Editor **Data Source Definition** window, click **Global Options**.

   The **Global Windows Data Source Options** window opens.
2. Select **Include Windows Remote Connection configuration**.
3. Click **OK**.

Results

The following connection-specific configuration properties can be accessed from the Agent Editor **Runtime Configuration Information** page by selecting **Configuration for Windows remote access** > **Windows Remote Connection**

Remote Windows host
Host name of remote Windows computer

Remote Windows password
Password for remote Windows

Remote Windows `DOMAIN\user name`
User name for the remote Windows host

What to do next

You can view, add, and change the configuration properties by using the Agent Editor. For instructions, see "Changing configuration properties by using the Agent Editor" on page 1333. If a Windows data source is defined in a subnode, you can also specify Subnode Configuration Overrides. For instructions, see "Subnode configuration" on page 1320.

Creating a user with Windows Management Instrumentation (WMI) permissions

You can add and configure a user on a Windows system with permissions to allow WMI browsing.

About this task

If your agent collects data from a remote system by using Windows Management Instrumentation (WMI), it requires permissions to access WMI data on the remote system. The agent can access WMI data on a remote system when you provide credentials of an account with permissions to access WMI data on the system. The procedure applies to Windows 7, Windows 2008 Server and Windows Vista.

Note: Your agent can also access data on a remote Windows system by using Windows Performance Monitor (Perfmon), and Windows Event Log data sources. However, in the case of Windows Performance Monitor (Perfmon), and Windows Event Log data sources, you must provide Administrator credentials for the remote system.

Procedure

1. Create a user account:
   a. Go to Windows **Start** > **Administrative Tools** > **Computer Management**. The **Computer Management** window opens.
   b. Expand **Local Users and Groups**.
   c. Right-click the **Users** folder and select **New User**.
   d. Complete the user details and click **Create** and **Close**.
2. Configure the group membership for the new user account:
   a. In the **Computer Management** window, select the **Users** folder.
   b. Right-click the new user account and select **Properties**.
   c. Click the **Member Of** tab.
   d. Click **Add**.
   e. Click **Advanced**.
   f. Click **Find Now**.
   g. Select the following groups:
      - Distributed COM Users
      - Performance Log Users

- Remote Desktop Users

  Tip: Press Ctrl and click to select multiple groups.

h. Click **OK** until you return to the **Computer Management** window.

i. Select **File** > **Exit** to exit the **Computer Management** window.

3. Assign Distributed Component Object Model (DCOM) rights:

   a. Go to Windows **Start** > **Administrative Tools** > **Component Services**. The **Component Services** window opens.

   b. Expand **Component Services** > **Computers** > **My Computer**.

   c. Right-click **My Computer** and select **Properties**. The **My Computer Properties** window opens.

   d. Click the **COM security** tab.

   e. In the **Access Permissions** area, click **Edit Limits**

   f. In **Distributed COM Users**, verify that **Local Access** and **Remote Access** are selected.

   g. Click **OK** to save settings.

   h. In the **My Computer Properties** window, **Launch and Activation Permissions** area, click **Edit Limits**

   i. In **Distributed COM Users**, verify that **Local Launch**, **Remote Launch**, **Local Activation**, and **Remote Activation** are selected.

   j. Click **OK** to save settings and click **OK** again to close the **My Computer Properties** window.

   k. Select **File** > **Exit** to exit the **Component Services** window.

4. Configure the WMI namespace security assignments

   a. Go to Windows **Start** > **Run...**.

   b. Enter `wmimgmt.msc` and click **OK**.

   c. Right-click **WMI Control (Local)** and select **Properties**.

   d. Click the **Security** tab.

   e. Click **Security**.

   f. Click **Add**.

   g. Click **Advanced**.

   h. Click **Find Now**.

   i. Select the new user account, and click **OK** until you return to the **Security for Root** window.

   j. Click **Advanced** and select the newly added user account.

   k. Click **Edit**.

   l. From the **Apply to:** menu selection, select **This namespace and subnamespaces**.

   m. In **Execute Methods**, verify that **Enable Account**, **Remote Enable**, and **Read Security** are selected.

   n. Click **OK** until you return to the **wmimgmt** window.

   o. Select **File** > **Exit** to exit the **wmimgmt** window.

What to do next

For more information about collecting WMI data from a remote system, see "Monitoring data from Windows Management Instrumentation (WMI)" on page 1200.

Configuring a Secure Shell (SSH) remote connection

Information about configuring an SSH remote connection

About this task

Script data sources can monitor data on the system where the agent is installed and also on remote systems. If script data sources are monitoring data remotely, they all share SSH remote connection configuration properties for the agent level where they are defined. Earlier versions of an agent might use the script data provider before remote monitoring was enabled. To maintain compatibility with earlier versions of agents, SSH remote connection configuration properties are not automatically added to the agent. The script data source in your agent monitors data on the local system where the agent is installed.

If you define a Script data source in a subnode and you select **Enable data collection using SSH**, you can configure each subnode instance to monitor a different remote system. All script data sources in the subnode share SSH remote connection configuration properties.

If you want the agent to remotely monitor a remote system, use the following procedure.

Procedure

In the Agent Editor **Data Source Definition** window for the script data source, select **Enable data collection using SSH**.

Results

The following connection-specific configuration properties can be accessed from the **Agent Editor**, **Runtime Configuration Information** page by selecting **Configuration for Secure Shell (SSH)** > **SSH Remote Connection**

Network address
The IP address or host name of the remote computer.

SSH Port Number
The IP port number on which the SSH server is running. The default value is 22.

Authentication Type
Type of authentication to use when you are logging on to the remote SSH server. You can choose Password or Public Key.

Disconnect from the remote system after each collection interval
An option to determine whether the script data provider drops the login session to the remote system after it collects data. By default, the value is No.

Remove script from the remote system after each collection interval
An option to delete the script from the remote system after each data collection interval. By default, the value is No.

If the Authentication Type is set to Password, the following configuration properties can be accessed from the **Agent Editor**, **Runtime Configuration Information** page by selecting **Configuration for Secure Shell (SSH)** > **Password**:

Username
User name for the remote system

Password
Password for the remote system

If Authentication Type is set to Public Key, the following configuration properties can be accessed from the **Agent Editor**, **Runtime Configuration Information** page by selecting **Configuration for Secure Shell (SSH)** > **Public Key**:

Username
User name that is associated with the public key file

Public Keyfile
Public key file that is associated with the user

Private Keyfile
Private key file that is associated with the user

Password
Password that is used to unlock the private key file

What to do next

You can view, add, and change the configuration properties by using the Agent Editor. For instructions, see "Changing configuration properties by using the Agent Editor" on page 1333. If the SSH Remote Connection configuration properties are included in a subnode, you can also specify Subnode Configuration Overrides. For instructions, see "Subnode configuration" on page 1320.

Creating workspaces, Take Action commands, and situations

After installing an agent in an IBM Tivoli Monitoring environment, you can create workspaces, queries, Take Action commands, and situations for your monitoring solution.

The situations, workspaces, Take Action commands, and queries that you create can be included in the installation package. To have one installation image for situations, workspaces, and the agent itself, the situation, and workspace files must be in the same project as the agent. The Agent Builder provides a wizard to create the appropriate files in the agent project. For information about importing application support files, see "Importing application support files" on page 1371.

Creating situations, Take Action commands, and queries

Find information to help create situations, Take Action commands, and queries.

To create situations, Take Action commands, and queries, use the Tivoli Enterprise Portal and the embedded Situation editor. For detailed information about how to create situations, see the Tivoli Enterprise Portal User's Guide. You can also use the help documentation that is installed with your Tivoli Enterprise Portal Server. An Agent Builder monitoring agent can recognize and perform special processing for a set of specific Take Action commands. For more information about these special Take Action commands, see "Take Action commands reference" on page 1470.

Situations for system monitor agents are created differently from the Enterprise situations that are created with the Tivoli Enterprise Portal Situation editor or the **`tacmd createSit`** command. For system monitor agents, private situations are created in a local private situation configuration XML file for the agent. For information about creating situations for system monitor agents, see "Private situations" in the "Agent Autonomy" chapter of the *IBM Tivoli Monitoring Administrator's Guide*.

Creating workspaces

Place the Tivoli Enterprise Portal in the Administrator mode to create workspaces that you can export and include in your solution.

About this task

Build the workspaces in the environment from which they are used. When you build workspaces, change the display settings on your computer to build workspaces at the minimum resolution that is normally used in your environment. Building workspaces at a greater resolution can create views that are too cluttered to be used reasonably at lesser resolutions.

To create workspaces that you can export and include in your solution, the Tivoli Enterprise Portal must be placed in the "Administrator" mode. To place the Tivoli Enterprise Portal in "Administrator" mode, use the following steps:

Procedure

1. Go to the *`ITM_INSTALL`*`/CNP` directory and open the `cnp.bat` file.

If you used the default installation, the directory is `C:\IBM\ITM\CNP`. In the `cnp.bat` file, you must update the `set _CMD= %_JAVA_CMD%` line to include option `-Dcnp.candle.mode="$_KCJ_$"`.

If you want to create extensions on Linux or AIX systems, use the following path:

```
/opt/IBM/ITM/li263/cj/bin/cnp.sh
```

Where *li263* is the operating system on which the Tivoli Enterprise Portal is running.

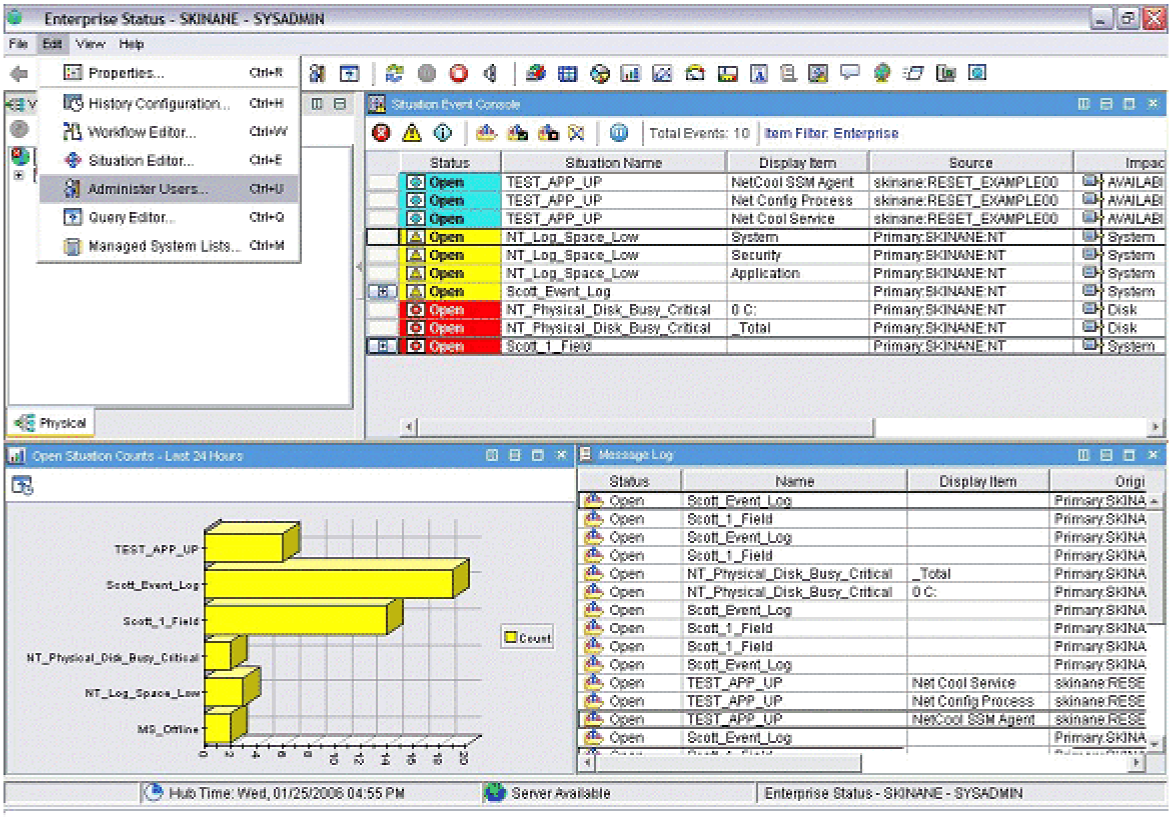
The updated `set _CMD= %_JAVA_CMD%` looks similar to the following example:

```
set _CMD= %_JAVA_CMD% -Dcnp.candle.mode="$_KCJ_$" -Xms64m -Xmx256m -showversion -noverify
 -classpath %CPATH% -Dkjr.trace.mode=LOCAL -Dkjr.trace.file=C:\IBM\ITM\CNP\LOGS\kcjras1.log
-Dkjr.trace.params=ERROR -DORBtcpNoDelay=true -Dibm.stream.nio=true
-Dice.net.maxPersistentConnections=16 -Dice.net.persistentConnectionTimeout=1
-Dcnp.http.url.host=SKINANE -Dvbroker.agent.enableLocator=false  -Dnv_inst_flag=%NV_INST_FLAG
%
-Dnvwc.cwd=%NVWC_WORKING_DIR% -Dnvwc.java=%NVWC_JAVA% candle.fw.pres.CMWApplet
```

Note: The command is shown here on multiple lines for formatting reasons only.

2. Open a new Tivoli Enterprise Portal Client, and log in with the `sysadmin` user ID.
3. Set the "sysadmin" user ID in "Administrator" mode. In the Tivoli Enterprise Portal, select **Edit** > **Administer Users**. Select **sysadmin** and then under the **Permissions** tab, select **Workspace Administration**. Select the **Workspace Administration Mode** check box.

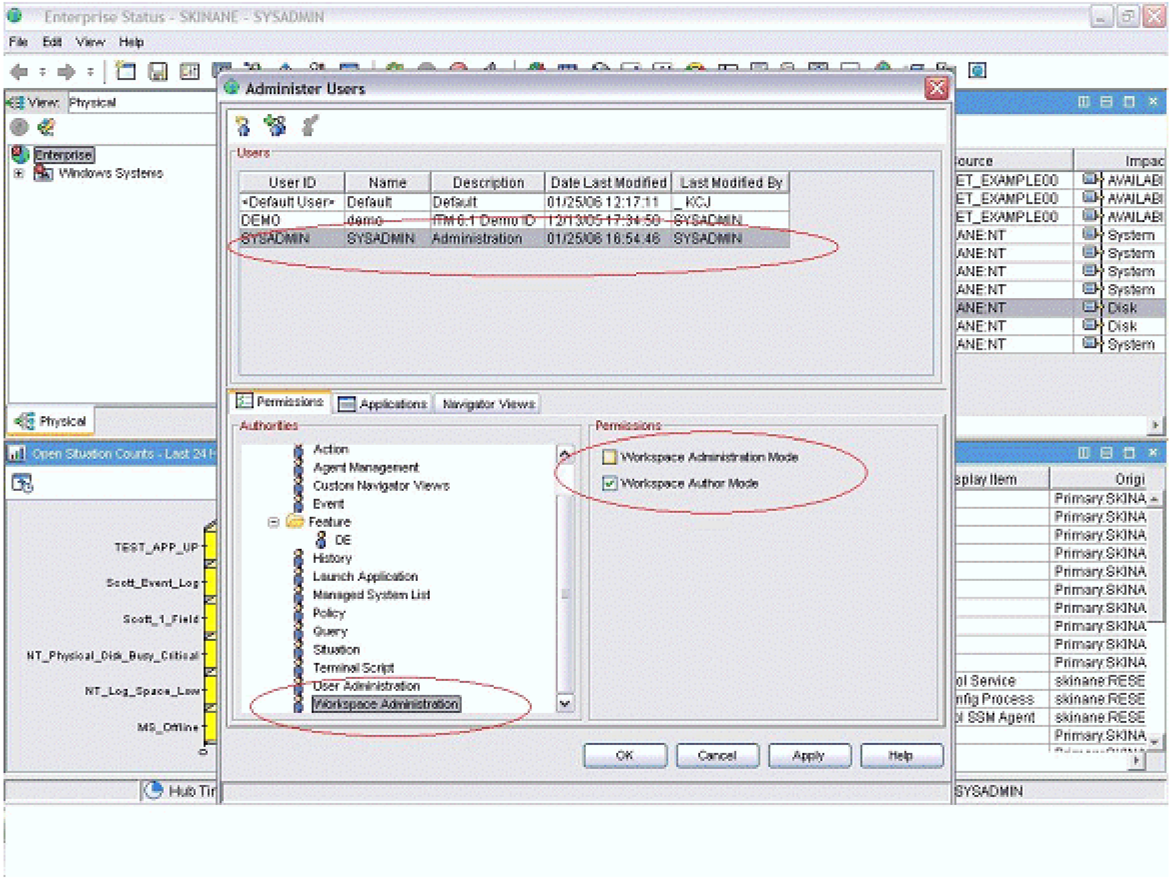
   If you make the selection correctly, ***ADMIN MODE*** is displayed in the desktop title bar.


Enterprise Status - SKINANE - SYSADMIN
File
Edit
View
Help
Properties... Ctrl+R
History Configuration... Ctrl+H
Workflow Editor... Ctrl+W
Situation Editor... Ctrl+E
Administer Users... Ctrl+U
Query Editor... Ctrl+Q
Managed System Lists... Ctrl+M
Physical
Situation Event Console
Total Events: 10
Item Filter: Enterprise
Open Situation Counts - Last 24 Hours
TEST_APP_UP
Scott_Event_Log
Scott_1_Field
NT_Physical_Disk_Busy_Critical
NT_Log_Space_Low
MS_Offline
Count
0
2
4
6
8
10
12
14
16
18
20
Message Log
Hub Time: Wed, 01/25/2006 04:55 PM
Server Available
Enterprise Status - SKINANE - SYSADMIN

Status	Situation Name	Display Item	Source	Impac
Open	TEST_APP_UP	NetCool SSM Agent	skinane:RESET_EXAMPLE00	AVAILABI
Open	TEST_APP_UP	Net Config Process	skinane:RESET_EXAMPLE00	AVAILABI
Open	TEST_APP_UP	Net Cool Service	skinane:RESET_EXAMPLE00	AVAILABI
Open	NT_Log_Space_Low	System	Primary:SKINANE:NT	System
Open	NT_Log_Space_Low	Security	Primary:SKINANE:NT	System
Open	NT_Log_Space_Low	Application	Primary:SKINANE:NT	System
Open	Scott_Event_Log		Primary:SKINANE:NT	System
Open	NT_Physical_Disk_Busy_Critical	0 C:	Primary:SKINANE:NT	Disk
Open	NT_Physical_Disk_Busy_Critical	_Total	Primary:SKINANE:NT	Disk
Open	Scott_1_Field		Primary:SKINANE:NT	System

Status	Name	Display Item	Origi
Open	Scott_Event_Log		Primary:SKINA
Open	Scott_1_Field		Primary:SKINA
Open	Scott_Event_Log		Primary:SKINA
Open	Scott_1_Field		Primary:SKINA
Open	NT_Physical_Disk_Busy_Critical	_Total	Primary:SKINA
Open	NT_Physical_Disk_Busy_Critical	0 C:	Primary:SKINA
Open	Scott_Event_Log		Primary:SKINA
Open	Scott_1_Field		Primary:SKINA
Open	Scott_1_Field		Primary:SKINA
Open	Scott_Event_Log		Primary:SKINA
Open	TEST_APP_UP	Net Cool Service	skinane:RESE
Open	TEST_APP_UP	Net Config Process	skinane:RESE
Open	TEST_APP_UP	NetCool SSM Agent	skinane:RESE
Open	Scott_Event_Log		Primary:SKINA

Figure 61. Setting the `sysadmin` user ID



User ID	Name	Description	Date Last Modified	Last Modified By
<Default User>	Default	Default	01/25/06 12:17:11	_KCJ
DEMO	demo	ITM 6.1 Demo ID	12/13/05 17:34:50	SYSADMIN
SYSADMIN	SYSADMIN	Administration	01/25/06 16:54:46	SYSADMIN

Authorities:

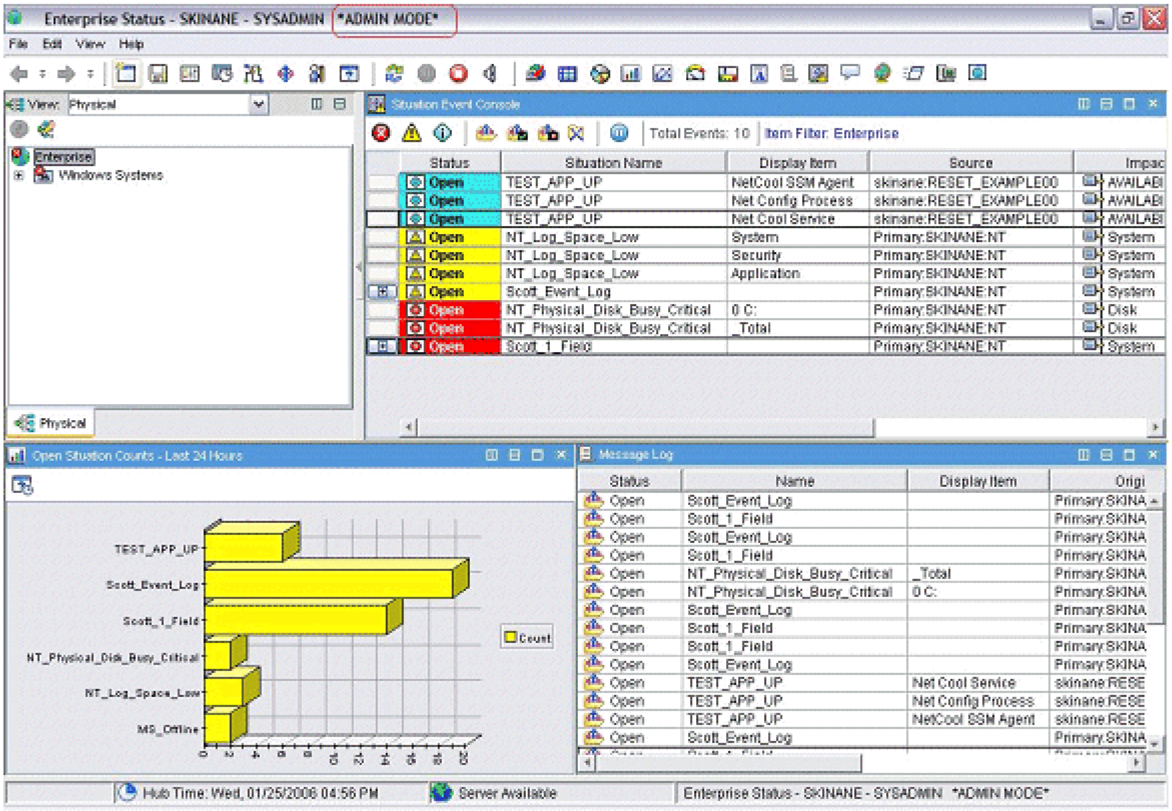
- Action
- Agent Management
- Custom Navigator Views
- Event
- Feature
  - DE
- History
- Launch Application
- Managed System List
- Policy
- Query
- Situation
- Terminal Script
- User Administration
- Workspace Administration

Permissions:

□ Workspace Administration Mode

☑ Workspace Author Mode

Figure 62. Setting the `sysadmin` user ID (continued)



Status	Situation Name	Display Item	Source	Impac
Open	TEST_APP_UP	NetCool SSM Agent	skinane:RESET_EXAMPLE00	AVAILABI
Open	TEST_APP_UP	Net Config Process	skinane:RESET_EXAMPLE00	AVAILABI
Open	TEST_APP_UP	Net Cool Service	skinane:RESET_EXAMPLE00	AVAILABI
Open	NT_Log_Space_Low	System	Primary:SKINANE:NT	System
Open	NT_Log_Space_Low	Security	Primary:SKINANE:NT	System
Open	NT_Log_Space_Low	Application	Primary:SKINANE:NT	System
Open	Scott_Event_Log		Primary:SKINANE:NT	System
Open	NT_Physical_Disk_Busy_Critical	0 C:	Primary:SKINANE:NT	Disk
Open	NT_Physical_Disk_Busy_Critical	_Total	Primary:SKINANE:NT	Disk
Open	Scott_1_Field		Primary:SKINANE:NT	System

Status	Name	Display Item	Origi
Open	Scott_Event_Log		Primary:SKINA
Open	Scott_1_Field		Primary:SKINA
Open	Scott_Event_Log		Primary:SKINA
Open	Scott_1_Field		Primary:SKINA
Open	NT_Physical_Disk_Busy_Critical	_Total	Primary:SKINA
Open	NT_Physical_Disk_Busy_Critical	0 C:	Primary:SKINA
Open	Scott_Event_Log		Primary:SKINA
Open	Scott_1_Field		Primary:SKINA
Open	Scott_1_Field		Primary:SKINA
Open	Scott_Event_Log		Primary:SKINA
Open	TEST_APP_UP	Net Cool Service	skinane:RESE
Open	TEST_APP_UP	Net Config Process	skinane:RESE
Open	TEST_APP_UP	NetCool SSM Agent	skinane:RESE
Open	Scott_Event_Log		Primary:SKINA

Figure 63. Setting the sysadmin user ID (continued)

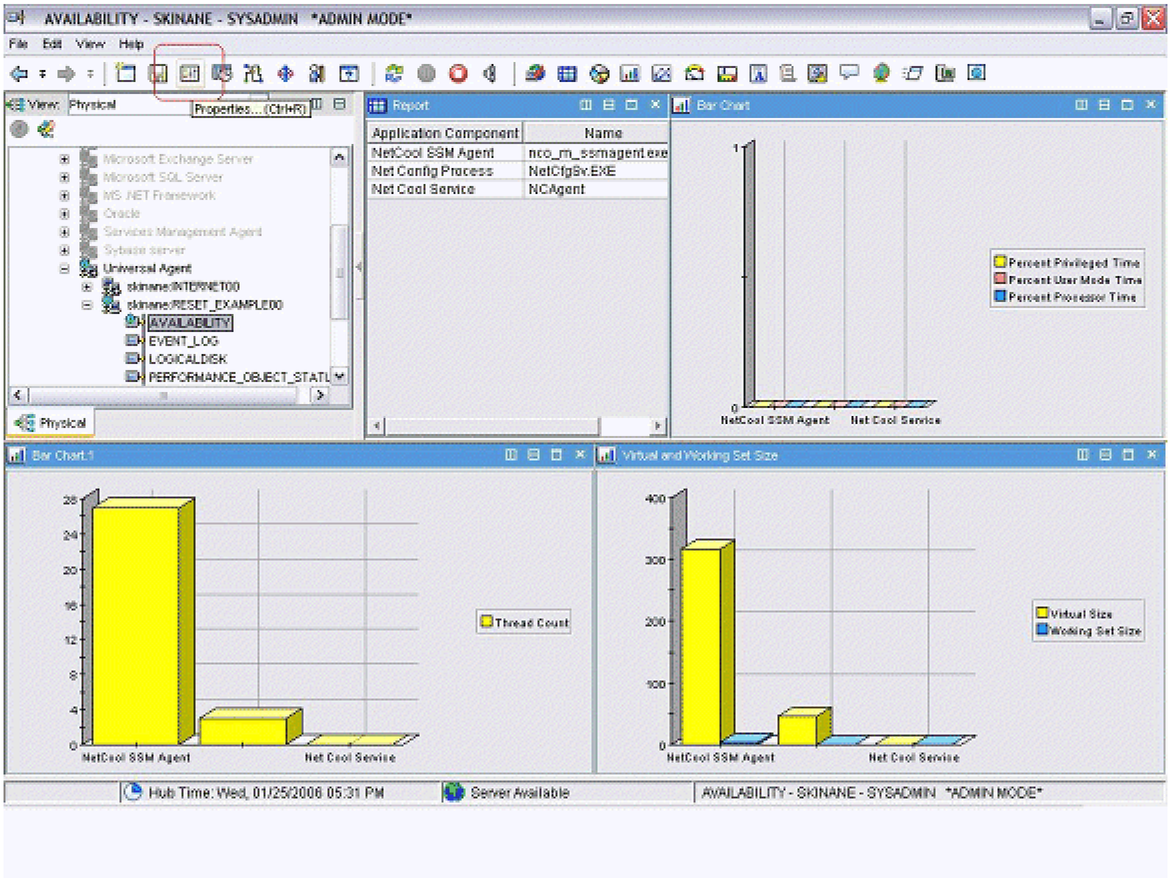
What to do next

After you are in "Administrator" mode as depicted in (Figure 63 on page 1340), you are now ready to create workspaces for your application. For information about how to customize and create workspaces, see the Tivoli Enterprise Portal User's Guide. Alternatively, use the help documentation that is installed with your Tivoli Enterprise Portal component.

If you want your workspaces to be "read-only" and to not be deleted by a customer, set the "not-editable" and "non-deletable" properties for each workspace. In the workspace properties, you must select the following properties:

- **Do not allow modifications**
- **Product-provided by IBM (mark as non-deletable)**

You can go to the properties by either viewing a workspace or clicking the icon with the controls on it. You can also go to one of the view property pages and then going to the workspace level in the properties tree. If you have more than one workspace for each navigator item, remember to set the properties for each workspace. As indicated in the following example screen capture:



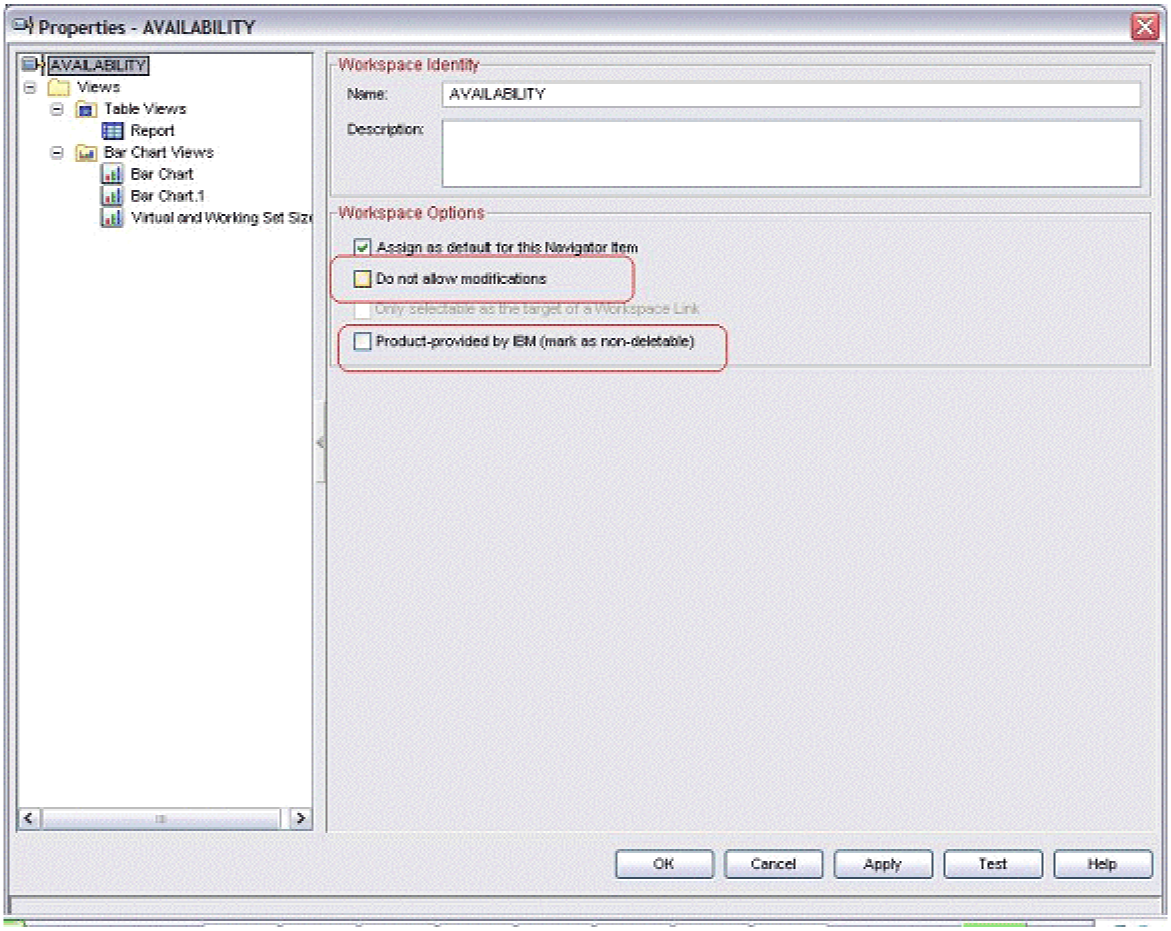
Application Component	Name
NetCool SSM Agent	nco_m_ssmagent.exe
Net Config Process	NetCfgSv.EXE
Net Cool Service	NCAgent

Hub Time: Wed, 01/25/2006 05:31 PM

Server Available

AVAILABILITY - SKINANE - SYSADMIN *ADMIN MODE*

Figure 64. Setting workspace properties



Workspace Identity

Name: AVAILABILITY

Description:

Workspace Options

☑ Assign as default for this Navigator item

□ Do not allow modifications

□ Only selectable as the target of a Workspace Link

□ Product-provided by IBM (mark as non-deletable)

Figure 65. Setting workspace properties (continued)

Preparing the agent for Cloud APM

If you want to use your agent with IBM Cloud Application Performance Management, you need to prepare it using the **Dashboard Setup** wizard. This wizard configures the information that you can see in the summary and detail dashboards in Cloud APM. It also sets the resource information that Cloud APM requires for the agent.

Before you begin

In order to prepare the agent for Cloud APM successfully, you need to ensure that the agent provides the following data:

- One or more data sets (attribute groups) that produce one row of data. You can use the attributes from these data sets to populate the summary dashboard.

  Important: To include any information in the summary dashboard, you need to provide it in a data set that produces a single row of data. Some data sources create data sets that produce multiple rows of data; for example, the process, Windows service, and command return code data sources place data into the single Availability data set, which produces multiple rows. In such cases, you need to create a filtered data set producing one row in order to include the data in a summary dashboard. For instructions, see "Creating a filtered attribute group" on page 1311.
- A numeric attribute within one of these data sets that indicates the status of the monitored service (normal, warning, critical, or other similar status values). You must define status severity values for this attribute. For instructions about defining status severity values, see "Specifying severity for an attribute used as a status indicator" on page 1177.

- If the port number on which the monitored application provides service is fixed, you must know the port. If the port might change between different deployments, one of the data sets that produce one row of data must contain a numeric field that indicates the port.
- If the agent can be installed on a host to monitor a server that is running on a different host, a string attribute within one of these data sets that indicates the server IP address. If the agent always monitors the host where it is running, such an attribute is not required.

  Tip: If an attribute that provides the host name is available, you can create a derived attribute for the IP address by using the `nameToIpAddress` function. For information about creating a derived attribute, see "Creating derived attributes" on page 1170. For information about the function, see "ipAddressToName" on page 1184.

If the agent has subnodes, these requirements apply to each subnode for which you want to create a dashboard.

About this task

Cloud APM monitors *resources*. A resource corresponds to instance of the agent, or sometimes a subnode. To define a resource, you need to supply a resource type name, server name, IP address, and port number that apply to the monitored service.

Cloud APM displays a summary dashboard for every monitored resource. The summary dashboard includes a status indicator; with this indicator (usually green, yellow, or red for normal, warning, or critical status) the user can see the status of the resource at a glance. The same dashboard can contain a few other high-level health metrics.

On the summary dashboard, data is displayed as single items. Therefore, the data set with this data must produce only one row.

Optionally, a detail dashboard can be available for the agent. The user can click the summary dashboard to view the detail dashboard. The detail dashboard can display tables, so data from any data set can be used on this dashboard.

You must select the attributes that are displayed on the summary dashboard (including the status indicator) and on the detail dashboard.

Important: The data in the attributes that you select is automatically passed from the agent to the Cloud APM server every minute. Specifying too much data can lead to overloading of the network, the server, or the monitored host. Select the required attributes only. For example, if a joined data set or a derived attribute must be displayed, do not specify the source attributes as well.

Important: No data other than these attributes is passed to Cloud APM. You cannot view or use other data in Cloud APM, except for thresholds, which are monitored at the agent level. If you use other data in thresholds, you might not be able to view the threshold status in the Cloud APM console.

Procedure

1. From the **Agent Information** view, click the **Dashboards** link.
2. Under **Dashboard Components**, select **Show agent components in the dashboard**.

   Tip: Alternatively, if you are creating an agent for use exclusively with IBM Tivoli Monitoring, you can select **No dashboard presence for this agent**. In this case, do not complete the subsequent steps of this procedure. You can not install such an agent in a Cloud APM environment.
3. Click the **Dashboard Setup Wizard** link.
4. If the agent has subnodes, define the arrangements of agent and subnode resources in Cloud APM:
   - Select **Base agent instances** to display the base agent (data outside of subnodes) as a resource.
   - For every subnode, select **Subnode "*name*" instances** to display this subnode as a resource.
   - Optionally, for any of the selected subnodes, select **Show as child of agent**. In this case, the subnode resource is displayed as a child under the agent resource in lists in the Cloud APM console.

Cloud APM displays a summary and detail dashboard for each of the components you selected.

Important: If you run the wizard again and unselect an agent or subnode, the resources for the agent or subnode are not removed automatically. To remove the resources, expand **Resources** in the Outline view, select the resources to be deleted, and press the Delete key on the keyboard.

5. In the **Attribute Selection - Status** page, select the attribute that indicates the status of the monitored service. Numeric attributes from groups that return a single data row are available.

   Tip: Alternatively, if you do not want to display status in the dashboard, unselect **Provide status for this agent**.
6. In the same page, you can select whether you want to display additional data in the summary and detail dashboards:

   - To display additional high-level health metrics in the summary dashboard, ensure the **Select additional attributes to display in this agent's summary information** box is selected. Otherwise, clear the box.
   - To display additional data in the detail dashboard, ensure the **Select additional attributes to display in this agent's detail information** box is selected. Otherwise, clear the box. (Typically, select this box, as a detail dashboard is required to display enough data to make a monitoring agent meaningful).

   Click **Next**.
7. If you selected **Select additional attributes to display in this agent's summary information**, in the **Attribute Selection - Summary** page, select up to four additional attributes to include in the summary dashboard. Attributes from groups that return a single data row are available. Click **Next**.
8. If you selected **Select additional attributes to display in this agent's detail information**, in the **Attribute Selection - Details** page, select the attributes to include in the detail dashboard. All attributes in the agent are available; to avoid performance issues, include as few attributes as possible. Click **Next**.
9. In the **Resource Type** page, enter the server type that you are monitoring, for example, `Email server` or `SampleCo Database Server`. Click **Next**.
10. In the **Attribute Selection - Software Server Name** page, enter a fixed software server name in the **Fixed Name** field or select an attribute from your agent that gives the software server name. This name is displayed to the user for this particular monitored instance, for example, the name of the JBoss application server instance. Click **Next**.

    Important: Do not run two or more monitoring agents, agent instances, or subnodes with the same software server name on the same monitored host. If your agent has instances or subnodes, ensure that a unique software server name is generated for every instance or subnode. If two different agents produce the same software server name, do not install them on the same monitored host.
11. In the **Attribute Selection - IP address** page, select an attribute from your agent that specifies the IP address (not host name) of the primary interface connection that the monitored server or application uses. For example, the HTTP connection for an HTTP server or the database client connection for a database server. Alternatively, select **Use the agent's IP address** to use the address of the host where the agent runs. Click **Next**.
12. In the **Attribute Selection - Port** page, enter the port on which the monitored application provides service or select a numeric attribute from your agent that specifies this port. Click **Finish**.
13. If you selected both an agent and a subnode or more than one subnode as resources, click **Next** to enter dashboard and resource information for the next component (agent or subnode). If the **Next** button is disabled, you entered the information for all necessary components; click **Finish** to complete the wizard.

Results

When you install the agent on a monitored host, you can view the summary and detail dashboards in the **Status Overview** tab.

Important: There can be a delay of up to 30 minutes between installation of the agent and availability of the dashboards, especially if this is the first time that this agent type and version is installed in your environment.

Click the summary dashboard for the agent to view the detail dashboard. By default, all information in the detail dashboard is displayed as tables.

You can use the **Attribute Details** tab to configure custom display of this information as tables and charts.

Preparing the agent for Cloud Pak for Multicloud Management

If you want to use your agent with IBM Cloud Pak for Multicloud Management, you need to prepare it using the **Resource wizard**. The Resource wizard creates a resource definition.

Before you begin

The resource definition process reads the agent definition and allows you to build resources. But if you then edit the agent definition, for example, add an attribute, rename an attribute, add an attribute group, or delete either one, these changes are not automatically reflected in the resource definition. For this reason ensure that you have the correct data before you run the Resource wizard.

About this task

Resources are defined by resource types and when the agent runs it will create resource instances that are displayed in the console. Resource types define a set of properties and metrics that provide the identity and metrics related to a managed resource.

Use the Resource wizard to create or edit a resource definition for the agent. The resource definition then automatically generates the widgets for the Resources dashboard. In the Resource wizard, you define an attribute group; this attribute group provides the data for the basic widgets in the Resources dashboard. Optionally, you can also add components; components are extra tables of data that appears in the Resources dashboard below the Events timeline widget. You can add as many component tables as required.

When you run the resource wizard the following widgets are created and displayed in the Resources dashboard in Cloud Pak for Multicloud Management:

- Events timeline
- Line graphs that display data for the "important" metrics (see step “5” on page 1345 for a description)
- Tabular display of the metrics in the main resource attribute group
- Optional (s) for each component created
- Related resources widget

In addition, if you have a status attribute in the attribute group that defines a resource, a threshold will automatically be created in the Cloud Pak for Multicloud Management.

Run the Resource wizard once per resource you want to define or edit for the agent.

Procedure

1. Select **Agent Definition**>**Resources**, the **Introduction** window is displayed, click **Next**.
2. In the **Resource Information** window, select **Add new Resource**, click **Next**.
3. In the **Attribute Group Selection** window, select an attribute grouping. At a minimum, the attribute grouping needs to identify the resource.
4. In the **Resource Information** window, enter a name and description for the resource. The name that you enter corresponds to the name you see in the Resources types list in the Resources dashboard. The description also appears in the resource list in the Resource tab. You can use any characters for name or description.
5. In the **Attributes** window, you can assign different characteristics to your attributes.

- **Not Available:** Use this to eliminate the attribute completely from both the Resources dashboard and the Threshold manager.
- **Not uploaded to the server, available for threshold calculation:** Use this to indicate that the attribute data isn't uploaded by default. Use this if you don't want this attribute to be displayed in the Resources dashboard, but you do want to be able to create thresholds against it in the Threshold manager.
- **Metric describes the state of the resource instance at a given time**. Use this to assign a string as a metric. Usually all strings are properties and all numbers are metrics (unless you earlier explicitly specified otherwise in the Attribute editor, for more information, search for 'purpose' in table 1 in the “Numeric aspects of attributes” on page 1175 topic.
- **Property describes the resource instance:** Use this to assign a numeric as a property. By default all the strings are properties and all the numbers are metrics.
- **Display Name:** By default the display name and key property is the node name. This is indicated with "D" in the **Opt** column. When node name is the display name, you see the origin node name from IBM Tivoli Monitoring displayed in the selection lists in the Resources dashboard. If there is an attribute that is more suited to being the display name, select Display name for that attribute.
- **Important**: This indicates that a metric is represented in a line chart in the Resources dashboard. The label and grouping for the line chart is based on the unit that you specified when you defined the attribute. For more information, see “Numeric aspects of attributes” on page 1175. It is recommended indicating 1 - 4 metrics as important.

6. If you want more tables to present extra data in the resource, add a component, you add one component for each table you want to add to a resource. In the **Component selection** window, select **Add a new component**. Click **Next**.
7. In the **Component Information** window, enter a name and description for the component. The name that you enter corresponds to the name that is given to the table in the Resources dashboard. You can use any characters for name or description. Click **Next**
8. Repeat step “3” on page 1345 and “5” on page 1345. Click **Finish**.

Results

You can now view the Resource dashboard in the Cloud Pak; you will see representations of your resource definition in the following widgets in the Resources dashboard:

- Events timeline
- Line graphs that display data for the "important" metrics (see step “5” on page 1345 for a description)
- Tabular display of the metrics in the main resource attribute group
- Optional (s) for each component created
- Related resources widget

.

Data Definition Designer

For information about the Data Definition Designer, see The Data Definition Designer guide.

Testing your agent in Agent Builder

After you use Agent Builder to create an agent, you can test the agent in Agent Builder.

Test the agent to ensure that the monitoring data you are expecting is the data that is being displayed. By testing your agent, you can learn to modify or tweak settings in the agent to ensure that the data displayed is beneficial and accurate.

You can test your agent in Agent Builder by using the following methods:

1. Begin by using the attribute group test function of Agent Builder to test individual attribute groups one at a time. For more information, see “Attribute group testing” on page 1347.

2. After you complete attribute group testing, you can use the agent test function of Agent Builder to test all attribute groups in your agent together. For more information, see “Full agent testing” on page 1350.

Important: When testing your agent in Agent Builder, you can see the following special values for numeric attributes:

- -1: a general error
- -2: missing data
- -3: no value (for example, NULL was returned by a database)

Attribute group testing

You can use attribute group testing to test the attributes groups of the agent you created with Agent Builder, one attribute group at a time. You can test many attribute groups before you complete the attribute group definition. For example, you can initiate testing from the **IBM Tivoli Monitoring Agent Wizard** when you are defining the attribute groups of a new agent. You can also initiate testing from the **IBM Tivoli Monitoring Agent Component Wizard** when you are adding attribute groups to an existing agent.

Before you begin

Before you start testing an attribute group, you can optionally:

- Set attribute group testing preferences. For more information, see “Attribute group testing - preferences” on page 1348.
- Set environment variables, configuration properties, and where applicable Java information. For more information, see “Attribute group testing - configuration” on page 1349.

About this task

Agent Builder supports an attribute group test function for most data sources

Procedure

- Start the Testing procedure in the following ways:

  1. During agent or attribute group creation click **Test** on the relevant data source Information page.
  2. After agent creation, select an attribute group on the Agent Editor **Data Source Definition** page and click **Test** . For more information about the Agent Editor, see “Using the Agent Editor to modify the agent” on page 1152.

  After you click **Test** in one of the previous two steps, the Attribute group Test window is displayed. This window is different for different data sources,

  Agent Builder supports an attribute group test function for most data sources.

  For more information about the test procedures for specific attribute groups, see the following Testing sections:

  – Windows Management Instrumentation (WMI), for more about the WMI test procedure, see “Testing WMI attribute groups” on page 1202
  – Windows Performance Monitor (Perfmon), for more about the Perfmon test procedure, see “Testing Perfmon attribute groups” on page 1204
  – Simple Network Management Protocol (SNMP), for more about the SNMP testing, see “Testing SNMP attribute groups” on page 1208
  – Simple Network Management Protocol (SNMP) event sender, for more about the SNMP event test procedure, see “Testing SNMP event attribute groups” on page 1212
  – Java Management Extensions (JMX), for more about the JMX test procedure, see “Testing JMX attribute groups” on page 1232

- Common Information Model (CIM), for more about the CIM test procedure, see "Testing CIM attribute groups" on page 1234
- Log file, for more about the log file test procedure, see "Testing log file attribute groups" on page 1244
- Script, for more about the script test procedure, see "Steps for monitoring output from a script" on page 1255
- Java Database Connectivity (JDBC), for more about the JDBC test procedure, see "Testing JDBC attribute groups" on page 1265
- Internet Control Message Protocol (ICMP) ping, for more about the ICMP test procedure, see "Testing Ping attribute groups" on page 1267
- Hypertext Transfer Protocol (HTTP) Availability, for more about the HTTP test procedure, see "Testing HTTP attribute groups" on page 1275
- SOAP, for more about the SOAP test procedure, see "Testing SOAP attribute groups" on page 1283
- Transmission Control Protocol socket (TCP) socket, for more about the socket test procedure, see "Testing socket attribute groups" on page 1292
- Java application programming interface (API), for more about the Java API test procedure, see "Testing Java application attribute groups" on page 1305

Some data sources do not have an attribute group test function, for example:

- When you can use the Agent Builder browser to view live data on a system. For example, you can view the processes that are currently running on the system (processes). Other examples are when you can view the services that are installed on the system (windows services) and the Windows Event Logs that are present.
- There is little or no customization that you can do in the agent (AIX Binary Log, command return code).
- Joined and Filtered attribute groups cannot be tested by using the attribute group test function because these groups are based on multiple attribute groups.

Note:

1. Use the full agent test to test data sources that cannot be tested by using the attribute group test function. For more information about the full agent test, see "Full agent testing" on page 1350.
2. When you test data sources, after you click **Collect Data**, data might not be displayed at all or might not be current after the first click. In such cases, click **Collect Data** a second time to display current data.

- Debugging:

  Each data source that is tested has a test directory that is created for it by Agent Builder. This directory is used for the test runtime environment of the data source. Log files that relate to tests run on the data source are stored under this directory. The log files can be useful to help debug issues that are found during testing.

  Note:

  1. The location of the test log file is shown as a status message in the **Test** window after you click **Start Agent** and also after you click **Stop Agent**.
  2. All test data source directories are deleted when the Agent Builder is shut down.

Attribute group testing - preferences

Set preferences before you test an attribute group.

About this task

Before you start testing an attribute group, you can optionally set some preferences that determine how attributes are treated during testing.

Procedure

1. Select **Window** > **Preferences** from the Agent Builder menu bar.

   The **Preferences** window opens.

2. Select **Agent Builder**.

   The preferences that are associated with testing attribute groups are shown:

   Show data types changed dialog when testing
   When selected, Agent Builder suggests changes to the data type of an attribute. Agent Builder suggests changes when the data type of an attribute does not match the data that is returned by a test for that attribute. For example, if the string length defined for an attribute is too short to hold a value that is returned by a test. In this example, Agent builder suggests redefining the attribute to have a longer string length. When this option is cleared, Agent Builder does not check or suggest data types during testing. This option is selected by default.

   Maximum script or log attributes created
   The value that is entered in this field determines the maximum number of attributes that Agent Builder parses during the initial test of a log file or script attribute group. The default value is 25.

3. When you are finished setting your preferences, click **OK** to save your settings and close the **Preferences** window.

   If you want to restore the default settings, click **Restore Defaults** before you click **OK**

Attribute group testing - configuration

Set environment variables, configuration properties, and Java information before you test an attribute group.

About this task

Before you start testing an attribute group, you can optionally set environment variables, configuration properties, and where applicable Java information from the data source Test window. The Java information is a subset of the configuration data. Some environment variables have special values that are set by default for attribute group testing. For more information about environment variables with special values for attribute group testing, see "Test Environment variables" on page 1354.

Procedure

1. Optional: Click **Set Environment** from the data source **Test** window.
   The **Environment Variables** window opens. When populated, the **Environment Variables** window lists all of the environment variables that are used during the running of the test. The initial view of the Environment variable window contains any existing environment variables that are defined in your agent. It also contains any environment variables that you added from previous tests of this agent.

   a) Click **Add** or **Edit** to add or edit individual variables.

   b) Click **Remove** to remove individual variables, or **Restore Default** to restore default variables and remove all others.

   c) Click **OK** to save your changes and return to the **Test** window.

2. Optional: Click **Configuration** from the data source **Test** window. The **Runtime Configuration** window opens.

   a) Click **Edit Agent Configuration** to add a configuration property or to edit existing agent configuration properties by using the **Configuration Properties** window.

   b) Select a configuration property and click **Edit** to edit an existing configuration property that relates to the attribute group you are testing.

   c) Select a configuration property and click **Restore Default** to restore a configuration property to its default value.

   Important: If a JMX data source connects to a remote WebSphere Application Server, ensure that a local WebSphere Application Server is installed and that the Java location is set to the JRE that this server uses. For details about setting up the connection, see "Monitoring Java Management Extensions (JMX) MBeans" on page 1214.

3. Click **OK** to save your changes and return to the **Test** window.
4. **Note:** You can set Java information for following types of attribute groups:

   - Java Management Extensions (JMX)
   - Java Database Connectivity (JDBC)
   - Hypertext Transfer Protocol (HTTP) Availability
   - SOAP
   - Java application programming interface (API)

   The Java information is a subset of the configuration data described in step "2" on page 1349

   Optional: Click **Java Information** from the data source **Test** window.

   The **Java Information** window opens.

   a) Enter Java Information.
      For example, Browse to or type the location of the Java Runtime Environment (JRE), select a **Java trace level**, or enter **JVM arguments**
   b) Click **OK** to save your changes and return to the **Test** window.

Full agent testing

Use full agent testing to test all attribute groups of your agent together. You can also use full agent testing to test data sources that cannot be tested by using the attribute group test function.

About this task

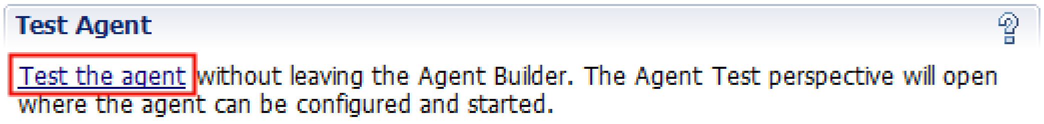
You can use full agent testing to run the agent in the same way it runs in IBM Tivoli Monitoring without needing an IBM Tivoli Monitoring installation.

Important: On Windows systems, If you want to run a full test of the agent inside Agent Builder (see "Full agent testing" on page 1350), ensure that the 32-bit version of the operating system on which you are running the Agent Builder, that is, 32-bit Windows, is selected in the Agent Information window. On Linux systems, the 64-bit version must be selected.

Procedure

1. Open the **Agent Test** perspective:

   a) In the agent editor, open the **Agent Information** tab.
   b) Click **Test the agent**.



Test Agent

Test the agent without leaving the Agent Builder. The Agent Test perspective will open where the agent can be configured and started.

*Figure 66. **Test Agent** section of the **Agent Editor, Agent Information** page.*

   Alternatively, from the Agent Builder menu select **Window** > **Open Perspective** > **Other**, select **Agent Test** and click **OK**

   The **Agent Test** perspective opens (Figure 68 on page 1352). The **Agent Test** view shows agents that you have opened in the agent editor; you can test any of these agents. An **Attribute Group Test** view is also displayed; this view is initially empty. The **Attribute Group Test** view shows data that is collected from a selected attribute group when the agent is running.

   Tip: If no agents are being edited, the **Agent Test** perspective is empty. To populate the view, go to the **IBM Tivoli Monitoring** perspective and open an agent in the **Agent Editor**. When an agent is opened in the **Agent Editor** return to the **Agent Test** perspective to test the agent.
2. Optional: Configure environment variables and configuration properties before you start the test.

You can access the **Environment Variables** and **Runtime Configuration** windows in two ways from the **Agent Test** view:

- Right-click the agent in the **Agent Test** view to open a selection menu. You can select **Set Environment** from the menu to open the **Environment Variables** window. You can select **Configuration** from the menu to open the **Runtime Configuration** window.
- Click the view menu icon ▽ on the **Agent Test** view toolbar to access the **Set Environment** and **Configuration** menu items as in the previous choice.

For more information about using the **Environment Variables** and **Runtime Configuration** windows, see "Attribute group testing" on page 1347.

Important:

a. The agent is populated automatically with the last set of configuration that relates to each tested attribute group.

b. Some environment variables can have different default values for attribute group testing and for full agent testing. For more information about environment variables with special values for attribute group testing, see ("Test Environment variables" on page 1354).

c. If a JMX data source connects to a remote WebSphere Application Server, ensure that a local WebSphere Application Server is installed and that the Java location is set to the JRE that this server uses. For details about setting up the connection, see "Monitoring Java Management Extensions (JMX) MBeans" on page 1214.

d. In a Java API, JDBC, JMX, HTTP, or SOAP data source, you can use the **Java** > **JVM arguments** setting to control agent trace logging. Set the following value:

```
-DJAVA_TRACE_MAX_FILES=files -DJAVA_TRACE_MAX_FILE_SIZE=size
```

where *files* is the maximum amount of trace log files that are kept (the default value is 4) and *size* is the maximum log file size in kilobytes (the default value is 5000). For example, you can set the following value:

```
-DJAVA_TRACE_MAX_FILES=7 -DJAVA_TRACE_MAX_FILE_SIZE=100
```

In this case, the agent writes 100 kilobytes into the first log file, then switches to the second log file, and so on. After writing seven log files of 100 kilobytes each, it overwrites the first log file.

e. If your agent has subnodes, in an installed version you can set different configuration values for different subnodes and separately for the base agent attribute groups. However, in full agent testing configuration you can only set every configuration value once; the setting applies to the base agent and any subnodes. You can only test one instance of every subnode.

3. In the **Agent Test** view, select the agent that you want to test and click the ▶ **Start Agent** icon.

   A window indicates that the agent is starting. When the agent starts, its attributes groups are shown as children of the agent in the **Agent Test** view. The attribute groups are indicated by the attribute group icon ▯.

   The status attribute groups that give information about the agent (**Performance Object Status**, **Thread Pool Status** and **Take Action Status**) are also shown as children of the agent in the **Agent Test** view. The status attribute groups are indicated by the i information icon.

   You can start and run more than one agent at the same time.

   The ■ **Stop Agent** icon becomes available when the agent is started.

   If your agent has subnodes or navigator groups, they are shown as nodes in the **Agent Test** view. Subnode definitions are shown under the agent. A subnode instance node is shown under the subnode definition node. Attribute groups and navigator groups are shown under the subnode instance node. For example:

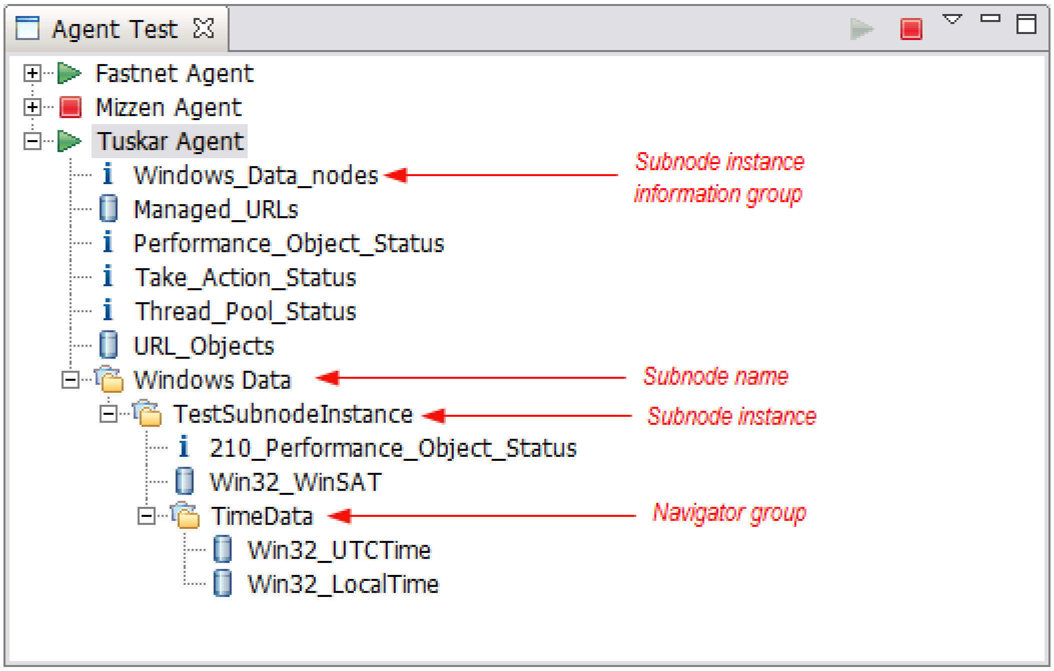


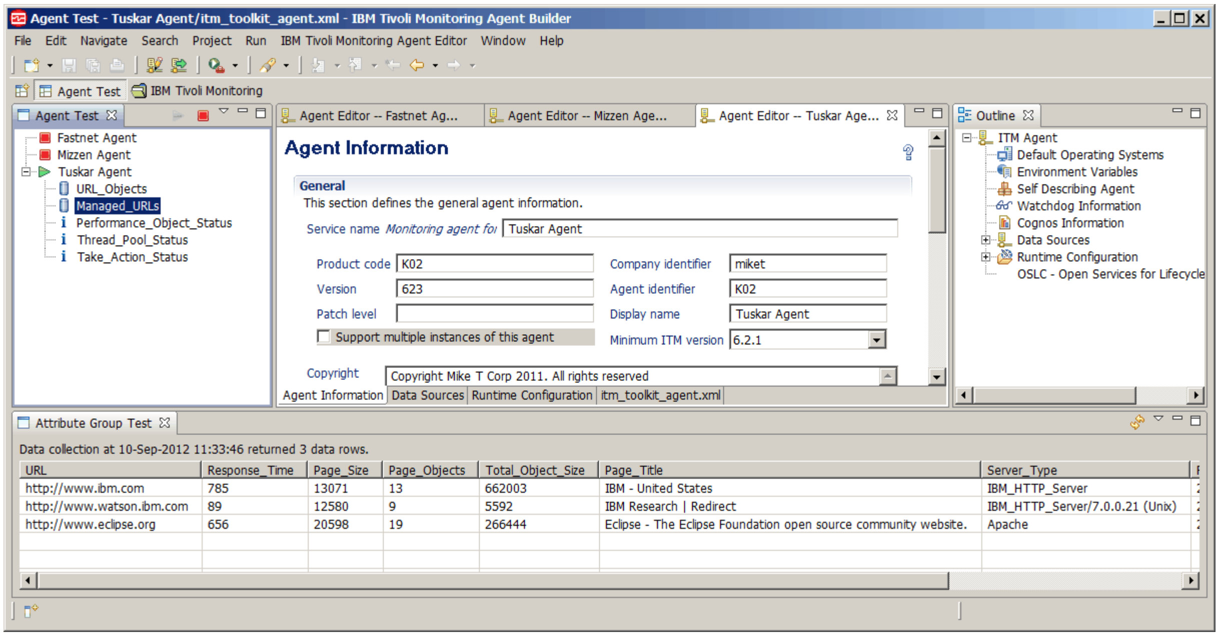
Figure 67. ***Agent Test*** *view with example subnode and navigator group highlighted.*

You can right-click on any of the nodes in the **Agent Test** view to access menu selections like **Edit** and **Stop Agent**. **Edit** opens the **Data Source Definition** for the selected node in the **Agent Editor**.

Note: Changes that you make with the **Agent Editor** are not visible in the running agent until you stop and restart the agent.

4. In the **Agent Test** view, select the first attribute group that you want to test.

When you select an attribute group, a data collection begins for the selected attribute group. If the collection takes some time, a window indicates that the data collection is in progress. When the data collection is complete the collected data is displayed in the **Attribute Group Test** view, for example:



Agent Information

General

This section defines the general agent information.

Service name *Monitoring agent for* Tuskar Agent

Product code K02

Version 623

Patch level

□ Support multiple instances of this agent

Company identifier miket

Agent identifier K02

Display name Tuskar Agent

Minimum ITM version 6.2.1

Copyright Copyright Mike T Corp 2011. All rights reserved

Data collection at 10-Sep-2012 11:33:46 returned 3 data rows.

URL	Response_Time	Page_Size	Page_Objects	Total_Object_Size	Page_Title	Server_Type
http://www.ibm.com	785	13071	13	662003	IBM - United States	IBM_HTTP_Server
http://www.watson.ibm.com	89	12580	9	5592	IBM Research | Redirect	IBM_HTTP_Server/7.0.0.21 (Unix)
http://www.eclipse.org	656	20598	19	266444	Eclipse - The Eclipse Foundation open source community website.	Apache

Figure 68. ***Agent Test*** *perspective*

If no data is displayed, a message `0 data rows returned` is shown in the **Attribute Group Test** view. There are several reasons why the agent might not return data. These reasons include:

- There is no data
- Incorrect definition
- Incorrect configuration

You can check the reason why no data is returned by looking at the value of the **Error_Code** in the **Performance Object Status** attribute group. For more information about viewing the **Performance Object Status** attribute group, see step "9" on page 1353

To collect data for another attribute group in the running agent, select the required attribute group.

When you select an attribute group in the **Agent Test** view, the corresponding attribute group is displayed in the **Agent Editor** view.

5. Optional: Run a second data collection, after the initial data collection, for certain attribute group types, to get useful data values.

   To run a data collection, click the collect data icon in the **Attribute Group Test** view.

   If the collection takes some time, a window indicates that a data collection is in progress. When the data collection is complete, the newly collected data is displayed in the **Attribute Group Test** view.

6. Optional: Click an attribute column heading in the **Attribute Group Test** view to open the **Attribute Information** in the **Agent Editor Data Source Definition** tab. You can also access the same **Attribute Information** by right-clicking on any data cell in the table and choosing **Edit** from the menu.

   You can edit properties of the attribute in the normal way. Changes that you make are not visible in the running agent until you stop and restart the agent.

7. Optional: Open multiple **Attribute Group Test** views at the same time.

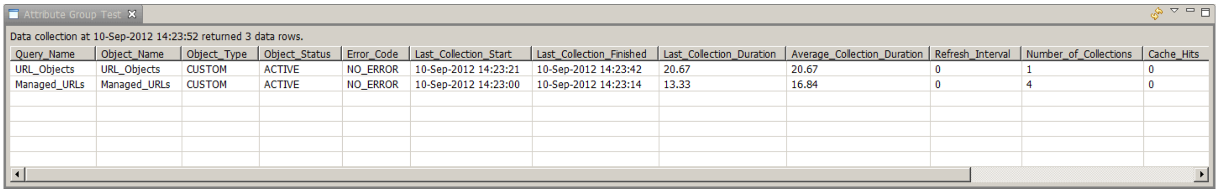
   To open an additional **Attribute Group Test** view, click the view menu icon ▽ on the **Attribute Group Test** view toolbar and then select **Open view for attribute group**.

   Note: When an additional **Attribute Group Test** view is opened, it displays the same attribute information as the original **Attribute Group Test** view. You can then select another attribute group in the **Agent Test** view to display different attribute group information in the original **Attribute Group Test** view. The first time another **Attribute Group Test** view is opened, it opens in the same location as the original view but with its own tab. If you want to see the two views simultaneously, you can drag the tab to another location in the workspace.

8. Optional: Select the subnode instance information attribute group, if your agent has subnodes, to see how the subnodes are listed in your agent (Figure 67 on page 1352). Selecting the subnode instance information attribute group shows subnode instance information in the **Attribute Group Test** view (for all online subnodes of the selected type).

9. Optional: To see more information about the operation of the agent, you can select the **Performance Object Status** and **Thread Pool Status** attribute groups in the **Agent Test** view. These status attribute groups are indicated by the information icon i. Select these groups to see status information about earlier data collections for your attribute groups.

   For example:



Data collection at 10-Sep-2012 14:23:52 returned 3 data rows.

Query_Name	Object_Name	Object_Type	Object_Status	Error_Code	Last_Collection_Start	Last_Collection_Finished	Last_Collection_Duration	Average_Collection_Duration	Refresh_Interval	Number_of_Collections	Cache_Hits
URL_Objects	URL_Objects	CUSTOM	ACTIVE	NO_ERROR	10-Sep-2012 14:23:21	10-Sep-2012 14:23:42	20.67	20.67	0	1	0
Managed_URLs	Managed_URLs	CUSTOM	ACTIVE	NO_ERROR	10-Sep-2012 14:23:00	10-Sep-2012 14:23:14	13.33	16.84	0	4	0

*Figure 69. The **Attribute Group Test** view that shows more information (Performance Object Status) about data collections for the **Managed_URLs** and **Managed_Nodes** attribute groups*

10. When you are finished testing your agent, click the stop agent icon

Test Environment variables

Use these environment variables to control the behavior of the agent during testing.

Environment variables are dynamic named values that determine how the agent runs. For attribute group test, some agent environment variables are set to special values. The special values are used so that the agent responds in a way that suits the testing of a single attribute group. For full agent test special values are not used and instead the default values are used. The default values mean that the agent behaves as it normally would, which is more appropriate to full agent testing.

The environment variables that have special values for attribute group testing are summarized in the following table. For more information about all agent environment variables, see ("List of environment variables" on page 1154). For more information about setting environment variables, see ("Environment variables" on page 1154).

Table 302. Environment variables

Environment variable	Default value (full agent test)	Attribute group test value	Reason for changed value for attribute group test
`CDP_DP_INITIAL_COLLECTION_DELAY`	varies	1	This value applies to an agent with a thread pool. This value is the time in seconds that the thread pool waits before the initial data collection request is sent to a data provider. **Note:** If `CDP_DP_INITIAL_COLLECTION_DELAY` is not set, the thread pool waits for a time that is specified by `CDP_DP_REFRESH_INTERVAL` or `CDP_ATTRIBUTE_GROUP_REFRESH_INTERVAL`. This wait time is the same time the thread pool waits between data collections, and might be too long to wait for the first data collection.
`CDP_DP_CACHE_TTL`	55	1	When set to 1 a Collect Data request is much more likely to cause the data provider to collect data immediately. Otherwise it might return cached data that is up to 60 seconds old.

Installing your agent into a monitoring infrastructure for testing and use

After you test your agent in Agent Builder, you can install the agent into an existing IBM Tivoli Monitoring, IBM Cloud Application Performance Management or IBM Cloud Pak for Multicloud Management environment for further testing and for use.

Installing and testing your agent in a monitoring infrastructure has the following benefits:

- You can configure and test multiple instances of an agent that run simultaneously.
- You can configure and test multiple instances of subnodes that run simultaneously.
- In a Tivoli Monitoring environment, you can build workspaces, situations, actions, and queries in the Tivoli Enterprise Portal.

Important: Deploy initial versions of your agent into a test version of the monitoring infrastructure. On Tivoli Monitoring, use a separate monitoring server and portal server. On Cloud APM, use a test cloud account or a separate test deployment of the on-premises monitoring server. Deploy the final version of your agent on a production infrastructure.

If you deploy a version of the agent on the production monitoring infrastructure and then change any data sets in the agent, the new version might conflict with the older version on the server. In this case it might be impossible to use any version of the agent.

Installing an agent

There are two methods for installing the agents that you create with Agent Builder.

1. To test your agent with a monitoring infrastructure that is running on the same system as the Agent Builder, you can install the agent into the local Tivoli Monitoring or Cloud APM installation.
2. To test or use the agent with a Tivoli Monitoring or Cloud APM system that is not running on the same system as the Agent Builder, you can generate a compressed file (*agent package*) that you can transfer to the other systems and deploy.

Note:

1. With Tivoli Monitoring, after you install an agent, you can see performance metrics in the Tivoli Enterprise Portal tables. For support of situations or workspaces, see "Importing application support files" on page 1371.
2. With Tivoli Monitoring, after you install the agent, you can use the Tivoli Enterprise Portal to verify the data from the agent. For more information, see "Changes in the Tivoli Enterprise Portal" on page 1365. If after you view the data in the Tivoli Enterprise Portal, you want to modify the agent, see "Using the Agent Editor to modify the agent" on page 1152.
3. For an agent that supports Linux or UNIX, generate the installer image on a Linux or UNIX system because a Linux or UNIX system creates the files with the appropriate permissions.

Installing an agent locally

Install the agent into a monitoring environment on the local system where Agent Builder is running.

About this task

Complete the following steps to install your agent into a monitoring environment on the local system:

1. Click the `itm_toolkit_agent.xml` file from the Project Explorer navigation tree of Agent Builder by using one of the following methods:
   a. Right-click the `itm_toolkit_agent.xml` file and select **IBM** > **Generate Agent**.
   b. Select the `itm_toolkit_agent.xml` file and select the **Generate Agent** icon on the toolbar.
   c. Double-click the `itm_toolkit_agent.xml` file and select **Agent Editor** > **Generate Agent**.
2. In the **Generate Agent Wizard** window, in the **Install the Agent Locally** section, enter the installation directory for the monitoring infrastructure. The Agent Builder completes the value that is found in the `CANDLE_HOME` environment variable. If this variable is not set, the default value for Windows, `C:\IBM\ITM`, is displayed.

   The check boxes are enabled as follows:

   Install the agent
   Enabled if the Agent Builder detects an appropriate Tivoli Enterprise Monitoring Agent or a IBM Cloud APM agent in the specified location. An appropriate agent is one that supports the local operating system and is the correct minimum version.

   Install the TEMS support
   Enabled in a Tivoli Monitoring environment if the Agent Builder detects a Tivoli Enterprise Monitoring Server in the specified location.

   Install the TEPS support
   Enabled in a Tivoli Monitoring environment if the Agent Builder detects a Tivoli Enterprise Portal Server in the specified location.
3. Select the components to install (agent, Tivoli Enterprise Monitoring Server support, Tivoli Enterprise Portal Server support).

4. In a Tivoli Monitoring environment, if the Tivoli Enterprise Monitoring Server or Tivoli Enterprise Portal Server is installed on the local computer and you are installing the support files for these servers, you can choose whether to restart the servers.

   In this case, the **Restart TEMS without credentials** and **Restart TEPS** check boxes are active in the **Install the Agent Locally** section of the Generate Agent wizard. You can clear the check boxes to install the support without recycling the servers.

   When you clear the **Restart TEMS without credentials** check box, you are prompted for the Tivoli Enterprise Monitoring Server user ID and password. Enter these details and click **Logon**. If you are running Tivoli Monitoring with security off, enter "sysadmin" for the user ID, leave the password blank, and click **Logon**.

   Alternatively, to continue without entering credentials, click **Logon** without specifying a user ID and password or click **Cancel**. If you complete these steps, the Tivoli Enterprise Monitoring Server is recycled.

   Important: To install support files without recycling the Tivoli Enterprise Monitoring Server, ensure that the Tivoli Enterprise Monitoring Server is running.
5. Select the agent components to generate. You can select **Base Agent**, **Cognos Reporting**, or both.
6. In a IBM Cloud APM environment, you can provide security signing for self-describing agents. Click **Edit all jar signing preferences**. You can add a time stamp to signed JAR files and specify the time stamping authority. Specify details about your Java Keystore File.

   Note: You must create the Java Keystore File by using Java tools. For example, to generate a private key and certificate with a corresponding public key in a Java Keystore File, you can run this command:

   - `ab_install_path/jre/bin/keytool -genkeypair -keystore keystore_file_path -storepass key_store_password -alias key_store_alias -dname "CN=common_name, OU=organizational_unit, L=city_or_locality, ST=state_or_province, C=country" -keypass key_password`

   Where:

   - *ab_install_path* is the location where Agent Builder is installed
   - *keystore_file_path* is the path where an existing JKS keystore is located, or where one is created
   - *key_store_password* is the password that is needed to access any items in this keystore
   - *key_store_alias* is a name that identifies this key within the keystore (defaults to "`mykey`")
   - *key_password* the password that is needed to access this particular key (defaults to `key_store_password`)

   The certificate must be included in the keystore for the server.
7. When you complete the **JAR Signing** details, click **OK**.
8. Click **Finish**.
9. Configure and start the agent. For more information, see "Configuring and starting the agent in an IBM Tivoli Monitoring environment" on page 1359 or "Configuring the agent" on page 1360 and "Starting and stopping the agent" on page 1361 in a IBM Cloud APM environment.

   For Tivoli Monitoring v6.2 FP1 or later, you can install the Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server support without restarting the servers. In this case, the **Restart TEMS without credentials** and **Restart TEPS** check boxes are active in the **Install the Agent Locally** section of the Generate Agent wizard. You can clear the check boxes to install the support without recycling the servers. When you clear the **Restart TEMS without credentials** check box, you are prompted for the Tivoli Enterprise Monitoring Server user ID and password. Enter the Tivoli Enterprise Monitoring Server user ID and password and click **Logon**. If you are running Tivoli Monitoring with security off, enter "sysadmin" for the user ID, leave the password blank, and click **Logon**. You can also continue without entering credentials (click **Logon** without specifying a user ID and password or click **Cancel**. Doing so causes the Tivoli Enterprise Monitoring Server to be recycled).

Note: The Tivoli Enterprise Monitoring Server must be running to install support files without recycling the Tivoli Enterprise Monitoring Server.

Creating the agent package

You can use Agent Builder to create a compressed agent installation package.

About this task

An agent package contains all the fines necessary to run the agent, as well as the installation and configuration scripts. The package also includes support files for the monitoring environment.

You can use an agent package to install the agent into the IBM Tivoli Monitoring and IBM Cloud Application Performance Management environments.

Procedure

1. Click the `itm_toolkit_agent.xml` file from the **Project Explorer** navigation tree of Agent Builder by using one of the following methods:
    - Right-click the `itm_toolkit_agent.xml` file and select **IBM** > **Generate Agent**.
    - Select the `itm_toolkit_agent.xml` file and select the **Generate Agent** icon on the toolbar.
    - Double-click the `itm_toolkit_agent.xml` file and select **Agent Editor** > **Generate Agent**.
2. Enter the name of the directory where you want to place the output (a compressed package or expanded files) in the **Generate Agent Image** section.
3. Select the **Keep intermediate files** check box to keep the generated expanded files separate from the zip or tar file.
4. Select the **Create a ZIP file** check box to create a compressed file in the specified directory. The compressed zip file is named `smai-`*`agent_name`*`-`*`version`*`.zip` for Windows systems by default.
5. Select the **Create a TAR file** check box to create a tar file in the specified directory. The compressed tar file is named `smai-`*`agent_name`*`-`*`version`*`.tgz` for UNIX and Linux systems by default.
6. Select the agent components to generate. You can select **Base Agent**, **Cognos Reporting**, or both.

    Important: For the IBM Cloud Application Performance Management environment, do not select **Cognos Reporting**, because the reports are currently not supported and including the reports increases the size of the package.
7. You can optionally provide security signing for agent application files. If you want to provide security signing, select **Sign self-describing support JAR**. Click **Edit all jar signing preferences**. You can add a timestamp to signed jar files and specify the time stamping authority. Specify details about your Java Keystore File.

    Important: You can create the Java Keystore File by using Java tools. For example, to generate a private key and certificate with a corresponding public key in a Java Keystore File, you can run this command:

    -
    ```
    ab_install_path/jre/bin/keytool -genkeypair -keystore keystore_file_path -
    storepass key_store_password -alias key_store_alias -dname "CN=common_name,
    OU=organizational_unit, L=city_or_locality, ST=state_or_province,
    C=country" -keypass key_password
    ```

    Where:

    - *ab_install_path* is the location where Agent Builder is installed
    - *keystore_file_path* is the path where an existing JKS key store resides, or where one will be created
    - *key_store_password* is the password needed to access any items in this key store
    - *key_store_alias* is a name identifying this key within the key store (defaults to "`mykey`")
    - *key_password* is the password needed to access this particular key (defaults to `key_store_password`)

Include this certificate in the server key store.

8. Click **Finish**.

Installing the package in an IBM Tivoli Monitoring environment

To test or use the agent in the IBM Tivoli Monitoring environment, use the generated package to install the agent on the monitored systems, hub Monitoring Server systems, and Portal Server system.

Before you begin

Before installing the agent on a monitored system, ensure that the Tivoli Monitoring operating system agent is present and working. For information about installing Tivoli Monitoring agents, see Installing monitoring agents in the Tivoli Monitoring Knowledge Center.

Important: To display agent information in the Tivoli Enterprise Portal, you must install the following components:

- The agent on all monitored systems
- Tivoli Enterprise Monitoring Server support files on the hub Tivoli Enterprise Monitoring Servers
- Tivoli Enterprise Portal Server support files on the Tivoli Enterprise Portal Server
- Tivoli Enterprise Portal support files on the Tivoli Enterprise Portal Server and, if applicable, any Tivoli Enterprise Portal desktop clients

Procedure

1. Copy the compressed file, which is named *product_code*.zip for Windows systems or *product_code*.tgz for UNIX and Linux systems by default, onto the system where you want to install the agent.
2. Extract the file to a temporary location.

   Note: Linux AIX For UNIX and Linux systems, this temporary location must not be /tmp/*product_code*, where the product code is lowercase.

   You can install the agent remotely by using the compressed file.

   - Linux On a Linux system, use the following command to extract your .tgz file:

     ```
     tar -xvzf filename
     ```

   - UNIX On an AIX system, use the following command to extract your .tgz file:

     ```
     gunzip filename
     tar -xvf filename
     ```

3. Run the appropriate installation script.

   - To install the agent, Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal support all at the same time:

     ```
     InstallIra.bat/.sh itm_install_location [[-h Hub_TEMS_hostname] -u
     HUB_TEMS_username -p Hub_TEMS_password]
     ```

   - To install the agent without installing support files:

     ```
     installIraAgent.bat/.sh itm_install_location
     ```

   - To install the Tivoli Enterprise Monitoring Server support:

     ```
     installIraAgentTEMS.bat/.sh itm_install_location [[-h Hub_TEMS_hostname] -u
     HUB_TEMS_username -p Hub_TEMS_password]
     ```

   - To install the Tivoli Enterprise Portal Server and Tivoli Enterprise Portal support:

     ```
     installIraAgentTEPS.bat/.sh itm_install_location
     ```

The installation location, *itm_install_location*m must be the first argument and is mandatory on all scripts: `installIra.bat/.sh`, `installIraAgent.bat/.sh`, `installIraAgentTEMS.bat/.sh`, and `installIraAgentTEPS.bat/.sh`. This is the location where Tivoli Monitoring components are installed on this system.

Other arguments are optional.

If you install Monitoring Server support files and do not provide a user ID is not provided, the Tivoli Enterprise Monitoring Server is recycled.

4. Configure and start the agent, see "Configuring and starting the agent in an IBM Tivoli Monitoring environment" on page 1359.

What to do next

If you changed the layout of your agent in a way that causes navigator items to be moved or removed, restart the Tivoli Enterprise Portal Server and Tivoli Enterprise Portal. The restart ensures that your changes are correctly recognized.

Configuring and starting the agent in an IBM Tivoli Monitoring environment

After installing an agent on a monitored system in the IBM Tivoli Monitoring, configure and start the agent.

Procedure

1. Open the **Manage Tivoli Monitoring Service**.

   The new entry **Monitoring Agent for *agent_name*** is displayed.
2. Right-click the entry and select **Configure Using Defaults**. Click **OK** to accept the defaults if you are prompted.

   Important:

   a. On UNIX systems, the option to select is **Configure**.
   b. For multi-instance agents, when you are configuring, you are prompted for an instance name.

   Tip: If your agent uses a JMX data source to connect to a remote WebSphere Application Server, ensure that WebSphere Application Server is also installed on the host that is running the agent and set the `Java` home setting to the Java runtime environment that the local WebSphere Application Server uses.

   Tip: For a Java API, JDBC, JMX, HTTP, or SOAP data source, you can use the **Java** > **JVM arguments** setting to control agent trace logging. Set the following value in this setting:

   ```
   -DJAVA_TRACE_MAX_FILES=files -DJAVA_TRACE_MAX_FILE_SIZE=size
   ```

   where *files* is the maximum number of trace log files that are kept (the default value is 4) and *size* is the maximum log file size in kilobytes (the default value is 5000). For example, you can set the following value:

   ```
   -DJAVA_TRACE_MAX_FILES=7 -DJAVA_TRACE_MAX_FILE_SIZE=100
   ```

   In this case, the agent writes 100 kilobytes into the first log file, then switches to the second log file, and so on. After writing seven log files of 100 kilobytes each, it overwrites the first log file.

   If you added runtime configuration elements to your agent, or if you selected a data source, then you are presented with configuration panels. You use these panels to collect the required information for your agent.
3. Right-click the agent entry and select **Start**
4. Open the Tivoli Enterprise Portal and go to the new agent.

Installing and using an agent in an IBM Cloud Application Performance Management environment
To test or use the agent in the IBM Cloud Application Performance Management environment, use the generated package to install the agent on all monitored systems. In some cases, you need to configure the agent before it can be started. You can start and stop the agent as necessary.

Installing the agent

Use the installation package prepared by Agent Builder to install the agent on all monitored systems.

Before you begin

Ensure that an agent for IBM Cloud Application Performance Management, usually the operating system agent, is already present on the monitored system and working.

Windows On Windows systems, use an Administrator command line shell to install and configure agents. To start an Administrator shell, select **Command Prompt** from the Windows Programs menu, right-click, and click **Run as Administrator**.

Procedure

1. Extract the package to a temporary directory and change to this directory.
2. Install the agent by using the following command, depending on your operating system:
   - Windows On Windows systems, `installIraAgent.bat` *`agent_install_location`*
   - Linux AIX On Linux and UNIX systems, `./installIraAgent.sh` *`agent_install_location`*

   Where *agent_install_location* is the installation location of the existing agent. The default location is:
   - Windows On Windows systems, `C:\IBM\APM`
   - Linux On Linux systems, `/opt/ibm/apm/agent`
   - AIX On AIX systems, `/opt/ibm/apm/agent`

   Important: If you have added any custom configuration properties in the **Runtime Configuration** window of the Agent Editor, if the agent supports multiple instances, or if the agent uses any predefined data source that needs configuration (for example, a user ID and password), you must configure the agent before it can start. If an agent does not require configuration, it starts automatically after installation.

Configuring the agent

If you have added any custom configuration properties in the Runtime Configuration window of the Agent Editor, if the agent supports multiple instances, or if the agent uses any predefined data source that needs configuration (for example, a user ID and password), you must configure the agent before it can start.

Before you begin

Windows On Windows systems, use an Administrator command line shell to install and configure agents. To start an Administrator shell, select **Command Prompt** from the Windows Programs menu, right-click, and click **Run as Administrator**.

About this task

In the configuration process, you can:

- Set the instance name to create or change an instance, if the agent supports multiple instances.
- Set any configuration properties that are available for the agent.
- Create and configure subnodes, if the agent supports subnodes.

Windows On Windows systems, to set any configuration properties or create any subnodes, you must use the silent configuration procedure. A sample silent configuration response file is located in the *install_dir*\samples directory and is named *agentname*_silent_config.txt. Create a copy of this file and set the configuration variables as necessary.

Linux AIX On Linux and UNIX systems, you can optionally use the silent configuration procedure. Alternatively, you can use the interactive procedure. If you start the configuration command without a response file name, the configuration utility prompts you for the configuration values.

Procedure

1. Change to the *install_dir*/bin directory.
2. Run the following command to configure the agent:
   - If the agent does not support multiple instances:
     - Windows On Windows systems, *name*-agent.bat config [*response_file*]
     - Linux AIX On Linux and UNIX systems, ./*name*-agent.sh config [*response_file*]
   - If the agent supports multiple instances:
     - Windows On Windows systems, *name*-agent.bat config *instance_name* [*response_file*]
     - Linux AIX On Linux and UNIX systems, ./*name*-agent.sh config *instance_name* [*response_file*]

   Where:

   - *instance_name* is the name of the instance. If an instance with this name does not exist, the instance is created. If the instance already exists, it is reconfigured. You must create at least one instance to use the agent.
   - *response_file* is the name of the silent configuration response file.

   Tip: If your agent uses a JMX data source to connect to a remote WebSphere Application Server, ensure that WebSphere Application Server is also installed on the host that is running the agent and set the Java home setting to the Java runtime environment that the local WebSphere Application Server uses.

   Tip: For a Java API, JDBC, JMX, HTTP, or SOAP data source, you can use the **Java** > **JVM arguments** setting to control agent trace logging. Set the following value in this setting:

   ```
   -DJAVA_TRACE_MAX_FILES=files -DJAVA_TRACE_MAX_FILE_SIZE=size
   ```

   where *files* is the maximum number of trace log files that are kept (the default value is 4) and *size* is the maximum log file size in kilobytes (the default value is 5000). For example, you can set the following value:

   ```
   -DJAVA_TRACE_MAX_FILES=7 -DJAVA_TRACE_MAX_FILE_SIZE=100
   ```

   In this case, the agent writes 100 kilobytes into the first log file, then switches to the second log file, and so on. After writing seven log files of 100 kilobytes each, it overwrites the first log file.

Starting and stopping the agent

To monitor a system, ensure that the agent is started on the system. You can start and stop the agent at any time. If the agent supports multiple instances, you can start and stop every instance independently.

Procedure

1. Change to the *install_dir*/bin directory.
2. Run the following command to start the agent:
   - If the agent does not support multiple instances:

- Windows On Windows systems, *name*`-agent.bat start`
- Linux AIX On Linux and UNIX systems, `./`*name*`-agent.sh start`

- If the agent supports multiple instances:
  - Windows On Windows systems, *name*`-agent.bat start` *instance_name*
  - Linux AIX On Linux and UNIX systems, `./`*name*`-agent.sh start` *instance_name*

3. Run the following command to stop the agent:
   - If the agent does not support multiple instances:
     - Windows On Windows systems, *name*`-agent.bat stop`
     - Linux AIX On Linux and UNIX systems, `./`*name*`-agent.sh stop`
   - If the agent supports multiple instances:
     - Windows On Windows systems, *name*`-agent.bat stop` *instance_name*
     - Linux AIX On Linux and UNIX systems, `./`*name*`-agent.sh stop` *instance_name*

Agent post-generation and installation results

Installation of an Agent Builder agent creates and changes certain files on your system. In an IBM Tivoli Monitoring environment, you can also see changes in the Tivoli Enterprise Portal.

New files on your system

After you generate and install the agent that you created with Agent Builder, you can see the following new files on your agent system:

Note: *xx* denotes the two character product code.

Windows

Windows systems:

TMAITM6\k*xx*agent.exe
Agent binary

TMAITM6\K*xx*ENV
Environment variable settings

TMAITM6\K*xx*.ref
Agent provider configuration

TMAITM6\SQLLIB\k*xx*.his
SQL description of agent attribute information

TMAITM6\SQLLIB\k*xx*.atr
Agent attribute information

TMAITM6*xx*_dd_*version*.xmll
Product description

TMAITM6*xx*_dd.properties
Product name

TMAITM6\kxxcma.ini
Agent service definition file

TMAITM6*your files*
Supplemental files included from the Java API or Socket data sources with a file type of *executable* or *library*. Scripts included from the Script or Command return code data sources.

Linux AIX

UNIX/Linux systems:

registry/*xxarchitecture*.ver
Internal versions and prerequisites file

***architecture*/*xx*/bin/*xx*_dd_*version*.xml**
Product description

***architecture*/*xx*/bin/k*xx*agent**
Agent binary

***architecture*/*xx*/bin/*xx*_dd.properties**
Product name

***architecture*/*xx*/work/k*xx*.ref**
Agent provider configuration

***architecture*/*xx*/tables/ATTRLIB/k*xx*.atr**
Agent attribute information

***architecture*/*xx*/hist/k*xx*.his**
SQL description of agent attribute information

architecture*/*xx*/bin/*your files
Supplemental files included from the Java API or Socket data sources with a file type of *executable*. Scripts included from the Script or Command return code data sources.

architecture*/*xx*/lib/*your files
Supplemental files included from the Java API or Socket data sources with a file type of Library.

config/.*xx*.rc
Internal setup file

config/*xx*.environment
Environment settings

config/*xx*_dd_*version*.xml
Product description

config/*xx*_dd.properties
Product name

config/.ConfigData/k*xx*env
Environment variable settings

Note: Run the following command to find out the architecture of the system:

```
cinfo -pxx
```

where *xx* is the two-character product code.

For example, for a Solaris 8 64-bit system that is running an agent with product code 19, here is the output:

```
# /opt/ibm/apm/agent/bin/cinfo -p 19

*********** Fri Aug 17 11:23:58 EDT 2007 ******************
User : root Group: other
Host name : guadalajara  Installer Lvl:06.20.00.00
CandleHome: /opt/IBM/ITM
***********************************************************
Platform codes:
sol286 : Current machine
sol286 : Product (19)
tmaitm6/sol286 : CT Framework (ax)
```

The line in bold is the relevant one. The string before the colon, `sol286`, indicates the architecture in use for this agent. This string is different for different combinations of operating system and computer hardware type. The agent must be previously installed for this feature to work.

The following files are for Java-based data sources. These files are created only if the agent contains JMX, JDBC, HTTP, or SOAP data sources:

- `cpci.jar`
- `jlog.jar`
- `common/jatlib-1.0.jar`

The following files are for JMX runtime support. These files are created only if the agent contains JMX data sources:

- `common/jmx-1.0.jar`
- `common/connectors/jboss/connJboss-1.0.jar`
- `common/connectors/jsr160/connJSR160-1.0.jar`
- `common/connectors/was/connWas-1.0.jar`
- `common/connectors/weblogic/connWeblogic-1.0.jar`

The following file is for JDBC runtime support. These files are created only if the agent contains JDBC data sources:

- `common/jdbc-1.0.jar`

The following file is for HTTP or SOAP runtime support. These files are created only if the agent contains HTTP or SOAP data sources:

- `http-1.0.jar`

The following files are for the Java API runtime support. These files are created only if the agent contains a Java API data source:

- `cpci.jar`
- `custom/`*`your JAR file`* The name of this JAR file is specified in the **Global settings** of a Java API data source.
- `custom/`*`your JAR file`* Supplemental files with a file type of Java resource.

The same files exist on Windows, UNIX, and Linux systems for Java-based data sources, but they are in different directories:

- Windows Windows path: `TMAITM6\kxx\jars`
- Linux AIX UNIX/Linux path: *`architecture`*`/`*`xx`*`/jars`

The following files are for log file monitoring runtime support. These files are created only if the agent contains log file data sources:

- Windows On Windows systems: TMAITM6\kxxudp.dll
- Linux On Solaris/Linux systems: *architecture*/*xx*/lib/libkxxudp.so
- On HP-UX systems: *architecture*/*xx*/lib/libkxxudp.sl
- UNIX On AIX systems: *architecture*/*xx*/lib/libkxxudp.a

The following files are for SSH script monitoring runtime support. These files are created only if the agent contains a script data source that is enabled for SSH collection:

- Windows On Windows systems: `TMAITM6\kxxssh.dll`
- Linux On Solaris/Linux systems: *`architecture`*`/`*`xx`*`/lib/libkxxssh.so`
- On HP-UX systems: *`architecture`*`/`*`xx`*`/lib/libkxxssh.sl`
- UNIX On AIX systems: *`architecture`*`/`*`xx`*`/lib/libkxxssh.a`

Changes in the Manage Tivoli Enterprise Monitoring Services window

After installing an agent in a IBM Tivoli Monitoring environment, you can see an entry for the agent in the **Manage Tivoli Enterprise Monitoring Services** window. The entry name is **Monitoring Agent for *agent_name***.

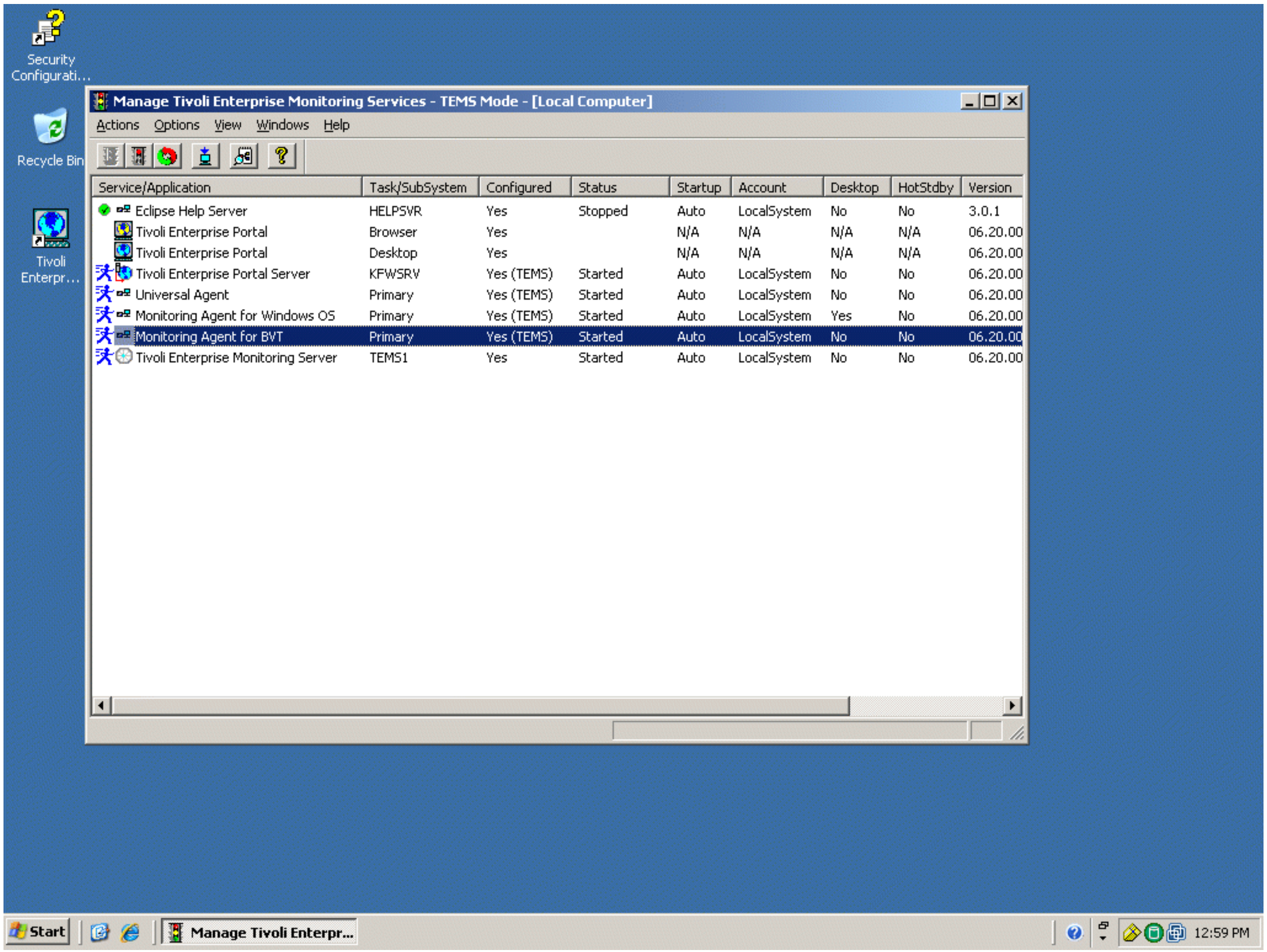
Important: Manage Tivoli Enterprise Monitoring Services is not supported in the IBM Cloud Application Performance Management environment.

Windows On Windows systems, this entry contains a **Task/Subsystem** column that identifies whether your agent supports multiple instances:

- A single instance agent displays a new application in the **Manage Tivoli Enterprise Monitoring Services** window. The name of the application is **Monitoring Agent for *agent_name***. A service is created for the agent (Figure 70 on page 1365). The **Task/Subsystem** column contains the value **Primary**.
- A multiple instance agent displays a new application template in the **Manage Tivoli Enterprise Monitoring Services** window. The name of the template is **Monitoring Agent for *agent_name***. A service is not created for the agent until you create an instance of the agent from this template. The **Task/Subsystem** column contains the value **Template** to indicate that this entry is a template that is used to create instances of the agent.

Linux AIX On Linux and UNIX systems, the entry for the agent is the same whether your agent supports multiple instances or not.

Note: The following screens are for a Windows system. UNIX and Linux systems have similar screens.



Service/Application	Task/SubSystem	Configured	Status	Startup	Account	Desktop	HotStdby	Version
Eclipse Help Server	HELPSVR	Yes	Stopped	Auto	LocalSystem	No	No	3.0.1
Tivoli Enterprise Portal	Browser	Yes		N/A	N/A	N/A	N/A	06.20.00
Tivoli Enterprise Portal	Desktop	Yes		N/A	N/A	N/A	N/A	06.20.00
Tivoli Enterprise Portal Server	KFWSRV	Yes (TEMS)	Started	Auto	LocalSystem	No	No	06.20.00
Universal Agent	Primary	Yes (TEMS)	Started	Auto	LocalSystem	No	No	06.20.00
Monitoring Agent for Windows OS	Primary	Yes (TEMS)	Started	Auto	LocalSystem	Yes	No	06.20.00
Monitoring Agent for BVT	Primary	Yes (TEMS)	Started	Auto	LocalSystem	No	No	06.20.00
Tivoli Enterprise Monitoring Server	TEMS1	Yes	Started	Auto	LocalSystem	No	No	06.20.00

Figure 70. Manage Tivoli Enterprise Monitoring Services window

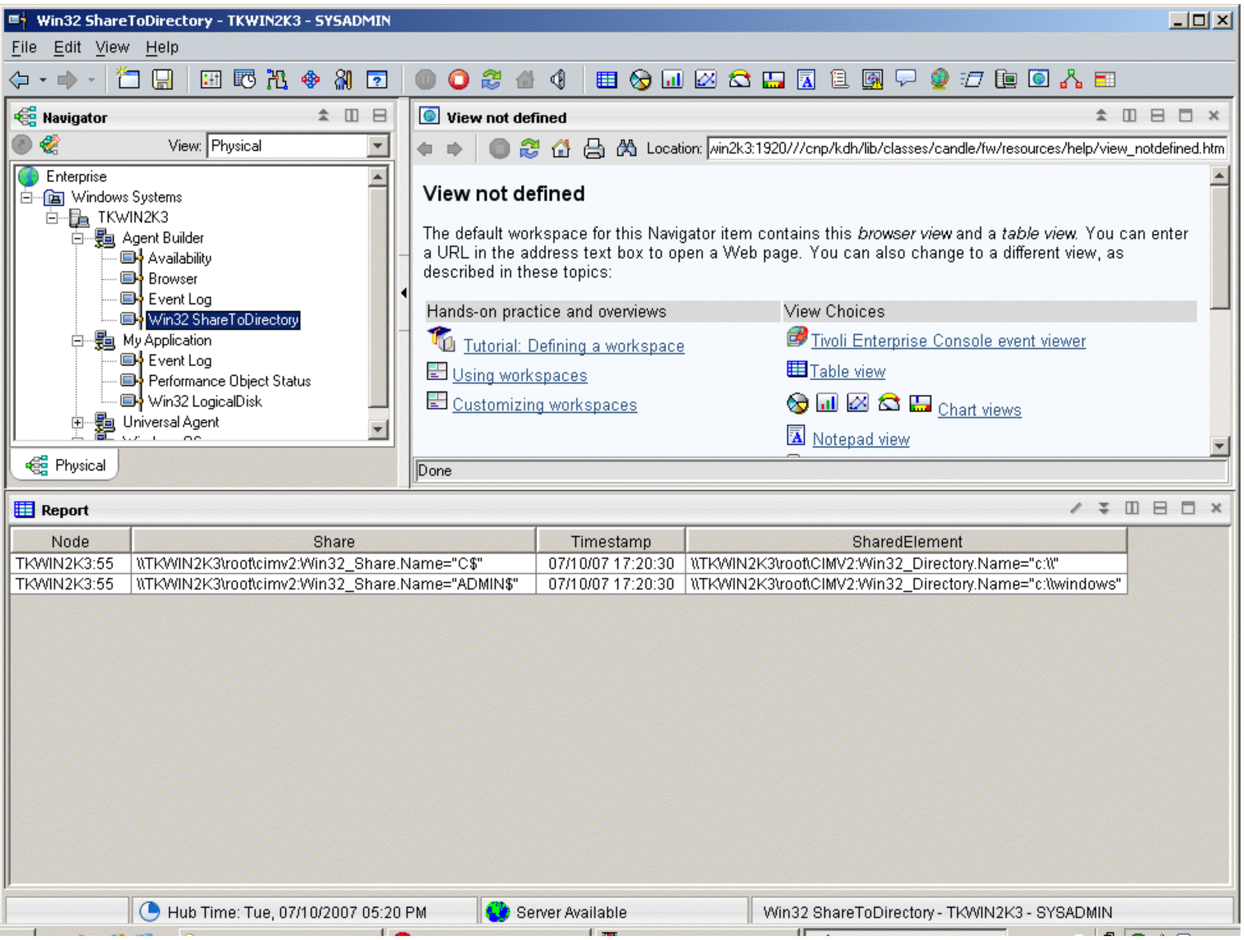
Changes in the Tivoli Enterprise Portal

In an IBM Tivoli Monitoring environment, after you install and start the agent, click the green **Refresh** icon in Tivoli Enterprise Portal. Then you can view the new agent. You can see the following changes in the portal:

- A new subnode for the agent in the Tivoli Enterprise Portal physical view.

- Nodes for every navigator group and data source that you defined by using the Agent Builder (Figure 71 on page 1366).

  Note: For each navigator item, you must define a default query.



View not defined

The default workspace for this Navigator item contains this *browser view* and a *table view*. You can enter a URL in the address text box to open a Web page. You can also change to a different view, as described in these topics:

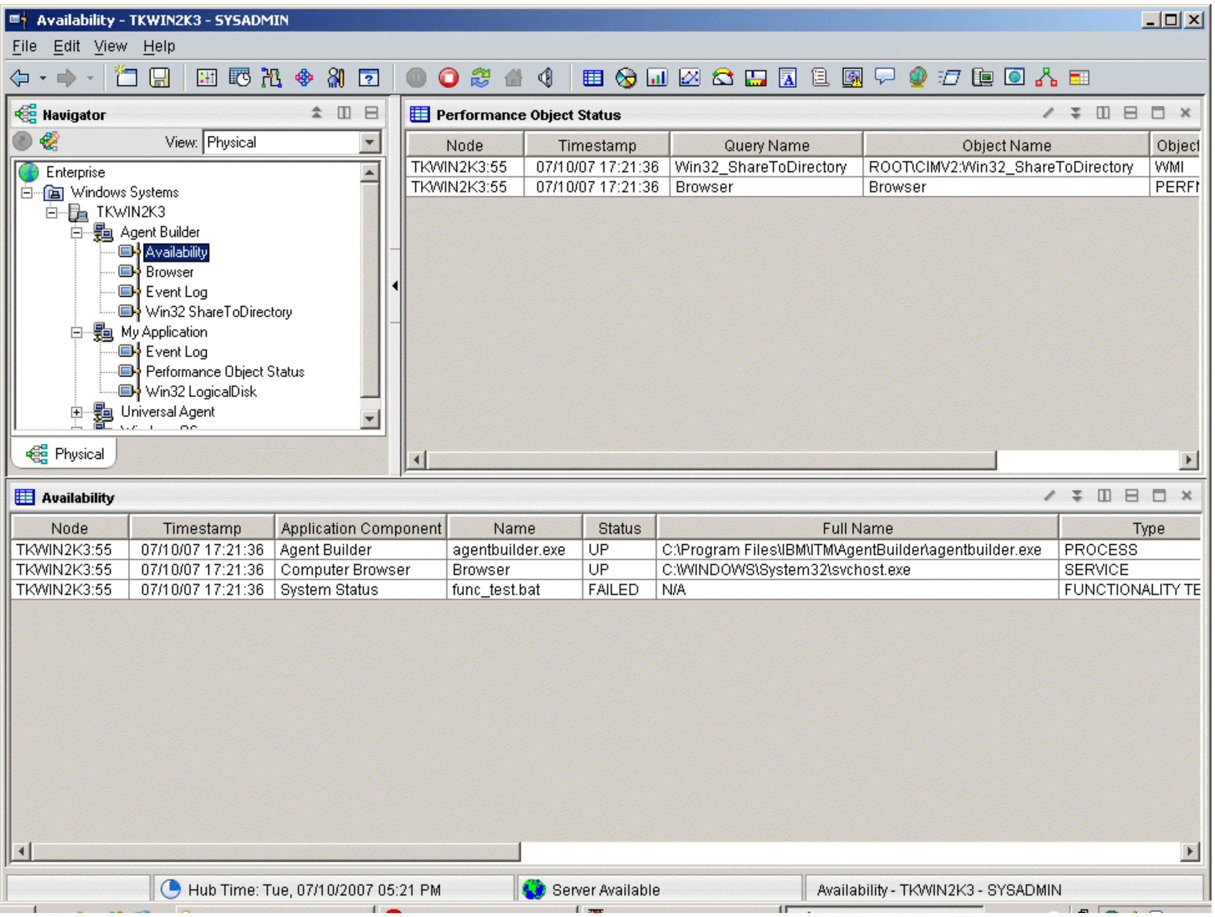
Hands-on practice and overviews	View Choices
Tutorial: Defining a workspace	Tivoli Enterprise Console event viewer
Using workspaces	Table view
Customizing workspaces	Chart views
	Notepad view

Node	Share	Timestamp	SharedElement
TKWIN2K3:55	\\TKWIN2K3\root\cimv2:Win32_Share.Name="C$"	07/10/07 17:20:30	\\TKWIN2K3\root\CIMV2:Win32_Directory.Name="c:\\"
TKWIN2K3:55	\\TKWIN2K3\root\cimv2:Win32_Share.Name="ADMIN$"	07/10/07 17:20:30	\\TKWIN2K3\root\CIMV2:Win32_Directory.Name="c:\\windows"

Figure 71. Nodes for attribute groups in the new agent.

- If your agent contains subnodes, an expandable node is present for each subnode that is defined in your agent. The following nodes are shown under the expandable node:
  - xxx performance object status, where xxx is the three-letter subnode type
  - Nodes for every Navigator group and data source that you defined in the subnode
  - xxx event log node if you have event logs
  - xxx JMX monitors node if you have JMX and you included JMX monitors
- The following automatic node:
  - An availability node if your agent contains an availability data source (Figure 72 on page 1367)

    Note: This node behaves differently depending on the contents of the agent. If the agent monitors only availability, the availability node represents the availability data source. If the agent monitors availability and performance, the availability node becomes the navigator item that represents the availability and performance object status data sources.



Performance Object Status

Node	Timestamp	Query Name	Object Name	Object
TKWIN2K3:55	07/10/07 17:21:36	Win32_ShareToDirectory	ROOT\CIMV2:Win32_ShareToDirectory	WMI
TKWIN2K3:55	07/10/07 17:21:36	Browser	Browser	PERFI

Availability

Node	Timestamp	Application Component	Name	Status	Full Name	Type
TKWIN2K3:55	07/10/07 17:21:36	Agent Builder	agentbuilder.exe	UP	C:\Program Files\IBM\ITM\AgentBuilder\agentbuilder.exe	PROCESS
TKWIN2K3:55	07/10/07 17:21:36	Computer Browser	Browser	UP	C:\WINDOWS\System32\svchost.exe	SERVICE
TKWIN2K3:55	07/10/07 17:21:36	System Status	func_test.bat	FAILED	N/A	FUNCTIONALITY TE

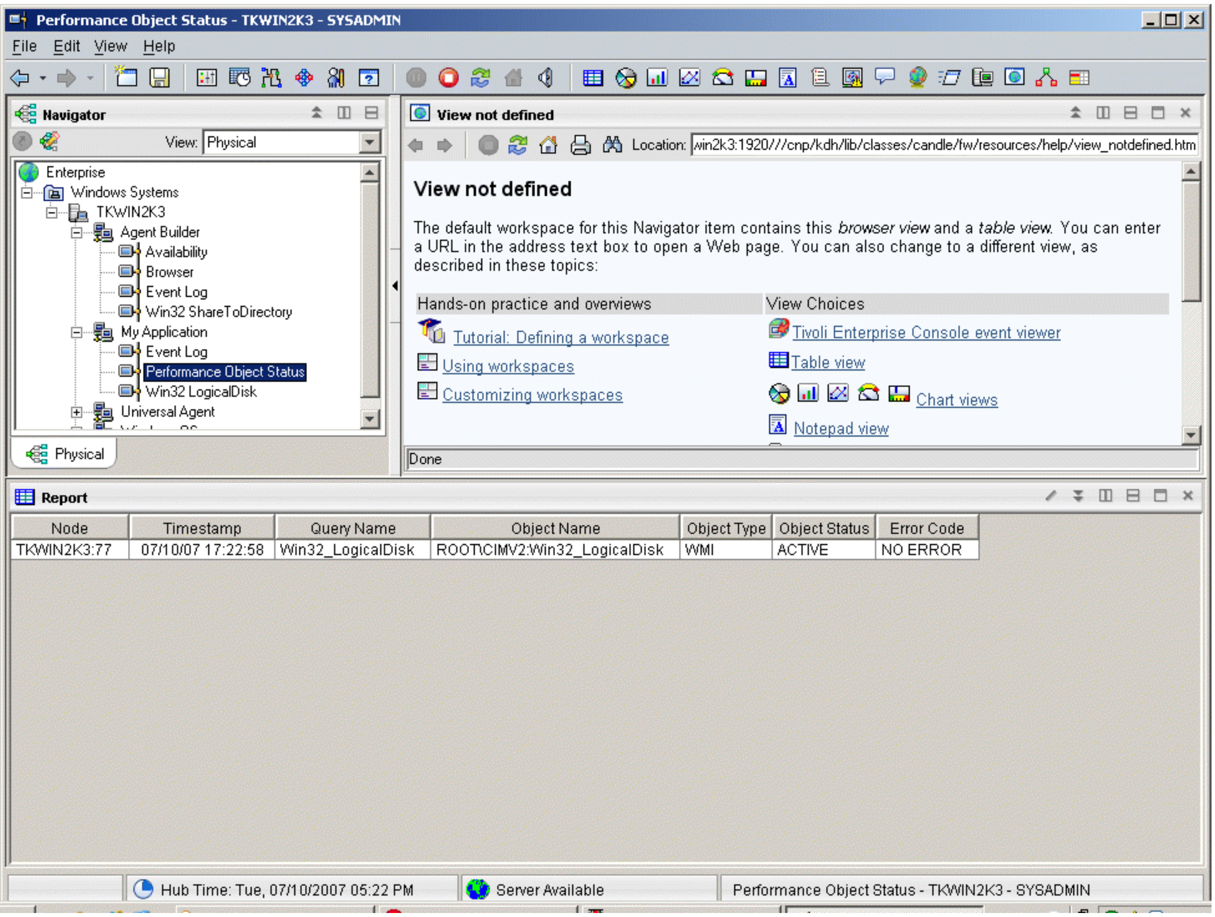
Hub Time: Tue, 07/10/2007 05:21 PM

Server Available

Availability - TKWIN2K3 - SYSADMIN

Figure 72. Availability node

– Performance Object Status, if the agent includes performance monitoring (not availability) data sources (Figure 73 on page 1368)


Performance Object Status - TKWIN2K3 - SYSADMIN
File
Edit
View
Help
Navigator
View: Physical
Enterprise
Windows Systems
TKWIN2K3
Agent Builder
Availability
Browser
Event Log
Win32 ShareToDirectory
My Application
Event Log
Performance Object Status
Win32 LogicalDisk
Universal Agent
Physical
View not defined
Location: win2k3:1920///cnp/kdh/lib/classes/candle/fw/resources/help/view_notdefined.htm
Done
Report
Hub Time: Tue, 07/10/2007 05:22 PM
Server Available
Performance Object Status - TKWIN2K3 - SYSADMIN

View not defined

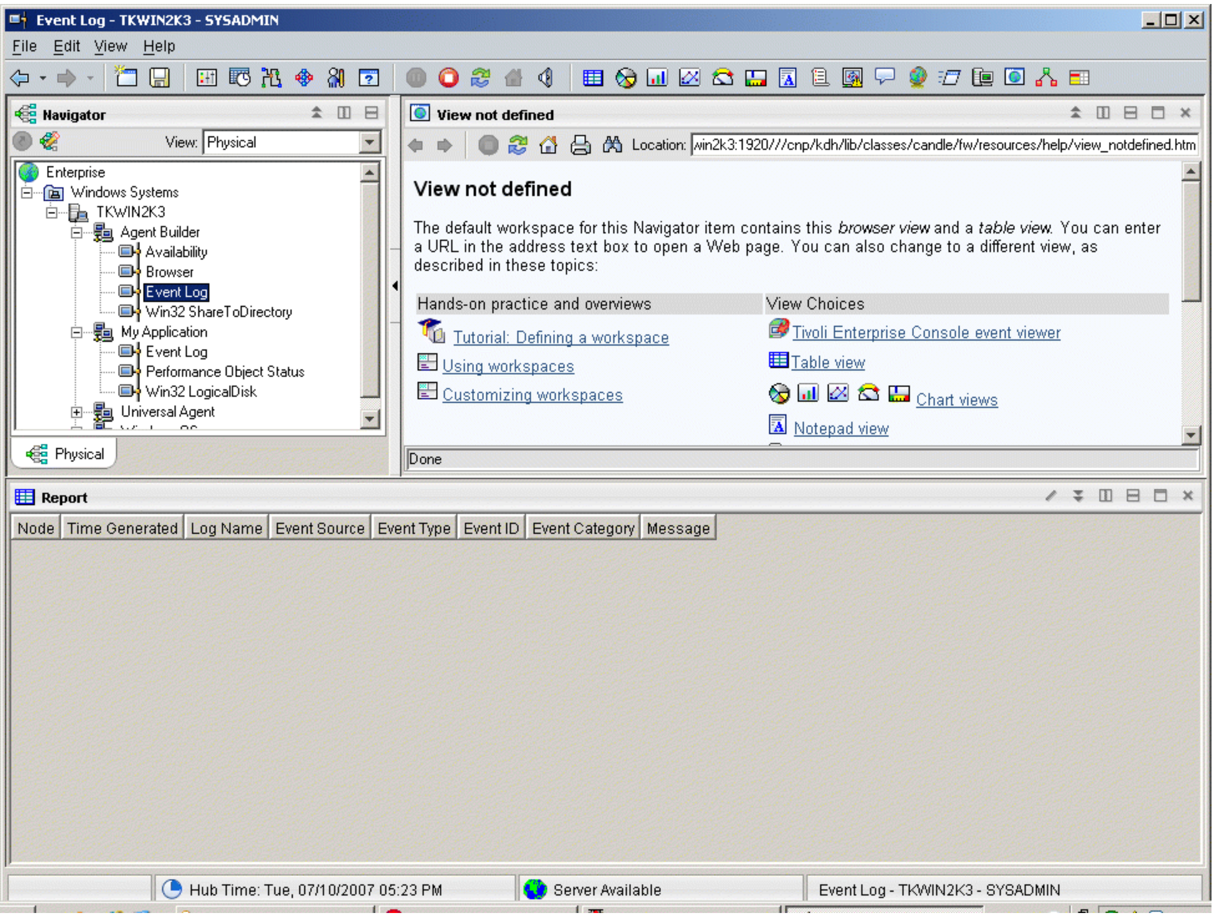
The default workspace for this Navigator item contains this *browser view* and a *table view*. You can enter a URL in the address text box to open a Web page. You can also change to a different view, as described in these topics:

Hands-on practice and overviews	View Choices
Tutorial: Defining a workspace	Tivoli Enterprise Console event viewer
Using workspaces	Table view
Customizing workspaces	Chart views
	Notepad view

Node	Timestamp	Query Name	Object Name	Object Type	Object Status	Error Code
TKWIN2K3:77	07/10/07 17:22:58	Win32_LogicalDisk	ROOT\CIMV2:Win32_LogicalDisk	WMI	ACTIVE	NO ERROR

Figure 73. Performance Object Status node

– Event log, if the agent contains data sources producing log data (Figure 74 on page 1369)



Location: win2k3:1920///cnp/kdh/lib/classes/candle/fw/resources/help/view_notdefined.htm

View not defined

The default workspace for this Navigator item contains this *browser view* and a *table view*. You can enter a URL in the address text box to open a Web page. You can also change to a different view, as described in these topics:

Node	Time Generated	Log Name	Event Source	Event Type	Event ID	Event Category	Message

Hub Time: Tue, 07/10/2007 05:23 PM

Event Log - TKWIN2K3 - SYSADMIN

Figure 74. Event log node

See "Attributes reference" on page 1383 for descriptions of the attribute groups and attributes for Agent Builder.

Uninstalling an agent

You can remove an agent that the Agent Builder generated from a monitored host.

About this task

The uninstallation process uninstalls only the agent from the agent system. This process does not uninstall any other agent or any monitoring infrastructure.

In an IBM Tivoli Monitoring environment, you can use one of the following procedures to remove an agent that the Agent Builder generated:

- "Removing a Tivoli Monitoring agent by using the Tivoli Enterprise Portal" on page 1369
- "Removing a Tivoli Monitoring agent without using the Tivoli Enterprise Portal" on page 1370

After removing the agent using any of these procedures, clear it from the Tivoli Enterprise Portal using the following procedure: "Clearing a Tivoli Monitoring agent from the Tivoli Enterprise Portal" on page 1370.

In an IBM Cloud Application Performance Management environment, use the following procedure: "Uninstalling an IBM Cloud Application Performance Management agent" on page 1371.

Removing a Tivoli Monitoring agent by using the Tivoli Enterprise Portal

In an IBM Tivoli Monitoring environment, you can use the Tivoli Enterprise Portal to remove an agent.

Before you begin

Your operating system agent must be running in order to remove your created agent.

Procedure

To use the Tivoli Enterprise Portal to remove an agent, complete the following step:

- In the Tivoli Enterprise Portal navigation tree, right-click the agent and select **Remove**.

Removing a Tivoli Monitoring agent without using the Tivoli Enterprise Portal

If a Tivoli Enterprise Portal is not available in your IBM Tivoli Monitoring environment, you can use operating system scripts and commands to remove an agent.

Procedure

To remove an agent that the Agent Builder generated from the target system without using a Tivoli Enterprise Portal, you can complete any of the following steps:

- Windows

  On Windows systems, use the commands:

  ```
  cd ITM_INSTALL/TMAITM6
  kxx_uninstall.vbs ITM_INSTALL
  ```

  where *xx* is the product code for the agent

- Windows

  Alternatively. on Windows systems, you can use the `cscript.exe` command to run the uninstallation script. This command is the command-line interface parser for vbs scripts and does not display a window; instead, a message is displayed on the console:

  ```
  cd ITM_INSTALL/TMAITM6
  cscript.exe kxx_uninstall.vbs ITM_INSTALL
  ```

- Linux AIX

  On Linux or UNIX systems. use the `uninstall.sh` file that is found in *`ITM_INSTALL`*`/bin`:

  ```
  uninstall.sh [-f] [-i] [-h ITM_INSTALL] [product platformCode]
  ```

Clearing a Tivoli Monitoring agent from the Tivoli Enterprise Portal

In an IBM Tivoli Monitoring environment, after you remove the agent, empty fields for information from the agent can remain in the Tivoli Enterprise Portal. To remove the fields, clear the agent from the Tivoli Enterprise Portal.

Procedure

1. Ensure that your Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server are up and running.
2. Log on to your Tivoli Enterprise Portal client.
3. From the Tivoli Enterprise Portal client Physical Navigator view, right-click **Enterprise** and select **Workspace** > **Managed System Status**.

   The Managed System Status workspace is displayed.
4. Select all of the IBM Tivoli Managed Systems for your agent.
5. Right-click and select **Clear off-line entry**, which clears all of the entries from that table.

Uninstalling an IBM Cloud Application Performance Management agent
You can uninstall your agent from any monitored system in an IBM Cloud Application Performance Management environment.

Procedure

1. On the system where the agent is installed, start a command line and change to the *install_dir*/bin directory, where *install_dir* the installation directory of the monitoring agents.
2. To uninstall a specific monitoring agent, enter the agent script name and the uninstall option where *name* is the agent script name:
   - On Windows systems, *name*-agent.bat uninstall
   - On Linux or AIX systems, ./*name*-agent.sh uninstall

Importing application support files

If an agent is to be used in an IBM Tivoli Monitoring environment, custom situations, workspaces, Take Action commands, and queries can be included in the installation package.

About this task

To have a single installation image for situations, workspaces, and the agent, the situation, and workspace files must be in the same project as the agent. The Agent Builder provides a wizard to create the appropriate files in the agent project.

Definitions that are associated with an agent can also be included in the installation package. The content of these definitions is different for an agent that is used in an enterprise monitoring environment and in a system monitor environment. An enterprise monitoring agent image can include custom situations, workspaces, Take Action commands and queries. A system monitor agent image can include private situations, trap definitions, and agent configuration information.

To have a single installation package that includes the appropriate definitions and the agent itself, the files must be in the same project as the agent. The Agent Builder provides a wizard to create the appropriate files for an enterprise monitoring installation. The files for a system monitor agent environment are created by using the process that is described in the *Agent Autonomy* chapter in the *IBM Tivoli Monitoring Administrator's Guide*. The resulting files are copied into the root of the Eclipse project for the agent.

Exporting and importing files for Tivoli Enterprise Monitoring Agents

About this task

After you create situations, workspaces, queries, and Take Action commands in the Tivoli Enterprise Portal, you can export and import them into another Tivoli Monitoring Version 6.2 environment. For more information about creating situations and workspaces, see ("Creating workspaces, Take Action commands, and situations" on page 1337). Use the following steps to extract the situations, workspaces, Take Action commands, and queries:

Procedure

1. From the **Project Explorer** tab, right-click the agent project folder.
2. Select **IBM Corporation** > **Import Application Support Files**.
3. Enter the host name of the Tivoli Enterprise Portal Server.
4. Enter the user name and password for the Tivoli Monitoring environment you are connecting to and click **Finish**.
5. If you defined situations for your agent, a dialog box is presented that lists the situations that are defined for the agent.

6. Select the situations that you want to export from the list and click **<<** to add them to the selected situations table and click **OK**.

   The import might take a few moments. When the task completes, you see the SQL files in the appropriate folders in the agent project.
7. If you defined Take Action commands for your agent, a dialog presents the Take Action commands defined. Choose the Take Action commands that you want to export from the list and click **>>** to add them to the Selected Take Actions table and click **OK**.

   The import might take a few moments. When the task completes, you see the SQL files in the appropriate folders in the agent project.
8. If you defined custom queries for your agent, a dialog presents the Queries defined. Select the queries that you want to export from the list and click **<<** to add them to the Selected Queries table and click **OK**.

   The import might take a few moments. When the task completes, you see the SQL files in the appropriate folders in the agent project. Workspaces are imported automatically.

What to do next

Re-create your custom agent, install your agent on the monitored host, and install the Tivoli Enterprise Portal support.

Exporting and importing files for Tivoli System Monitor Agents

About this task

The system monitor agent definitions are contained in three types of files:

- Private situations are defined in a file named `*xx*_situations.xml`, where *xx* is the two-character product code
- Trap configuration information is defined in a file named `*xx*_trapcnfg.xml`, where *xx* is the two-character product code
- For agents that require configuration, the configuration is defined in one file for each instance of the agent. When the agent is a single instance agent, the file is named `*xx*.cfg`. When the agent is a multi-instance agent, there is a file is present for each instance. The file names are `*xx*_*instance name*.cfg`, where *xx* is the two-character product code and *instance name* is the name of the agent instance.

Procedure

- Create the files by using the process that is described in the *Agent Autonomy* chapter in the *IBM Tivoli Monitoring Administrator's Guide*. Copy the files into the root of the project directory manually, or use the Eclipse import function to select the files to be imported: **File** > **Import** > **General** > **File System**.

  These files are included in the agent image and installed by the installer.

  When the agent is installed the installation:

  – Copies the included files into the appropriate locations.
  – Any private situations that are defined in the `pc_situations.xml` file that is run on the agent.
  – The trap definitions that are defined in the `pc_trapcnfg.xml` are used to forward traps that are based on the situations.
  – The agent is automatically configured and started if:
    - The agent is a single instance agent with no configuration defined as part of the agent.
    - The agent is a single instance agent with configuration defined as part of the agent and the image includes a `pc.cfg`file.
    - The agent is a multi-instance agent (all multi-instance agents require configuration): the installer starts one instance of the agent for each `pc_*inst*.cfg` file.

Event filtering and summarization

An attribute group is defined to be *pure event* or *sampled*. Pure event attribute groups contain data rows that occur asynchronously. As each new row of data arrives, it is processed immediately by Tivoli Monitoring. Sampled attribute groups collect the current set of data rows each time the data is requested. The following attribute groups illustrate the difference:

- An `SNMPEvent` attribute group is created that represents all of the SNMP Traps and informs that are sent to the agent. Traps or informs arrive asynchronously as they are sent by the monitored systems. As each event arrives, it is passed to Tivoli Monitoring.
- A `Disk` attribute group is created to represent information about all of the disks on a system. The disk information is collected periodically. Each time disk information is collected, the agent returns a number of rows of data, one for each disk.

The difference between pure event and sampled attribute groups affects various aspects of Tivoli Monitoring. These aspects include: situations, warehouse data, and Tivoli Enterprise Portal views.

Each situation is assigned (or *distributed*) to one or more managed systems to be monitored for a specific condition of a set of conditions. When the determination of the event must be made based on observations that are made at specific intervals, the event is known as a *sampled event*. When the event is based on a spontaneous occurrence, the event is known as a *pure event*. Therefore, situations for sampled events have an interval that is associated with them, while situations for pure events do not. Another characteristic of sampled events is that the condition that caused the event can change, thus causing it to be no longer true. Pure events cannot change. Therefore, alerts that are raised for sampled events can change from true to false, while a pure event stays true when it occurs.

An example of a sampled event is `number of processes > 100`. An event becomes true when the number of processes exceeds 100 and later becomes false again when this count drops to 100 or less. A situation that monitors for `invalid logon attempt by user` is a pure event; the event occurs when an invalid logon attempt is detected, and does not become a False event. While you can create situations that are evaluated on a specific interval for sampled attribute groups, such evaluations are not possible for pure event attribute groups.

Similarly, for historical data, you can configure how frequently sampled data is collected. However, when you turn collection on for pure event data, you get each row as it happens.

The data that is displayed in the Tivoli Enterprise Portal for sampled data is the latest set of collected rows. The data that is displayed for pure event attribute groups is the contents of a local cache that is maintained by the agent. It does not necessarily match the data that is passed to Tivoli Monitoring for situation evaluation and historical collection.

Controlling duplicate events

Use the event filtering and summarization options to control how duplicate events are sent to Tivoli Monitoring.

Before you begin

For more information about event filtering and summarization, see “Event filtering and summarization” on page 1373.

About this task

The Agent Builder defines attribute groups that represent event data as *pure event* in Tivoli Monitoring. These attribute groups include log file, AIX Binary Log, SNMP events, and JMX notifications. These attribute groups can produce multiple duplicate events. You can control how these duplicate events are sent to Tivoli Monitoring. You can activate these controls for log file, SNMP events, and JMX notifications attribute groups in the **Event Information** tab under **Advanced Data Source Properties** in the **Advanced** window.

Whether an event is treated as a duplicate of other events is determined by the key attributes, you define in the attribute group. A duplicate event occurs when the values for all key attributes in the event match the values for the same key attributes in an existing event. When event filtering and summarization is enabled, the attributes for the `isSummary`, `occurrenceCount`, `summaryInterval`, and `eventThreshold` functions are added automatically.

Procedure

- In the **Event Filtering and Summarization Options** area, select one of the following options:
  - **No event filtering or summarization**: Sends all events without any event filtering or summarization. This option is the default option.
  - **Filter and summarize events**: Creates a summary record for each event with duplicates and each unique event that is based on the key attributes. Select also to choose the event filtering option. In the **Summarization Options** area, enter the summary interval. You can enter either a value in seconds or insert a configuration property.

    The event filtering options are:

    - **Only send summary events**: Sends only the summary records for the specified interval.
    - **Send all events**: Sends all events and summary records.
    - **Send first event**: For each event, sends only the first event that is received in the summary interval that is specified and no duplicate events. This option also sends the summary records.
    - **Event threshold**: Sends an event to Tivoli Monitoring when the number of duplicate events that are received in the interval is evenly divisible by the threshold. For example, if you set the event threshold to 5 and you receive less than five duplicates (including the first event) in the interval, no event is sent to Tivoli Monitoring. If you receive 5, 6, 7, 8, or 9 duplicates, one event is sent. If you receive 10 duplicates, 2 events are sent. In the **Event threshold** field, you can enter a number or insert a configuration property. This option also sends the summary records.

Viewing event filtering and summarization in the Tivoli Enterprise Portal

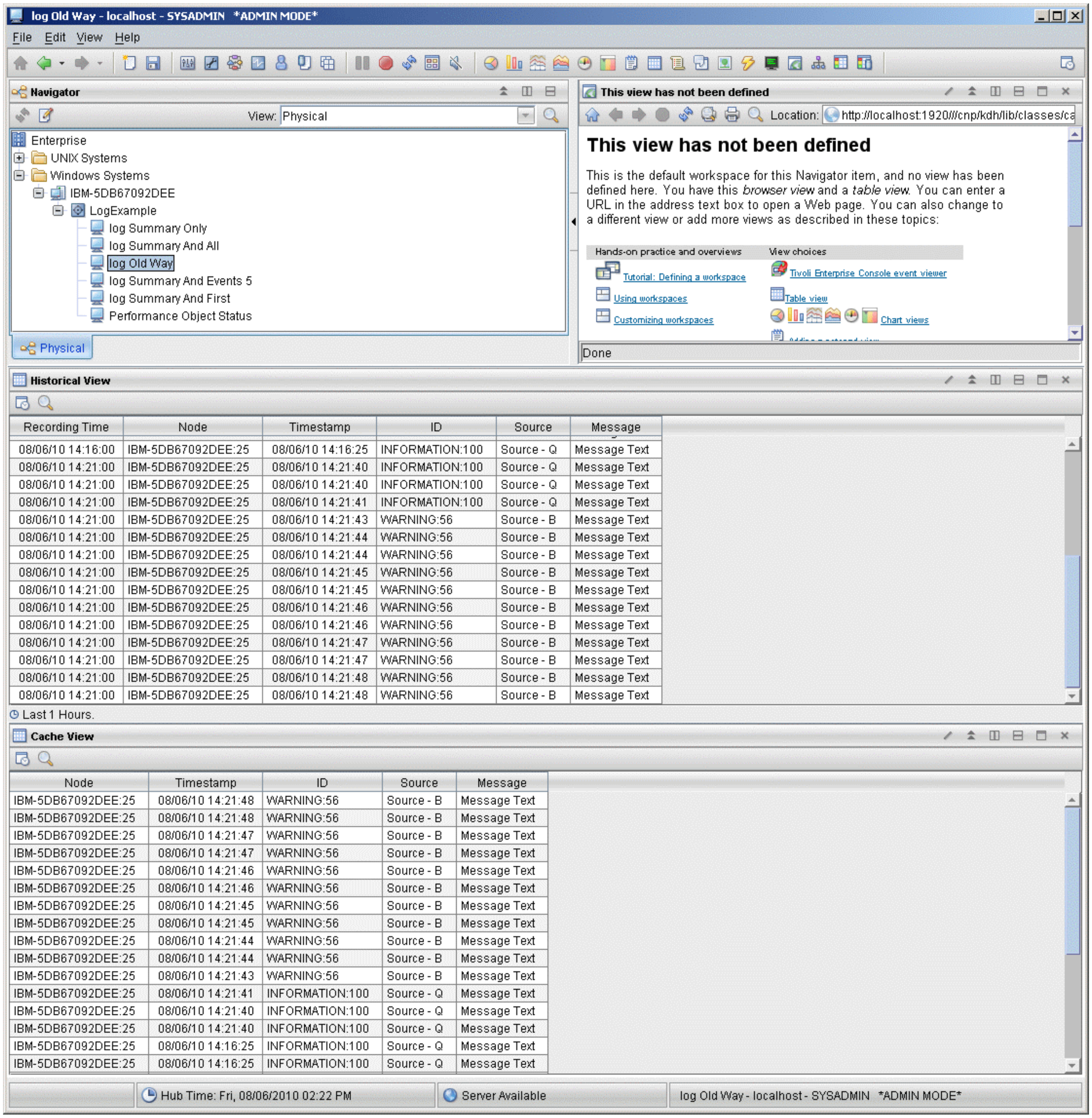
Examples of how data is treated depending on your event filtering and summarization choices.

The agent maintains a cache of the last events received. By default, this cache is 100 in size. If you enable agent event filtering and summarization, differences can occur between the number of events in the cache and the number sent to IBM Tivoli Monitoring. Additional events in the cache might not reach the designated threshold for sending. Or you might have fewer events in the cache if you selected the **Send all events** option. If the **Send all events** option is set, an event is sent each time a duplicate occurs. However, only one copy of the event is kept in the cache, and the occurrence count is incremented each time that the event occurs. To view the events that are sent to IBM Tivoli Monitoring, create a historical view. For information about creating historical views, see *Historical Reporting* in the Tivoli Enterprise Portal User's Guide. You can compare this view with the real-time cache view in the Tivoli Enterprise Portal. You can also use situations to make the same comparison.

The following examples indicate how the same log data is treated depending on your choice, if any, of event filtering and summarization. The example agent was created to illustrate different behaviors. Each attribute group was defined to monitor the same log file. In each example, a historical view and a real-time (cache) view is shown. The names of the nodes in the Tivoli Enterprise Portal reflect the settings selected. By default, the historical view displays the newest events last. The default real-time view of the cache displays the newest events first. In these examples, the historical view shows the last 1 hour.

As new events arrive, you can see them in the cache view. As duplicates of an event arrive, the data is updated in the existing row. When a summary interval elapses, the existing events are converted to summary events and sent. New rows are then added for the next summary interval.

(Figure 75 on page 1375) shows the historical view and cache view if you did not enable event filtering or summarization. Both views display the same data, but in reverse order. To display the corresponding events, the historical view is scrolled down and the real time (cache) view is scrolled up.



This view has not been defined

This is the default workspace for this Navigator item, and no view has been defined here. You have this *browser view* and a *table view*. You can enter a URL in the address text box to open a Web page. You can also change to a different view or add more views as described in these topics:

Historical View

Recording Time	Node	Timestamp	ID	Source	Message
08/06/10 14:16:00	IBM-5DB67092DEE:25	08/06/10 14:16:25	INFORMATION:100	Source - Q	Message Text
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:40	INFORMATION:100	Source - Q	Message Text
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:40	INFORMATION:100	Source - Q	Message Text
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:41	INFORMATION:100	Source - Q	Message Text
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:43	WARNING:56	Source - B	Message Text
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:44	WARNING:56	Source - B	Message Text
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:44	WARNING:56	Source - B	Message Text
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:45	WARNING:56	Source - B	Message Text
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:45	WARNING:56	Source - B	Message Text
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:46	WARNING:56	Source - B	Message Text
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:46	WARNING:56	Source - B	Message Text
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:47	WARNING:56	Source - B	Message Text
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:47	WARNING:56	Source - B	Message Text
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:48	WARNING:56	Source - B	Message Text
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:48	WARNING:56	Source - B	Message Text

Last 1 Hours.

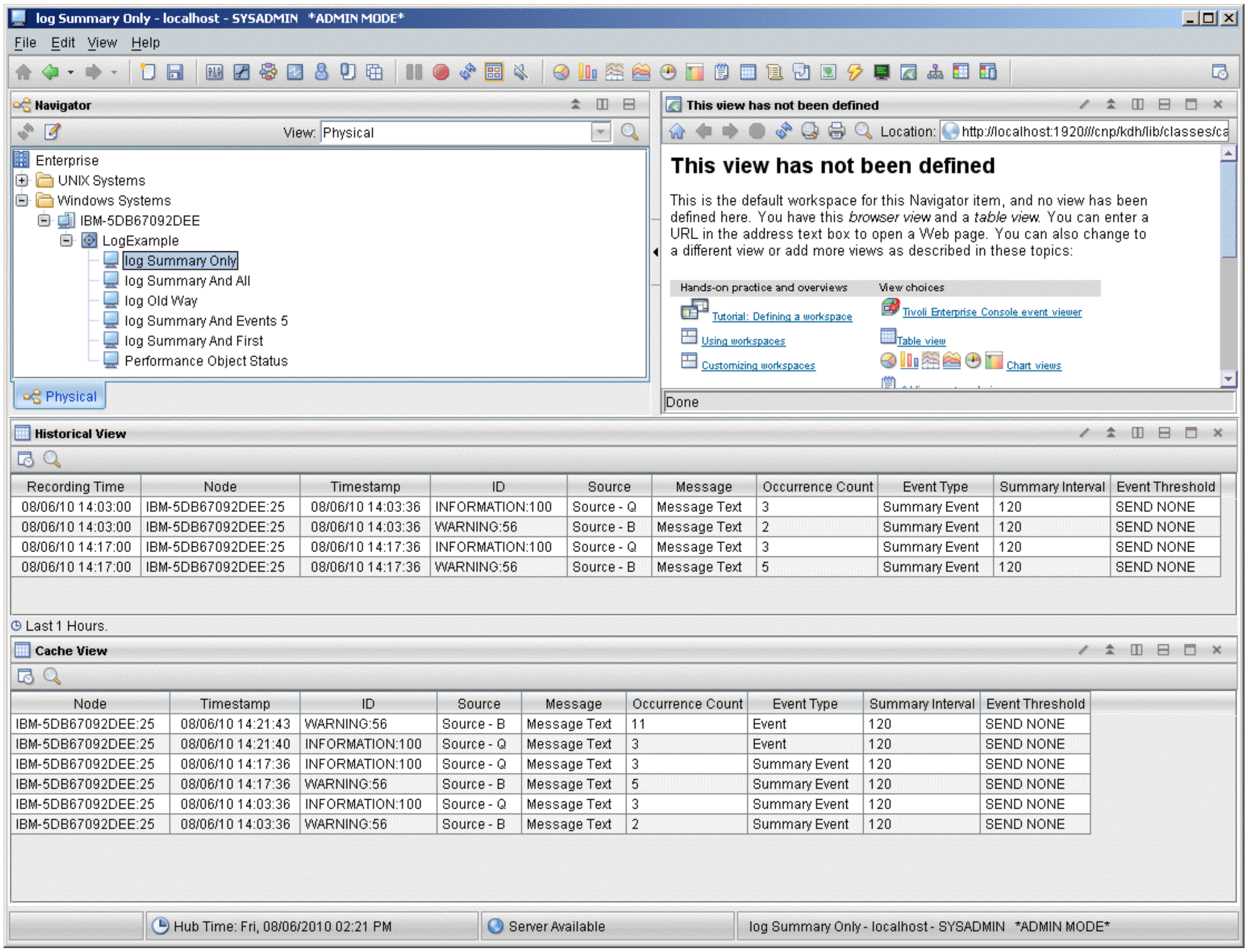
Cache View

Node	Timestamp	ID	Source	Message
IBM-5DB67092DEE:25	08/06/10 14:21:48	WARNING:56	Source - B	Message Text
IBM-5DB67092DEE:25	08/06/10 14:21:48	WARNING:56	Source - B	Message Text
IBM-5DB67092DEE:25	08/06/10 14:21:47	WARNING:56	Source - B	Message Text
IBM-5DB67092DEE:25	08/06/10 14:21:47	WARNING:56	Source - B	Message Text
IBM-5DB67092DEE:25	08/06/10 14:21:46	WARNING:56	Source - B	Message Text
IBM-5DB67092DEE:25	08/06/10 14:21:46	WARNING:56	Source - B	Message Text
IBM-5DB67092DEE:25	08/06/10 14:21:45	WARNING:56	Source - B	Message Text
IBM-5DB67092DEE:25	08/06/10 14:21:45	WARNING:56	Source - B	Message Text
IBM-5DB67092DEE:25	08/06/10 14:21:44	WARNING:56	Source - B	Message Text
IBM-5DB67092DEE:25	08/06/10 14:21:44	WARNING:56	Source - B	Message Text
IBM-5DB67092DEE:25	08/06/10 14:21:43	WARNING:56	Source - B	Message Text
IBM-5DB67092DEE:25	08/06/10 14:21:41	INFORMATION:100	Source - Q	Message Text
IBM-5DB67092DEE:25	08/06/10 14:21:40	INFORMATION:100	Source - Q	Message Text
IBM-5DB67092DEE:25	08/06/10 14:21:40	INFORMATION:100	Source - Q	Message Text
IBM-5DB67092DEE:25	08/06/10 14:16:25	INFORMATION:100	Source - Q	Message Text
IBM-5DB67092DEE:25	08/06/10 14:16:25	INFORMATION:100	Source - Q	Message Text

Hub Time: Fri, 08/06/2010 02:22 PM | Server Available | log Old Way - localhost - SYSADMIN *ADMIN MODE*

Figure 75. Historical view and cache view when event filtering or summarization is not enabled

(Figure 76 on page 1376) shows the historical view and cache view if you selected the **Only send summary events** option in the **Event Information** tab. The summary events are displayed in both views, but the new events are only displayed in the real-time (cache) view.


log Summary Only - localhost - SYSADMIN *ADMIN MODE*
File Edit View Help
Navigator
View: Physical
Enterprise
UNIX Systems
Windows Systems
IBM-5DB67092DEE
LogExample
log Summary Only
log Summary And All
log Old Way
log Summary And Events 5
log Summary And First
Performance Object Status
Physical
This view has not been defined
Location: http://localhost:1920///cnp/kdh/lib/classes/ca
Hands-on practice and overviews
Tutorial: Defining a workspace
Using workspaces
Customizing workspaces
View choices
Tivoli Enterprise Console event viewer
Table view
Chart views
Done
Historical View
Last 1 Hours.
Cache View
Hub Time: Fri, 08/06/2010 02:21 PM
Server Available
log Summary Only - localhost - SYSADMIN *ADMIN MODE*

This view has not been defined

This is the default workspace for this Navigator item, and no view has been defined here. You have this *browser view* and a *table view*. You can enter a URL in the address text box to open a Web page. You can also change to a different view or add more views as described in these topics:

Historical View

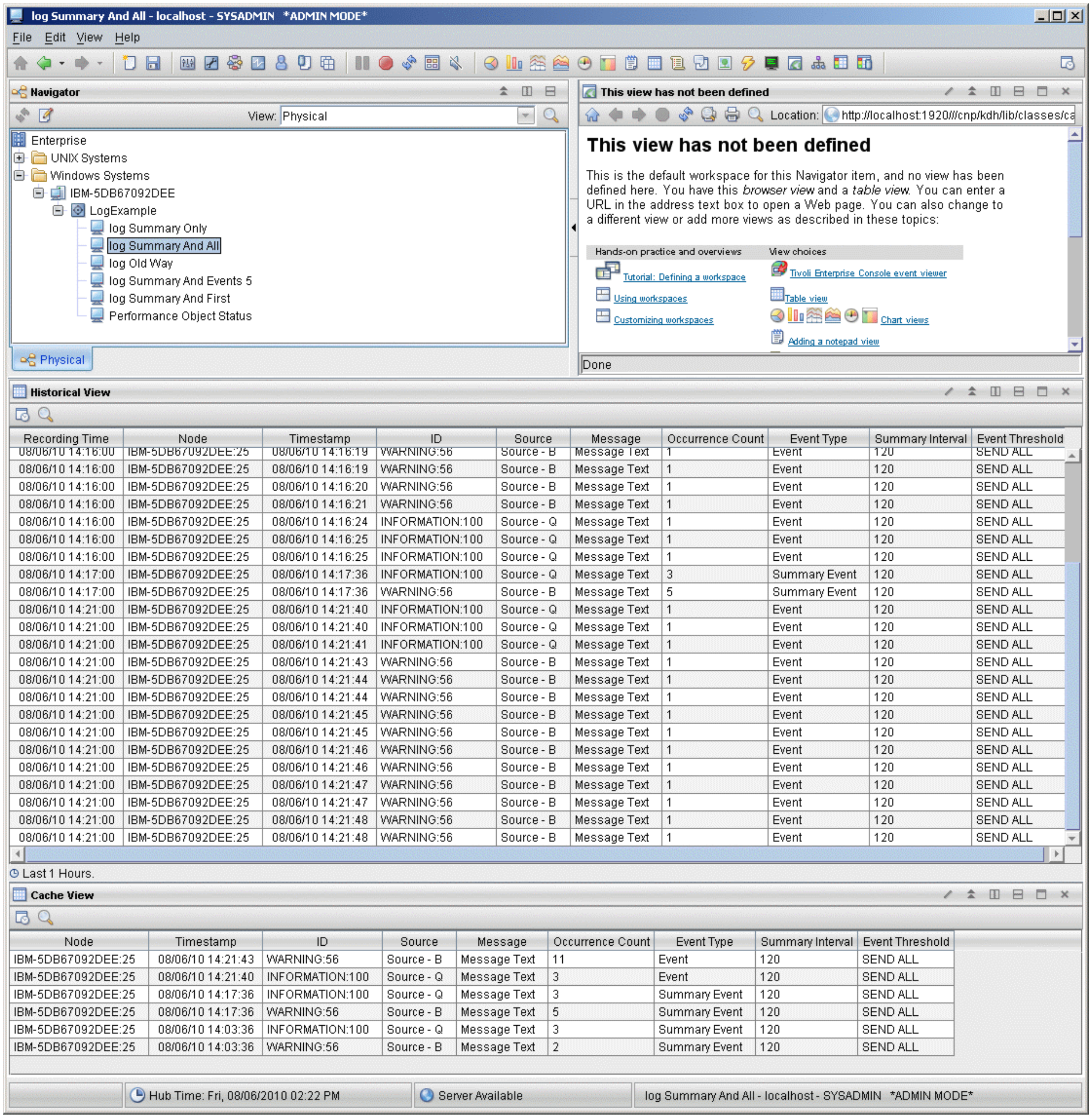
Recording Time	Node	Timestamp	ID	Source	Message	Occurrence Count	Event Type	Summary Interval	Event Threshold
08/06/10 14:03:00	IBM-5DB67092DEE:25	08/06/10 14:03:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	SEND NONE
08/06/10 14:03:00	IBM-5DB67092DEE:25	08/06/10 14:03:36	WARNING:56	Source - B	Message Text	2	Summary Event	120	SEND NONE
08/06/10 14:17:00	IBM-5DB67092DEE:25	08/06/10 14:17:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	SEND NONE
08/06/10 14:17:00	IBM-5DB67092DEE:25	08/06/10 14:17:36	WARNING:56	Source - B	Message Text	5	Summary Event	120	SEND NONE

Cache View

Node	Timestamp	ID	Source	Message	Occurrence Count	Event Type	Summary Interval	Event Threshold
IBM-5DB67092DEE:25	08/06/10 14:21:43	WARNING:56	Source - B	Message Text	11	Event	120	SEND NONE
IBM-5DB67092DEE:25	08/06/10 14:21:40	INFORMATION:100	Source - Q	Message Text	3	Event	120	SEND NONE
IBM-5DB67092DEE:25	08/06/10 14:17:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	SEND NONE
IBM-5DB67092DEE:25	08/06/10 14:17:36	WARNING:56	Source - B	Message Text	5	Summary Event	120	SEND NONE
IBM-5DB67092DEE:25	08/06/10 14:03:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	SEND NONE
IBM-5DB67092DEE:25	08/06/10 14:03:36	WARNING:56	Source - B	Message Text	2	Summary Event	120	SEND NONE

Figure 76. Historical view and cache view when ***Only send summary events*** *is selected*

(Figure 77 on page 1377) shows the historical view and cache view if you selected the **Send all events** option in the **Event Information** tab. All of the events are shown in both views, but you also see the summary events that are created at the end of each interval. The real-time view changes when the interval elapses. The existing events are converted into summary records and then the new events are added. The addition of the other two available event attributes that are used to display the summary interval (120 seconds in this example) and the *SEND ALL* threshold.


log Summary And All - localhost - SYSADMIN *ADMIN MODE*
File
Edit
View
Help
Navigator
View: Physical
Physical
This view has not been defined
Location: http://localhost:1920///cnp/kdh/lib/classes/ca
Done
Historical View
Cache View

Enterprise
- UNIX Systems
- Windows Systems
  - IBM-5DB67092DEE
    - LogExample
      - log Summary Only
      - log Summary And All
      - log Old Way
      - log Summary And Events 5
      - log Summary And First
      - Performance Object Status

This view has not been defined

This is the default workspace for this Navigator item, and no view has been defined here. You have this *browser view* and a *table view*. You can enter a URL in the address text box to open a Web page. You can also change to a different view or add more views as described in these topics:

Hands-on practice and overviews	View choices
Tutorial: Defining a workspace	Tivoli Enterprise Console event viewer
Using workspaces	Table view
Customizing workspaces	Chart views
	Adding a notepad view

Recording Time	Node	Timestamp	ID	Source	Message	Occurrence Count	Event Type	Summary Interval	Event Threshold
08/06/10 14:16:00	IBM-5DB67092DEE:25	08/06/10 14:16:19	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL
08/06/10 14:16:00	IBM-5DB67092DEE:25	08/06/10 14:16:19	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL
08/06/10 14:16:00	IBM-5DB67092DEE:25	08/06/10 14:16:20	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL
08/06/10 14:16:00	IBM-5DB67092DEE:25	08/06/10 14:16:21	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL
08/06/10 14:16:00	IBM-5DB67092DEE:25	08/06/10 14:16:24	INFORMATION:100	Source - Q	Message Text	1	Event	120	SEND ALL
08/06/10 14:16:00	IBM-5DB67092DEE:25	08/06/10 14:16:25	INFORMATION:100	Source - Q	Message Text	1	Event	120	SEND ALL
08/06/10 14:16:00	IBM-5DB67092DEE:25	08/06/10 14:16:25	INFORMATION:100	Source - Q	Message Text	1	Event	120	SEND ALL
08/06/10 14:17:00	IBM-5DB67092DEE:25	08/06/10 14:17:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	SEND ALL
08/06/10 14:17:00	IBM-5DB67092DEE:25	08/06/10 14:17:36	WARNING:56	Source - B	Message Text	5	Summary Event	120	SEND ALL
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:40	INFORMATION:100	Source - Q	Message Text	1	Event	120	SEND ALL
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:40	INFORMATION:100	Source - Q	Message Text	1	Event	120	SEND ALL
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:41	INFORMATION:100	Source - Q	Message Text	1	Event	120	SEND ALL
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:43	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:44	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:44	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:45	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:45	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:46	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:46	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:47	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:47	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:48	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:48	WARNING:56	Source - B	Message Text	1	Event	120	SEND ALL

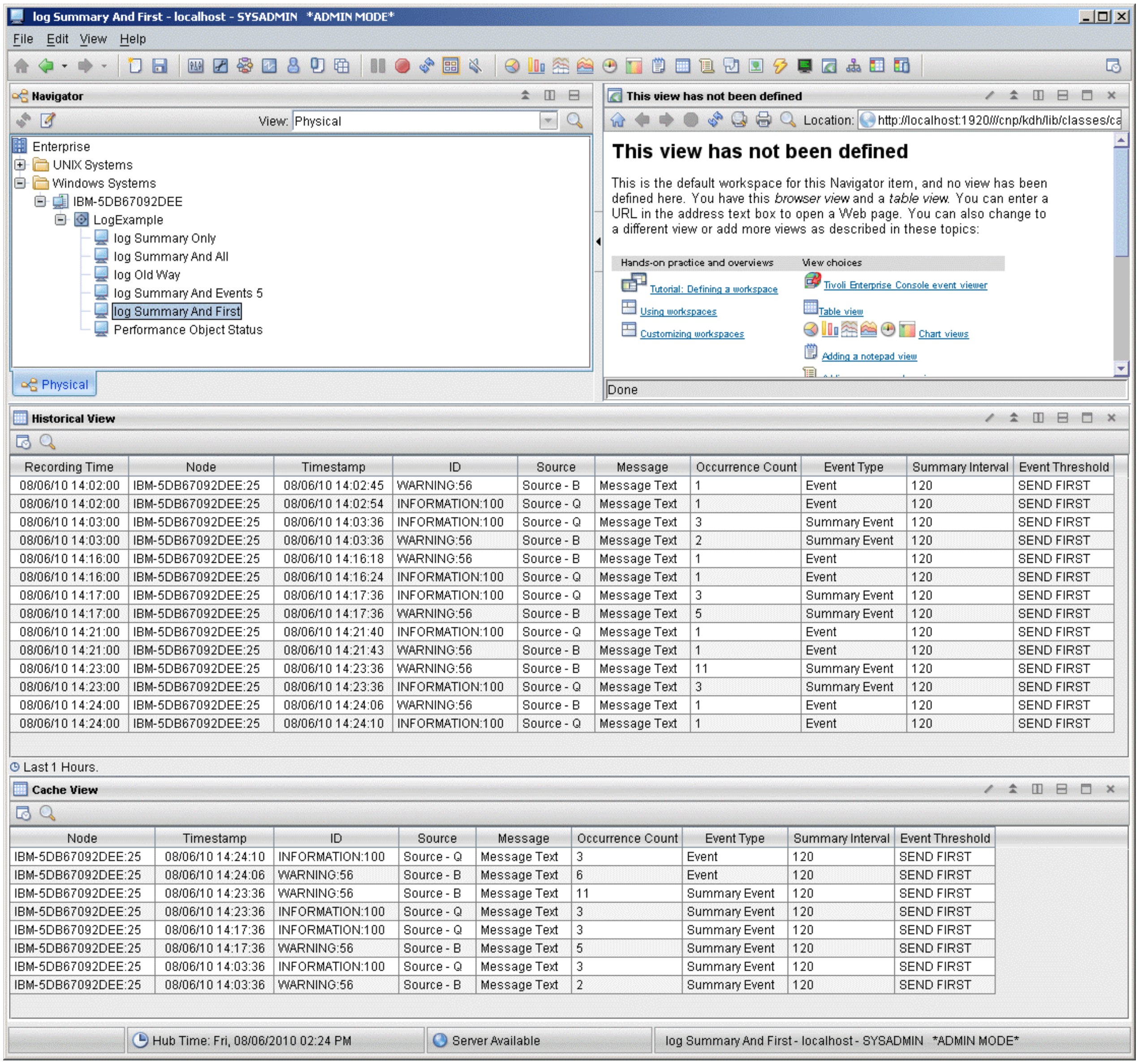
Last 1 Hours.

Node	Timestamp	ID	Source	Message	Occurrence Count	Event Type	Summary Interval	Event Threshold
IBM-5DB67092DEE:25	08/06/10 14:21:43	WARNING:56	Source - B	Message Text	11	Event	120	SEND ALL
IBM-5DB67092DEE:25	08/06/10 14:21:40	INFORMATION:100	Source - Q	Message Text	3	Event	120	SEND ALL
IBM-5DB67092DEE:25	08/06/10 14:17:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	SEND ALL
IBM-5DB67092DEE:25	08/06/10 14:17:36	WARNING:56	Source - B	Message Text	5	Summary Event	120	SEND ALL
IBM-5DB67092DEE:25	08/06/10 14:03:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	SEND ALL
IBM-5DB67092DEE:25	08/06/10 14:03:36	WARNING:56	Source - B	Message Text	2	Summary Event	120	SEND ALL

Hub Time: Fri, 08/06/2010 02:22 PM | Server Available | log Summary And All - localhost - SYSADMIN *ADMIN MODE*

*Figure 77. Historical view and cache view when **Send all events** is selected*

(Figure 78 on page 1378) shows the historical view and cache view if you selected the **Send first event** option in the **Event Information** tab. The summary events are displayed in both views, but all the new events are only displayed in the real-time (cache) view. For each event, the historical view displays only the first event that is received in the interval and no duplicate events.


log Summary And First - localhost - SYSADMIN *ADMIN MODE*
File Edit View Help
Navigator
View: Physical
Enterprise
UNIX Systems
Windows Systems
IBM-5DB67092DEE
LogExample
log Summary Only
log Summary And All
log Old Way
log Summary And Events 5
log Summary And First
Performance Object Status
Physical
This view has not been defined
Location: http://localhost:1920///cnp/kdh/lib/classes/ca
Hands-on practice and overviews
Tutorial: Defining a workspace
Using workspaces
Customizing workspaces
View choices
Tivoli Enterprise Console event viewer
Table view
Chart views
Adding a notepad view
Done
Historical View
Last 1 Hours.
Cache View
Hub Time: Fri, 08/06/2010 02:24 PM
Server Available
log Summary And First - localhost - SYSADMIN *ADMIN MODE*

This view has not been defined

This is the default workspace for this Navigator item, and no view has been defined here. You have this *browser view* and a *table view*. You can enter a URL in the address text box to open a Web page. You can also change to a different view or add more views as described in these topics:

Historical View

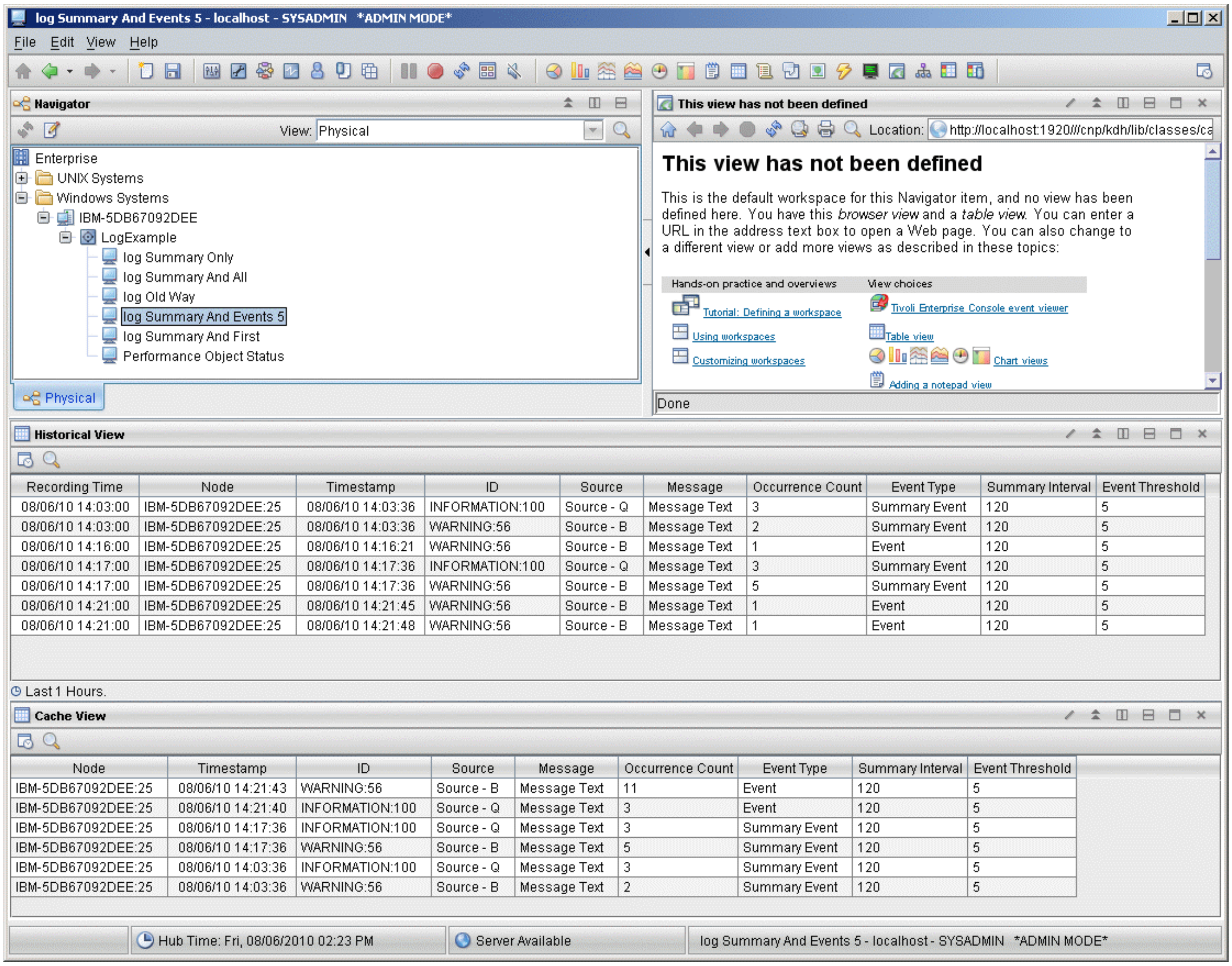
Recording Time	Node	Timestamp	ID	Source	Message	Occurrence Count	Event Type	Summary Interval	Event Threshold
08/06/10 14:02:00	IBM-5DB67092DEE:25	08/06/10 14:02:45	WARNING:56	Source - B	Message Text	1	Event	120	SEND FIRST
08/06/10 14:02:00	IBM-5DB67092DEE:25	08/06/10 14:02:54	INFORMATION:100	Source - Q	Message Text	1	Event	120	SEND FIRST
08/06/10 14:03:00	IBM-5DB67092DEE:25	08/06/10 14:03:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	SEND FIRST
08/06/10 14:03:00	IBM-5DB67092DEE:25	08/06/10 14:03:36	WARNING:56	Source - B	Message Text	2	Summary Event	120	SEND FIRST
08/06/10 14:16:00	IBM-5DB67092DEE:25	08/06/10 14:16:18	WARNING:56	Source - B	Message Text	1	Event	120	SEND FIRST
08/06/10 14:16:00	IBM-5DB67092DEE:25	08/06/10 14:16:24	INFORMATION:100	Source - Q	Message Text	1	Event	120	SEND FIRST
08/06/10 14:17:00	IBM-5DB67092DEE:25	08/06/10 14:17:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	SEND FIRST
08/06/10 14:17:00	IBM-5DB67092DEE:25	08/06/10 14:17:36	WARNING:56	Source - B	Message Text	5	Summary Event	120	SEND FIRST
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:40	INFORMATION:100	Source - Q	Message Text	1	Event	120	SEND FIRST
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:43	WARNING:56	Source - B	Message Text	1	Event	120	SEND FIRST
08/06/10 14:23:00	IBM-5DB67092DEE:25	08/06/10 14:23:36	WARNING:56	Source - B	Message Text	11	Summary Event	120	SEND FIRST
08/06/10 14:23:00	IBM-5DB67092DEE:25	08/06/10 14:23:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	SEND FIRST
08/06/10 14:24:00	IBM-5DB67092DEE:25	08/06/10 14:24:06	WARNING:56	Source - B	Message Text	1	Event	120	SEND FIRST
08/06/10 14:24:00	IBM-5DB67092DEE:25	08/06/10 14:24:10	INFORMATION:100	Source - Q	Message Text	1	Event	120	SEND FIRST

Cache View

Node	Timestamp	ID	Source	Message	Occurrence Count	Event Type	Summary Interval	Event Threshold
IBM-5DB67092DEE:25	08/06/10 14:24:10	INFORMATION:100	Source - Q	Message Text	3	Event	120	SEND FIRST
IBM-5DB67092DEE:25	08/06/10 14:24:06	WARNING:56	Source - B	Message Text	6	Event	120	SEND FIRST
IBM-5DB67092DEE:25	08/06/10 14:23:36	WARNING:56	Source - B	Message Text	11	Summary Event	120	SEND FIRST
IBM-5DB67092DEE:25	08/06/10 14:23:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	SEND FIRST
IBM-5DB67092DEE:25	08/06/10 14:17:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	SEND FIRST
IBM-5DB67092DEE:25	08/06/10 14:17:36	WARNING:56	Source - B	Message Text	5	Summary Event	120	SEND FIRST
IBM-5DB67092DEE:25	08/06/10 14:03:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	SEND FIRST
IBM-5DB67092DEE:25	08/06/10 14:03:36	WARNING:56	Source - B	Message Text	2	Summary Event	120	SEND FIRST

Figure 78. Historical view and cache view when ***Send first event*** *is selected*

(Figure 79 on page 1379) shows the historical view and cache view if you selected the **Event threshold** option and entered a value of 5. The summary events are displayed in both views, but all the new events are only displayed in the real-time (cache) view. In this example, a threshold of 5 is specified. The historical view displays an event only when five duplicates of an event (including the first event) are received in the interval. If less than 5 are received, no event is displayed. If 6, 7, 8, or 9 duplicates are received in the interval, one event is displayed. If 10 duplicates are received, 2 events are displayed.


log Summary And Events 5 - localhost - SYSADMIN *ADMIN MODE*
File
Edit
View
Help
Navigator
View: Physical
Enterprise
UNIX Systems
Windows Systems
IBM-5DB67092DEE
LogExample
log Summary Only
log Summary And All
log Old Way
log Summary And Events 5
log Summary And First
Performance Object Status
Physical
This view has not been defined
Location: http://localhost:1920///cnp/kdh/lib/classes/ca
Hands-on practice and overviews
Tutorial: Defining a workspace
Using workspaces
Customizing workspaces
View choices
Tivoli Enterprise Console event viewer
Table view
Chart views
Adding a notepad view
Done
Historical View
Last 1 Hours.
Cache View
Hub Time: Fri, 08/06/2010 02:23 PM
Server Available
log Summary And Events 5 - localhost - SYSADMIN *ADMIN MODE*

This view has not been defined

This is the default workspace for this Navigator item, and no view has been defined here. You have this *browser view* and a *table view*. You can enter a URL in the address text box to open a Web page. You can also change to a different view or add more views as described in these topics:

Recording Time	Node	Timestamp	ID	Source	Message	Occurrence Count	Event Type	Summary Interval	Event Threshold
08/06/10 14:03:00	IBM-5DB67092DEE:25	08/06/10 14:03:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	5
08/06/10 14:03:00	IBM-5DB67092DEE:25	08/06/10 14:03:36	WARNING:56	Source - B	Message Text	2	Summary Event	120	5
08/06/10 14:16:00	IBM-5DB67092DEE:25	08/06/10 14:16:21	WARNING:56	Source - B	Message Text	1	Event	120	5
08/06/10 14:17:00	IBM-5DB67092DEE:25	08/06/10 14:17:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	5
08/06/10 14:17:00	IBM-5DB67092DEE:25	08/06/10 14:17:36	WARNING:56	Source - B	Message Text	5	Summary Event	120	5
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:45	WARNING:56	Source - B	Message Text	1	Event	120	5
08/06/10 14:21:00	IBM-5DB67092DEE:25	08/06/10 14:21:48	WARNING:56	Source - B	Message Text	1	Event	120	5

Node	Timestamp	ID	Source	Message	Occurrence Count	Event Type	Summary Interval	Event Threshold
IBM-5DB67092DEE:25	08/06/10 14:21:43	WARNING:56	Source - B	Message Text	11	Event	120	5
IBM-5DB67092DEE:25	08/06/10 14:21:40	INFORMATION:100	Source - Q	Message Text	3	Event	120	5
IBM-5DB67092DEE:25	08/06/10 14:17:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	5
IBM-5DB67092DEE:25	08/06/10 14:17:36	WARNING:56	Source - B	Message Text	5	Summary Event	120	5
IBM-5DB67092DEE:25	08/06/10 14:03:36	INFORMATION:100	Source - Q	Message Text	3	Summary Event	120	5
IBM-5DB67092DEE:25	08/06/10 14:03:36	WARNING:56	Source - B	Message Text	2	Summary Event	120	5

Figure 79. Historical view and cache view when ***Event threshold*** *is selected*

Related concepts

"Event filtering and summarization" on page 1373

Troubleshooting and support

Review the troubleshooting information for problems that you might experience with installing, configuring, or using IBM Agent Builder.

For help with troubleshooting issues while developing, installing, or using custom agents in the IBM Cloud Application Performance Management environment, see the IBM Cloud Application Performance Management Troubleshooting Guide.

For logging and message reference information and for help with troubleshooting issues related to the IBM Tivoli Monitoring environment, see the IBM Agent Builder Version 6.3.1 Troubleshooting Guide.

Sharing project files

Share an IBM Tivoli Monitoring agent project with someone.

Procedure

1. Obtain their files. You need the entire contents of the directory with the same name as the project in your workspace directory.

For example, if your workspace directory is `c:\Documents and Settings\User1\workspace` and you want to share your project named `TestProject`. You must make the directory `c:\Documents and Settings\User1\workspace\TestProject` and all of its contents accessible to your system.

2. Select **File** > **Import**.
3. Open **IBM Tivoli Monitoring**.
4. Select **IBM Tivoli Monitoring Agent** and click **Next**.
5. Type the full path to the agent xml file or click **Browse** to browse to the file.
6. Click **Finish**.

Results

When the wizard completes, you see the new IBM Tivoli Monitoring agent project in your workspace.

Share a Solution Installer Project

Share a Solution Installer Project with someone

Procedure

1. Obtain their files. You must have the entire contents of the directory with the same name as the Solution Installer project in your workspace directory.
   For example, if your workspace directory is `c:\Documents and Settings\User1\workspace` and you want to share your Solution Installer project named `TestProject Installer`. You must make the directory `c:\Documents and Settings\User1\workspace\TestProject Installer` and all of its contents accessible to your system.
2. Click **File** > **Import**.
3. Open **General**.
4. Select **Existing Projects into Workspace**, and click **Next**.
5. Type the full path to the root directory of the Solution Installer project, or click **Browse** to browse to the root directory of the Solution Installer project. (In this example the `TestProject Installer` directory.) The Project in that directory is displayed in the Projects list and is selected by default.
6. Optional: Click **Copy projects into workspace**.
7. Click **Finish**.

Command-line options

Commands available from the Agent Builder command-line interface (CLI).

The Tivoli Monitoring Agent Builder contains a command-line interface (CLI) that you can use to generate the Tivoli Monitoring Agent without starting the Eclipse graphical user interface (GUI). You can generate the agent as part of a build, for example:

On Windows systems, you can use a batch file in the following directory to access the CLI:

```
install_location\agenttoolkit.bat
```

On UNIX and Linux systems, you can use a script in the following directory to access the CLI:

```
install_location/agenttoolkit.sh
```

The commands that are described in this documentation are formatted for Windows systems, which use a backslash (\) for directory paths.

For UNIX® or Linux® systems, use the same commands as for Windows systems, but with the following changes:

- Use a forward slash (/) for directory paths instead of a backslash (\).
- Use the `agenttoolkit.sh` script instead of the `agenttoolkit.bat` script.

Commands

Table 303 on page 1381 lists the name and purpose statement for each command option for the text command:

Table 303. Command quick-reference table

Command	Purpose
generatelocal	Loads and validates the `itm_toolkit_agent.xml` file and generates the files that run the Tivoli Monitoring Agent. The installation is into a local Tivoli Monitoring environment.
generatemappingfile	Creates the mapping file for porting custom IBM Tivoli Monitoring v5.x resource models to IBM Tivoli Monitoring v6 agents.
generatezip	Generates a compressed file named *`productcode`*`.zip` or *`productcode`*`.tgz`.

The command descriptions that are referenced from the table describes how to run the commands by covering the following information:

Purpose
Lists the purpose of the command.

Format
Specifies the syntax that you type on the command line. The syntax contains the command name and a list of the parameters for the command. A definition of each parameter follows the command name.

Examples
The example for the command contains a brief description of the example and an example of the syntax.

Usage
Provides an explanation of the command and its purpose.

Comments
Provides commands or text that can give you more information.

Command - `generatelocal`

Use this command to load and validate XML and to generate files to run the Tivoli Monitoring Agent.

Purpose

Loads and validates the `itm_toolkit_agent.xml` file and generates the files for running the Tivoli Monitoring Agent. The installation is into a local Tivoli Monitoring environment.

Format

For Windows systems:

```
install_location\agenttoolkit.bat project_dir -generatelocal itm_install_dir
```

where:

install_location
Directory where the Agent Builder is installed

project_dir
Name of the directory that contains the `itm_toolkit_agent.xml` file

itm_install_dir
Location where Tivoli Monitoring is installed (for example `c:\IBM\ITM`)

Examples

In the following example for Windows, the agent definition in `C:\ABCAgent` is validated and the files that are required to run `ABCAgent` are generated in `C:\IBM\ITM`:

```
install_location\agenttoolkit.bat C:\ABCAgent -generatelocal C:\IBM\ITM
```

Command - generatemappingfile

Use this command to migrate custom IBM Tivoli Monitoring v5.x resource models to IBM Tivoli Monitoring v6 agents.

Purpose

This command creates the mapping file for migrating custom IBM Tivoli Monitoring v5.x resource models to IBM Tivoli Monitoring v6 agents.

Format

For Windows systems:

```
install_location\agenttoolkit.bat project_dir -generatemappingfile output_dir
   itm5_interp_list
```

Where:

install_location
Directory where the Agent Builder is installed

project_dir
Name of the directory that contains `itm_toolkit_agent.xml`

output_dir
Name of the directory where the mapping file is written

itm5_interp_list
Comma-separated list of the ITM 5x operating systems on which the custom resource model ran. The following values are allowed:

- `aix4-r1`
- `hpux10`
- `linux-ix86`
- `linux-ppc`
- `linux-s390`
- `os2-ix86`
- `os400`
- `solaris2`
- `solaris2-ix86`
- `w32-ix86`

Examples

For Windows systems

```
install_location\agenttoolkit.bat c:\ABCAgent -generatemappingfile c:\output
   linux-ix86,linux-ppc,linux-s390
```

Command - generatezip

Use this command to load and validate XML and to generate a compressed file that can be used to install the agent on another system.

Purpose

Loads and validates the `itm_toolkit_agent.xml` file and generates a compressed file named *productcode*`.zip` or *productcode*`.tgz`. The generated compressed file can be used to install the agent on another system. Depending on your environment, both file types can be generated.

Format

For Windows systems:

```
install_location\agenttoolkit.bat project_dir -generatezip output_dir
```

Where:

project_dir
Name of a directory that contains the `itm_toolkit_agent.xml` file

output_dir
Name of a directory where the compressed file is written

Examples

In the following example for Windows, the agent definition in `C:\ABCAgent` is validated and a compressed file that contains the required files for running `ABCAgent` is generated in `C:\Output`:

```
install_location\agenttoolkit.bat\ C:\ABCAgent -generatezip C:\Output
```

Attributes reference

Contains descriptions of the attributes for each attribute generated group included in the Agent Builder.

Availability node

The Availability attribute group contains availability data for the application.

The table provides a common format for representing application availability, which includes relevant information for three aspects of an application: services (Windows only), processes, and command return codes.

The following list contains information about each attribute in the Availability attribute group:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The local time at the agent when the data was collected

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

Application Component attribute - This attribute is a key attribute

Description
The descriptive name of a part of the application

Type
String

Names

Attribute name
Application_Component

Column name
COMPONENT

Name attribute

Description
The name of the process, service, or functional test. This name matches the executable name of the process, the service short name, or the name of the process that is used to test the application.

Type
String

Names

Attribute name
Name

Column name
NAME

Status attribute

Description
The status of the application component.

- For processes, the values are `UP`, `DOWN`, `WARNING`, or `PROCESS_DATA_NOT_AVAILABLE`. `PROCESS_DATA_NOT_AVAILABLE` is displayed for a process when the matching process is running but the resource use information cannot be collected for that process.
- For services, the values are `UP`, `DOWN`, or `UNKNOWN`. `UNKNOWN` is displayed when the service is not installed.
- For command return codes, the values are `PASSED` or `FAILED`.

Type
String

Names

Attribute name
Status

Column name
STATUS

Full Name attribute

Description
The full name of the process which includes information that is process-dependent. The name might include the full path if the process was started that way. The name can also include a partial path or even a path that is changed by the process.

Type
String

Names

Attribute name
Full_Name

Column name
FULLNAME

Type attribute

Description
Identifies the type of the application component. Components are processes, services, or command return codes.

Type
Integer (gauge)

Names

Attribute name
Type

Column name
TYPE

Virtual Size attribute

Description
The virtual size (in MB) of the process

Type
Integer (gauge)

Names

Attribute name
Virtual_Size

Column name
VIRTSIZE

Page Faults Per Sec attribute

Description
The rate of page faults for the process that is measured in faults per second. This value contains only valid data for processes.

Type
Integer (gauge)

Names

Attribute name
Page_Faults_Per_Sec

Column name
PAGEFAULTS

Working Set Size attribute

Description
The working set size of the process in MB. This value contains only valid data for processes.

Type
Integer (gauge)

Names

Attribute name
Working_Set_Size

Column name
WORKSET

Thread Count attribute

Description
The number of threads that are currently allocated by this process. This value contains only valid data for processes.

Type
Integer (gauge)

Names

Attribute name
Thread_Count

Column name
THREADS

PID attribute

Description
The process id that is associated with the process. This value contains only valid data for processes.

Type
Integer (gauge)

Names

Attribute name
PID

Column name
PID

Percent Privileged Time attribute

Description
The percentage of the available processor time that is being used by this process for privileged operation

Type
Integer (gauge)

Names

Attribute name
Percent_Privileged_Time

Column name
PERCPRIV

Percent User Mode Time attribute

Description
The percentage of the available processor time that is being used by this process for user mode operation

Type
Integer (gauge)

Names

Attribute name
Percent_User_Mode_Time

Column name
PERCUSER

Percent Processor Time attribute

Description
The percentage of the elapsed time that this process used the processor to run instructions

Type
Integer (gauge)

Names

Attribute name
Percent_Processor_Time

Column name
PERCPROC

Command Line attribute

Description
The program name and any arguments that are specified on the command line when the process was started. This attribute has the value *N/A* if you are running a Service or Functionality test.

Type
String

Names

Attribute name
Command_Line

Column name
CMDLINE

Functionality Test Status attribute

Description
The return code of the functionality test. When the monitored application is running correctly, `SUCCESS` is returned. `NOT_RUNNING` is returned when the application is not running correctly. `N/A` is returned when the row does not represent a functionality test.

Type
Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal, the warehouse, and queries return the numbers. The defined values are: `N/A(1)`, `SUCCESS (0)`, `GENERAL_ERROR (2)`, `WARNING (3)`, `NOT_RUNNING (4)`, `DEPENDENT_NOT_RUNNING (5)`, `ALREADY_RUNNING (6)`, `PREREQ_NOT_RUNNING (7)`, `TIMED_OUT (8)`, `DOESNT_EXIST (9)`, `UNKNOWN (10)`, `DEPENDENT_STILL_RUNNING (11)`, or `INSUFFICIENT_USER_AUTHORITY (12)`. Any other values display the numeric value in the Tivoli Enterprise Portal.

Names

Attribute name
Functionality_Test_Status

Column name
FUNCSTATUS

Functionality Test Message attribute

Description
The text message that corresponds to the Functionality Test Status. This attribute is valid only for command return codes.

Type
String

Names

Attribute name
Functionality_Test_Message

Column name
FUNCMSG

Performance Object Status node

Use the Performance Object Status attribute group to see the status of all of the attribute groups that make up the agent. Each of the attribute groups is represented by a row in this table or other type of view. The status of an attribute group reflects the result of the last data collection attempt, or data reception event, for the attribute group. When you check the status information, you can see whether the agent is operating correctly. When your agent does not collect data, but receives it (event data), attributes that relate to sampled data do not contain useful data. Only the first seven attributes that are listed are relevant for event data.

Historical group
This attribute group is eligible for use with Tivoli Data Warehouse.

Attribute descriptions
The following list contains information about each attribute in the Performance Object Status attribute group:

Node attribute: This attribute is a key attribute.

Description
The managed system name of the agent.

Type
String

Warehouse name
NODE

Timestamp attribute

Description
The local time at the agent when the data was collected.

Type
String

Warehouse name
TIMESTAMP

Query Name attribute: This attribute is a key attribute.

Description
The name of the attribute group.

Type
String

Warehouse name
QUERY_NAME or ATTRGRP

Object Name attribute

Description
The name of the performance object.

Type
String

Warehouse name
OBJECT_NAME or OBJNAME

Object Type attribute

Description
The type of the performance object.

Type
Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- WMI (0)
- PERFMON (1)
- WMI ASSOCIATION GROUP (2)
- JMX (3)
- SNMP (4)
- SHELL COMMAND (5)
- JOINED GROUPS (6)
- CIMOM (7)
- CUSTOM (8)
- ROLLUP DATA (9)
- WMI REMOTE DATA (10)
- LOG FILE (11)
- JDBC (12)
- CONFIG DISCOVERY (13)
- NT EVENT LOG (14)
- FILTER (15)
- SNMP EVENT (16)
- PING (17)
- DIRECTOR DATA (18)
- DIRECTOR EVENT (19)
- SSH REMOTE SHELL COMMAND (20)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name
OBJECT_TYPE or OBJTYPE

Object Status attribute

Description
The status of the performance object.

Type
Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- ACTIVE (0)
- INACTIVE (1)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name
OBJECT_STATUS or OBJSTTS

Error Code attribute

Description
The error code that is associated with the query.

Type
Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NO ERROR (0)
- GENERAL ERROR (1)
- OBJECT NOT FOUND (2)
- COUNTER NOT FOUND (3)
- NAMESPACE ERROR (4)
- OBJECT CURRENTLY UNAVAILABLE (5)
- COM LIBRARY INIT FAILURE (6)
- SECURITY INIT FAILURE (7)
- PROXY SECURITY FAILURE (9)
- NO INSTANCES RETURNED (10)
- ASSOCIATOR QUERY FAILED (11)
- REFERENCE QUERY FAILED (12)
- NO RESPONSE RECEIVED (13)
- CANNOT FIND JOINED QUERY (14)
- CANNOT FIND JOIN ATTRIBUTE IN QUERY 1 RESULTS (15)
- CANNOT FIND JOIN ATTRIBUTE IN QUERY 2 RESULTS (16)
- QUERY 1 NOT A SINGLETON (17)
- QUERY 2 NOT A SINGLETON (18)
- NO INSTANCES RETURNED IN QUERY 1 (19)
- NO INSTANCES RETURNED IN QUERY 2 (20)
- CANNOT FIND ROLLUP QUERY (21)
- CANNOT FIND ROLLUP ATTRIBUTE (22)
- FILE OFFLINE (23)
- NO HOSTNAME (24)

- MISSING LIBRARY (25)
- ATTRIBUTE COUNT MISMATCH (26)
- ATTRIBUTE NAME MISMATCH (27)
- COMMON DATA PROVIDER NOT STARTED (28)
- CALLBACK REGISTRATION ERROR (29)
- MDL LOAD ERROR (30)
- AUTHENTICATION FAILED (31)
- CANNOT RESOLVE HOST NAME (32)
- SUBNODE UNAVAILABLE (33)
- SUBNODE NOT FOUND IN CONFIG (34)
- ATTRIBUTE ERROR (35)
- CLASSPATH ERROR (36)
- CONNECTION FAILURE (37)
- FILTER SYNTAX ERROR (38)
- FILE NAME MISSING (39)
- SQL QUERY ERROR (40)
- SQL FILTER QUERY ERROR (41)
- SQL DB QUERY ERROR (42)
- SQL DB FILTER QUERY ERROR (43)
- PORT OPEN FAILED (44)
- ACCESS DENIED (45)
- TIMEOUT (46)
- NOT IMPLEMENTED (47)
- REQUESTED A BAD VALUE (48)
- RESPONSE TOO BIG (49)
- GENERAL RESPONSE ERROR (50)
- SCRIPT NONZERO RETURN (51)
- SCRIPT NOT FOUND (52)
- SCRIPT LAUNCH ERROR (53)
- CONF FILE DOES NOT EXIST (54)
- CONF FILE ACCESS DENIED (55)
- INVALID CONF FILE (56)
- EIF INITIALIZATION FAILED (57)
- CANNOT OPEN FORMAT FILE (58)
- FORMAT FILE SYNTAX ERROR (59)
- REMOTE HOST UNAVAILABLE (60)
- EVENT LOG DOES NOT EXIST (61)
- PING FILE DOES NOT EXIST (62)
- NO PING DEVICE FILES (63)
- PING DEVICE LIST FILE MISSING (64)
- SNMP MISSING PASSWORD (65)
- DISABLED (66)
- URLS FILE NOT FOUND (67)

- XML PARSE ERROR (68)
- NOT INITIALIZED (69)
- ICMP SOCKETS FAILED (70)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name
ERROR_CODE or ERRCODE

Last Collection Start attribute

Description
The most recent time a data collection of this group started.

Type
Time stamp with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NOT COLLECTED (0691231190000000)
- NOT COLLECTED (0000000000000001)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name
LAST_COLLECTION_START or COLSTRT

Last Collection Finished attribute

Description
The most recent time a data collection of this group finished.

Type
Time stamp with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- NOT COLLECTED (0691231190000000)
- NOT COLLECTED (0000000000000001)

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name
LAST_COLLECTION_FINISHED or COLFINI

Last Collection Duration attribute

Description
The duration of the most recently completed data collection of this group in seconds.

Type
Real number (32-bit counter) with two decimal places of precision

Warehouse name
LAST_COLLECTION_DURATION or COLDURA

Average Collection Duration attribute

Description
The average duration of all data collections of this group in seconds.

Type
Real number (32-bit counter) with two decimal places of precision with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the values that are shown in parentheses. The following values are defined:

- `NO DATA (-100)`

Any other value is the value that is returned by the agent in the Tivoli Enterprise Portal.

Warehouse name
AVERAGE_COLLECTION_DURATION or COLAVGD

Refresh Interval attribute

Description
The interval at which this group is refreshed in seconds.

Type
Integer (32-bit counter)

Warehouse name
REFRESH_INTERVAL or REFRINT

Number of Collections attribute

Description
The number of times this group is collected since agent start.

Type
Integer (32-bit counter)

Warehouse name
NUMBER_OF_COLLECTIONS or NUMCOLL

Cache Hits attribute

Description
The number of times an external data request for this group is satisfied from the cache.

Type
Integer (32-bit counter)

Warehouse name
CACHE_HITS or CACHEHT

Cache Misses attribute

Description
The number of times an external data request for this group was not available in the cache.

Type
Integer (32-bit counter)

Warehouse name
CACHE_MISSES or CACHEMS

Cache Hit Percent attribute

Description
The percentage of external data requests for this group that are satisfied from the cache.

Type
Real number (32-bit counter) with two decimal places of precision

Warehouse name
CACHE_HIT_PERCENT or CACHPCT

Intervals Skipped attribute

Description
The number of times a background data collection was skipped because the previous collection was still running when the next one was due to start.

Type
Integer (32-bit counter)

Warehouse name
INTERVALS_SKIPPED or INTSKIP

Thread Pool Status attribute group

The Thread Pool Status attribute group contains information that reflects the status of the internal thread pool that is used to collect data asynchronously.

The following comprises a list of the attributes for this attribute group. The name in bold text shows how the attribute is displayed in the Tivoli Enterprise Portal.

The following list contains information about each attribute in the Thread Pool Status attribute group:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The time that is collected from the agent system when the data row was built and sent from the agent to the Tivoli Enterprise Monitoring Server. Or stored for historical purposes. It represents the local time zone of the agent system.

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

Thread Pool Size attribute

Description
The number of threads currently existing in the thread pool.

Type
Integer

Names

Attribute name
Thread_Pool_Size

Column name
THPSIZE

Thread Pool Max Size attribute

Description
The maximum number of threads that are allowed to exist in the thread pool.

Type
Integer

Names

Attribute name
Thread_Pool_Max_Size

Column name
TPMAXSZ

Thread Pool Active Threads attribute

Description
The number of threads in the thread pool currently active doing work.

Type
Integer

Names

Attribute name
Thread_Pool_Active_Threads

Column name
TPACTTH

Thread Pool Avg Active Threads attribute

Description
The average number of threads in the thread pool simultaneously active doing work.

Type
Integer

Names

Attribute name
Thread_Pool_Avg_Active_Threads

Column name
TPAVGAT

Thread Pool Min Active Threads attribute

Description
The minimum number of threads in the thread pool simultaneously active doing work.

Type
Integer

Names

Attribute name
Thread_Pool_Min_Active_Threads

Column name
TPMINAT

Thread Pool Max Active Threads attribute

Description
The peak number of threads in the thread pool simultaneously active doing work.

Type
Integer

Names

Attribute name
Thread_Pool_Max_Active_Threads

Column name
TPMAXAT

Thread Pool Queue Length attribute

Description
The number of jobs currently waiting in the thread pool queue.

Type
Integer

Names

Attribute name
Thread_Pool_Queue_Length

Column name
TPQLGTH

Thread Pool Avg Queue Length attribute

Description
The average length of the thread pool queue during this run.

Type
Integer

Names

Attribute name
Thread_Pool_Avg_Queue_Length

Column name
TPAVGQL

Thread Pool Min Queue Length attribute

Description
The minimum length the thread pool queue reached.

Type
Integer

Names

Attribute name
Thread_Pool_Min_Queue_Length

Column name
TPMINQL

Thread Pool Max Queue Length attribute

Description
The peak length the thread pool queue reached.

Type
Integer

Names

Attribute name
Thread_Pool_Max_Queue_Length

Column name
TPMAXQL

Thread Pool Avg Job Wait attribute

Description
The average time a job spends waiting on the thread pool queue.

Type
Integer

Names

Attribute name
Thread_Pool_Avg_Job_Wait

Column name
TPAVJBW

Thread Pool Total Jobs attribute

Description
The number of jobs that are completed by all threads in the pool since agent start.

Type
Integer

Names

Attribute name
Thread_Pool_Total_Jobs

Column name
TPTJOBS

Event log attribute node

The Event log attribute group contains any recent event log entries that pertain to the application.

By default, the agent displays only events that occur after the agent is started. Events are removed from the Event Log view 1 hour after they occur.

The following list contains information about each attribute in the Event Log attribute group:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Log Name attribute

Description
The event log - Application, System, Security, or an application-specific log

Type
String

Names

Attribute name
Log_Name

Column name
LOGNAME

Event Source attribute

Description
The event source that is defined by the application

Type
String

Names

Attribute name
Event_Source

Column name
EVTSOURCE

Event Type attribute

Description
Event Type - Error(0), Warning(1), Informational(2), Audit_Success(3), Audit_Failure(4), Unknown(5)

Type
Integer

Names

Attribute name
Event_Type

Column name
EVTTYPE

Event ID attribute

Description
The ID of the event

Type
Integer

Names

Attribute name
Event_ID

Column name
EVTID

Event Category attribute

Description
The category of the event

Type
String

Names

Attribute name
Event_Category

Column name
EVTCATEG

Message attribute

Description
The event message

Type
String

Names

Attribute name
Message

Column name
MESSAGE

Time Generated attribute

Description
The time the event was generated

Type
Time

Names

Attribute name
Time_Generated

Column name
TIMESTAMP

Log File Summary

The attributes of this attribute group are included in summary attribute groups when that option is selected in the advanced properties of the data source.

A Summary node is created for each Log File data source when **Include attribute in summary attribute group** is selected in the advanced properties of the data source. The name of the summary node is the name of the data source with Summary added to the end.

The following list contains information about each of the default attributes in the Log File Summary attribute group. These attributes are always included in summary attribute groups. If you select **Include attribute in summary attribute group**, see step "9" on page 1239 in ("Monitoring a log file" on page 1235), then the summary attribute group for that log attribute group also contains each of the attributes you selected. The values are a copy of the corresponding attribute in the log file attribute group.

All of the added attributes together become a key and the summary table includes one row per unique set of keys. The row indicates how many log records are received during the interval where all of the provided keys matched the value reported in the corresponding attributes.

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The local time at the agent when the data was collected

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

Interval Unit attribute

Description
The number of seconds between summary attribute generation

Type
Integer (gauge)

Names

Attribute name
_Interval_Unit

Column name
IU

Interval attribute

Description
Offset of the current interval within the next larger unit of time (for example, minutes within an hour)

Type
Integer (gauge)

Names

Attribute name
_Interval

Column name
INV

Occurrences attribute

Description
The number of occurrences that are recorded during the interval

Type
Integer (gauge)

Names

Attribute name
_Occurrences

Column name
OCC

LocalTimeStamp attribute

Description
The time that the summary data was generated

Type
Timestamp

Names

Attribute name
_LocalTimeStamp

Column name
LTS

DateTime attribute

Description
The time that the summary data was generated

Type
String

Names

Attribute name
_Date_Time

Column name
DT

Interval Unit Name attribute

Description
The word description of the interval unit

Type
String

Names

Attribute name
_Interval_Unit_Name

Column name
IUN

AIX Binary Log attribute group

The AIX Binary Log attribute group displays events from the AIX Binary Log as selected by the provided `errpt` command string.

The following list contains information about each attribute in the AIX Binary Log Attribute Group:

Note: The Agent Builder prevents removing, reordering, or changing the size of the `Identifier`, `ErrptTimestamp`, `Type`, `Class`, `ResourceName`, and `Description` attributes. The agent parses the data that comes back from an `errpt` command that is based on columns within the line of text. These columns are defined by the order and size of the `Identifier`, `ErrptTimestamp`, `Type`, `Class`, `ResourceName`, and `Description` attributes. Removing, reordering, or changing the size of these attributes, changes the attribute that the various columns go into. The resulting row as seen in Tivoli Monitoring is then incorrect.

You can, however, rename these attributes.

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Identifier attribute - This attribute is a key attribute

Description
The event identifier reported by `errpt`

Type
String

Names

Attribute name
Identifier

Column name
IDENTIFIER

ErrptTimestamp attribute

Description
The time the event is recorded as reported by `errpt`.

Note: This attribute is hidden at run time. This attribute contains a raw value. Other attributes that are derived from this attribute display the value in a more usable form. This attribute is available from within Agent Builder for that purpose, but by default it is not visible in the Tivoli Monitoring environment at run time. If you want to make it visible, select the attribute in the **Data Source Definition** page in the Agent Editor and select **Display attribute in the Tivoli Enterprise Portal**.

Type
String

Names

Attribute name
ErrptTimestamp

Column name
ERRPTTIMES

Type

Description
The single character event type reported by `errpt`, one of `I(NFO)`, `P(END/ERF/ERM)`, `T(EMP)`, and `U(NKN)`

Type
String

Names

Attribute name
Type

Column name
TYPE

Class attribute - This attribute is a key attribute

Description
The event class reported by `errpt`, one of `Hardware`, `Software`, `Operator`, and `Undertermined`. These values are enumerated. The raw values for use with situations are `H`, `S`, `O`, and `U`.

Type
String

Names

Attribute name
Class

Column name
CLASS

ResourceName

Description
The resource name reported by `errpt`, identifies the origin of the error record

Type
String

Names

Attribute name
ResourceName

Column name
RESOURCENA

Description attribute

Description
The description reported by `errpt`, typically a short text message that describes the nature of the error

Type
String

Names

Attribute name
Description

Column name
DESCRIPTIO

LogFile attribute

Description
The full name of the binary `errpt` log including the path.

Note: This attribute is hidden at run time. This attribute contains a raw value. Other attributes that are derived from this attribute display the value in a more useable form. This attribute is available from within Agent Builder for that purpose, but by default it is not visible in the Tivoli Monitoring environment at run time. If you want to make it visible, select the attribute in the **Data Source Definition** page in the Agent Editor and select **Display attribute in the Tivoli Enterprise Portal**.

Type
String

Names

Attribute name
LogFile

Column name
LOGFILE

System attribute

Description
The host name of the system where the error was collected

Type
String

Names

Attribute name
System

Column name
SYSTEM

LogName attribute

Description
The base name of the binary `errpt` log from which the record was collected

Type
String

Names

Attribute name
LogName

Column name
LOGNAME

LogPath attribute

Description
The directory name that contains the binary `errpt` log from which the record was collected

Type
String

Names

Attribute name
LogPath

Column name
LOGPATH

EntryTime attribute

Description
The time the event is recorded as reported by `errpt` in Tivoli Timestamp format. This time is not necessarily identical to the time when the agent received the event, as recorded in the **`Timestamp`** field.

Type
Time stamp

Names

Attribute name
EntryTime

Column name
ENTRYTIME

Monitor and Notification attribute groups

Definitions for the Monitor and Notification attribute groups.

The first 4 are specific to monitors and the last is for notifications (all are related to JMX).

Each one is listed with an indication whether it is event-based or not. For non-event based attribute groups, data is collected when needed. For event-based attribute groups, the agent maintains a cache of the last 100 events received. These events are used to respond to requests from the Tivoli Enterprise Portal. The events are forwarded immediately for analysis by situations and warehousing.

Counter Notifications

The Counter Notifications attribute group is a non-event based attribute group that sends events that are received by all counter monitors.

The following list contains information about each attribute in the Counter Notifications attribute group:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The local time at the agent when the data was collected

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

Notification Type attribute

Description
The type of notification received. Describes how the observed attribute of the MBean triggered the notification.

Type
String

Names

Attribute name
Notification_Type

Column name
NOTIFICATI

Monitor ID attribute

Description
Monitor ID of the monitor who generated this notification

Type
Integer

Names

Attribute name
Monitor_ID

Column name
MONITOR_ID

Observed MBean attribute

Description
The MBean whose attribute is being monitored

Type
String

Names

Attribute name
Observed_MBean

Column name
OBSERVED_M

Observed Attribute attribute

Description
Name of the attribute that is monitored in the Observed MBean

Type
String

Names

Attribute name
Observed_Attribute

Column name
OBSERVED_A

Threshold attribute

Description
The current threshold of the monitor

Type
String

Names

Attribute name
Threshold

Column name
THRESHOLD

Offset attribute

Description
The value added to the threshold each time the attribute exceeds the threshold. This value forms a new threshold.

Type
String

Names

Attribute name
Offset

Column name
OFFSET

Modulus attribute

Description
The maximum value of the attribute. When it reaches this value, it rolls over and begins counting again from zero.

Type
Integer

Names

Attribute name
Modulus

Column name
MODULUS

Counter Value attribute

Description
Value of the counter that triggered the notification

Type
Integer

Names

Attribute name
Counter_Value

Column name
COUNTER_VA

Notification Time Stamp attribute

Description
Time that the notification was triggered

Type
Time

Names

Attribute name
Notification_Time_Stamp

Column name
NOTIFICAT0

Notification Message attribute

Description
The message in the notification

Type
String

Names

Attribute name
Notification_Message

Column name
NOTIFICAT1

Gauge Notifications

The Gauge Notifications attribute group is a non-event based attribute group that sends events that are received by all gauge monitors.

The following list contains information about each attribute in the Gauge Notifications attribute group:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The local time at the agent when the data was collected

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

Notification Type attribute

Description
The type of notification received. Describes how the observed attribute of the MBean triggered the notification.

Type
String

Names

Attribute name
Notification_Type

Column name
NOTIFICATI

Monitor ID attribute

Description
Monitor ID of the monitor who generated this notification

Type
Integer

Names

Attribute name
Monitor_ID

Column name
MONITOR_ID

Observed MBean attribute

Description
The MBean whose attribute is being monitored

Type
String

Names

Attribute name
Observed_MBean

Column name
OBSERVED_M

Observed Attribute attribute

Description
Name of the attribute that is monitored in the Observed MBean

Type
String

Names

Attribute name
Observed_Attribute

Column name
OBSERVED_A

Low Threshold attribute

Description
The threshold that the monitor is watching for the observed attribute to cross

Type
String

Names

Attribute name
Low_Threshold

Column name
LOW_THRESH

High Threshold attribute

Description
The threshold that the monitor is watching for the observed attribute to cross

Type
String

Names

Attribute name
High_Threshold

Column name
HIGH_THRES

Gauge Value attribute

Description
Value of the gauge that triggered the notification

Type
String

Names

Attribute name
Gauge_Value

Column name
MODULUSGAUGE_VALU

Notification Time Stamp attribute

Description
Time that the notification was triggered

Type
Time

Names

Attribute name
Notification_Time_Stamp

Column name
NOTIFICAT0

Notification Message attribute

Description
The message in the notification

Type
String

Names

Attribute name
Notification_Message

Column name
NOTIFICAT1

Registered Monitors

The Registered Monitors attribute group is an event-based attribute group that shows a list of all JMX Monitors that are created by the agent.

The following list contains information about each attribute in the Registered Monitors attribute group:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The local time at the agent when the data was collected

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

Monitor ID attribute - This attribute is a key attribute

Description
The unique integer identifier for a monitor

Type
Integer

Names

Attribute name
Monitor_ID

Column name
MONITOR_ID

Monitor Parameters attribute

Description
The parameters that are used to create the monitor

Type
String

Names

Attribute name
Monitor_Parameters

Column name
MONITOR_PA

Monitor Name attribute

Description
The JMX Object Name of the monitor MBean

Type
String

Names

Attribute name
Monitor_Name

Column name
MONITOR_NA

String Notifications

The String Notifications attribute group is a non-event based attribute group that sends events that are received by all string monitors.

The following list contains information about each attribute in the String Notifications attribute group:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The local time at the agent when the data was collected

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

Notification Type attribute

Description
The type of notification received. Describes how the observed attribute of the MBean triggered the notification.

Type
String

Names

Attribute name
Notification_Type

Column name
NOTIFICATI

Monitor ID attribute - This attribute is a key attribute

Description
The unique integer identifier for a monitor

Type
Integer

Names

Attribute name
Monitor_ID

Column name
MONITOR_ID

Observed MBean attribute

Description
The MBean whose attribute is being monitored

Type
String

Names

Attribute name
Observed_MBean

Column name
OBSERVED_M

Observed Attribute attribute

Description
Name of the attribute that is monitored in the Observed MBean

Type
String

Names

Attribute name
Observed_Attribute

Column name
OBSERVED_A

Compare String attribute

Description
The string that is used in the comparison operation

Type
String

Names

Attribute name
Compare_String

Column name
COMPARE_ST

String Value attribute

Description
Value of the attribute that triggered the notification

Type
String

Names

Attribute name
String_Value

Column name
STRING_VAL

Notification Time Stamp attribute

Description
Time that the notification was triggered

Type
Time

Names

Attribute name
Notification_Time_Stamp

Column name
NOTIFICAT0

Notification Message attribute

Description
The message in the notification

Type
String

Names

Attribute name
Notification_Message

Column name
NOTIFICAT1

SNMP Event attribute groups

SNMP event attribute groups are used to receive traps and informs. These attribute groups are event-based attribute groups

The following list contains information about each attribute in the SNMP Event Attribute Groups:

Note: You can change the default display name of these attributes. These display names are distinct from the internal ID for each attribute.

Enterprise_OID
The enterprise OID that generated the trap.

Source_Address
Host name or IP address of the SNMP agent that sent the trap.

Generic_Trap
Generic trap number that is extracted from the received trap. Possible values:

```
0 ColdStart
1 WarmStart
2 LinkDown
3 LinkUp
4 Authentication Failure
5 EGPNeighborLoss
```

Specific_Trap
Enterprise-specific trap number that is extracted from the received trap. Applies only when Generic_Trap = 6.

Alert_Name
Trap name as specified in the definition in the trap configuration file.

Category
Trap category as specified in the definition in the trap configuration file.

Description
Trap description as specified in the definition in the trap configuration file. The maximum description length is 256 characters.

Enterprise_Name
Trap Enterprise name as specified in the trap configuration file and determined through the trap object identifier.

Source_Status
Status of the agent that originated the trap after the trap is sent as specified in the trap definition in the trap configuration file.

Source_Type
Type of the agent that originated the trap as specified in the trap definition in the trap configuration file.

Event_Variables
Variable binding (VarBind) data that is received in the trap protocol data unit (PDU). The string is constructed as:

```
{OID[type]=value}{OID[type]=value}{oid[type]=value}...
```

Where:

`oid`
MIB variable object identifier

`type`
SMI data type

`value`
Variable value

`{}`
Each triplet is surrounded by braces ({}).

Note: The attributes Alert Name, Category, Description, Enterprise_Name, Source_Status, and Source_Type provide more information. In the **SNMP MIB Browser** window, select the **Include attributes that show information defined in the trap configuration file** check box to include these attributes.

JMX Event attribute groups

JMX event attribute groups are used to receive notifications from an MBean server.

These attribute groups are non-event based attribute groups and are generated with the following attributes that can be edited by the agent developer.

The following list contains information about each attribute in the JMX Event Attribute Groups:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The local time at the agent when the data was collected

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

Type attribute

Description
The notification type

Type
String

Names

Attribute name
Type

Column name
TYPE

Source attribute

Description
The MBean that caused the notification to be sent

Type
String

Names

Attribute name
Source

Column name
SOURCE

Sequence Number attribute

Description
The sequence number from the notification object

Type
String

Names

Attribute name
Sequence_Number

Column name
SEQUENCE_N

Message attribute

Description
The notification message

Type
String

Names

Attribute name
Message

Column name
MESSAGE

User Data attribute

Description
The user data object from the notification

Type
String

Names

Attribute name
User_Data

Column name
USER_DATA

Ping attribute group

The Ping attribute group contains the results of ICMP pings that are sent to lists of devices.

The following list contains information about each attribute in the Ping Attribute Group:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent.

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The time that is collected from the agent system when the data row was built and sent from the agent to the Tivoli Enterprise Monitoring Server. Or stored for historical purposes. It represents the local time zone of the agent system.

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

Address attribute - This attribute is a key attribute

Description
The IP address of the host that is monitored.

Type
String with enumerated value. The value UNKNOWN_ADDRESS is displayed if the IP address is unknown. The warehouse and queries return 0.0.0.0 for this enumeration. Any other IP address values are displayed as is.

Names

Attribute name
Address

Column name
PNGADDR

Device Entry attribute - This attribute is a key attribute

Description
The entry in the device list file for this node.

Type
String

Names

Attribute name
Device_Entry

Column name
PINGDEVC

Current Response Time attribute

Description
The current network response time for ICMP requests for the managed node in milliseconds.

Type
Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the numbers. The defined values are `TIMEOUT(-1)` and `SEND_FAILURE(-2)`. Any other values show the numeric value.

Names

Attribute name
Current_Response_Time

Column name
PINGRSTM

Name attribute

Description
The host name of the managed node. If the node address cannot be resolved through DNS, then the dotted decimal IP address is shown.

Type

String with enumerated value. The value UNKNOWN_HOSTNAME is displayed if the host name is unknown. The warehouse and queries return 0.0.0.0 for this enumeration. Any other host name values are displayed as is.

Names

Attribute name

Name

Column name

PNGNAME

Node Description attribute

Description

The description of the managed node.

Type

String

Names

Attribute name

Node_Description

Column name

PNGDESC

Node Status attribute

Description

The current operating status of the managed node.

Type

Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the numbers. The defined values are `INVALID(-2)`, `UNKNOWN(-1)`, `INACTIVE(0)`, and `ACTIVE(1)`.

Names

Attribute name

Node_Status

Column name

PNGSTAT

Node Type attribute

Description

The type of the managed node. If the node is online, it is an IP Node. If it is offline, the type is Unknown.

Type

Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the numbers. The defined values are `UNKNOWN(0)` and `IP NODE(1)`.

Names

Attribute name

Node_Type

Column name

PNGTYPE

Status Timestamp

Description
The date and time the node was last checked.

Type
Time

Names

Attribute name
Status_Timestamp

Column name
PNGTMSP

HTTP attribute groups

The two HTTP attribute groups, `Managed URLs` and `URL Objects`, are used to receive information from URLs and the objects within theses URLs.

For information about the syntax that is used in the Managed URLs and URL Objects tables, see ("Specific fields for HTTP attributes" on page 1270).

Managed URLs

The following list contains information about each attribute in the Managed URL Attribute Group:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The local time at the agent when the data was collected

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

URL attribute - This attribute is a key attribute

Description
The URL that is being monitored.

Type
String

Names

Attribute name
URL

Column name
HTTPURL

Response Time attribute

Description
The amount of time it took to download the response in milliseconds.

Type
Integer with enumerated value. The string is displayed in the Tivoli Enterprise Portal, the warehouse, and queries return the number. The defined value is `TIMEOUT (-1)`.

Names

Attribute name
Response_Time

Column name
HTTPURL

Page Size attribute

Description
The size of the page that is returned by the HTTP request.

Type
Integer with enumerated value. The string is displayed in the Tivoli Enterprise Portal, the warehouse, and queries return the number. The defined value is `NO_RESPONSE_RECEIVED(-1)`.

Names

Attribute name
Page_Size

Column name
PAGESZ

Page Objects attribute

Description
The total number of objects that are associated with the monitored page.

Type
Integer with enumerated value. The string is displayed in the Tivoli Enterprise Portal, the warehouse, and queries return the number. The defined value is `NOT_COLLECTED(-1)`.

Names

Attribute name
Page_Objects

Column name
PGOBJS

Total Object Size attribute

Description
The size of the page that is returned by the HTTP request.

Type
Integer with enumerated value. The string is displayed in the Tivoli Enterprise Portal, the warehouse, and queries return the number. The defined value is `NOT_COLLECTED(-1)`.

Names

Attribute name
Total_Object_Size

Column name
TOTOSZ

Page Title attribute

Description
The page title of the received URL page.

Type
String

Names

Attribute name
Page_Title

Column name
PAGETTL

Server Type attribute

Description
The type of server that is used at the target URL website.

Type
String

Names

Attribute name
Server_Type

Column name
SRVTYP

Response Code attribute

Description
The response code of the HTTP request.

Type
Integer with enumerated value. The string is displayed in the Tivoli Enterprise Portal, the warehouse, and queries return the number. The defined value is `NO_RESPONSE_RECEIVED(-1)`.

Names

Attribute name
Response_Code

Column name
CODE

Status attribute

Description
The current managed URL status (OK or status description).

Type
String

Names

Attribute name
Status

Column name
STATUS

URL Alias attribute

Description
The user-specified alias for the URL.

Type
String

Names

Attribute name
URL_Alias

Column name
ALIAS

User Data attribute

Description
The user data that is specified with the URL.

Type
String

Names

Attribute name
User_Data

Column name
USER

URL Objects

The following list contains information about each attribute in the URL Objects Attribute Group:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The local time at the agent when the data was collected

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

URL attribute - This attribute is a key attribute

Description
The URL that is being monitored.

Type
String

Names

Attribute name
URL

Column name
HTTPURL

Object Name attribute

Description
The name of the page object within the target URL.

Type
String

Names

Attribute name
Object_Name

Column name
ONAME

Object Size attribute

Description
The size (bytes) of the page object within the target URL.

Type
Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the numbers. The defined values are `NOT_COLLECTED (-1)`, `OBJECT_NOT_FOUND (-2)`. Any other values show the numeric value.

Names

Attribute name
Object_Size

Column name
SIZE

Object Response Time attribute

Description
The amount of time it took to download the object in milliseconds.

Type
Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The warehouse and queries return the numbers. The defined values are `NOT_COLLECTED (-1)`, `NO_RESPONSE_RECEIVED (-2)`, `STATUS_CODE_ERROR (-3)`. Any other values show the numeric value.

Names

Attribute name
Object_Response_Time

Column name
ORTIME

Discovery attribute groups

An attribute group that represents the set of subnode instances that are defined for a subnode type

When you create a subnode type, an attribute group is created that represents the set of subnode instances that are defined for that subnode type. Each of these attribute groups includes the same set of attributes.

The following list contains information about each attribute in a Discovery attribute group. The name in bold text shows how the attribute is displayed in the Tivoli Enterprise Portal:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The time from the agent system when the data row was built and sent to the Tivoli Enterprise Monitoring Server (or stored for historical purposes). It represents the local time zone of the agent system.

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

Subnode MSN attribute

Description
The Managed System Name of the subnode agent.

Type
String

Names

Attribute name
Subnode_MSN

Column name
SN_MSN

Subnode Affinity attribute

Description
The affinity for the subnode agent.

Type
String

Names

Attribute name
Subnode_Affinity

Column name
SN_AFFIN

Subnode Type attribute

Description
The node type of this subnode.

Type
String

Names

Attribute name
Subnode_Type

Column name
SN_TYPE

Subnode Resource Name attribute

Description
The resource name of the subnode agent.

Type
String

Names

Attribute name
Subnode_Resource_Name

Column name
SN_RES

Subnode Version attribute

Description
The version of the subnode agent.

Type

Names

Attribute name
Subnode_Version

Column name
SN_VER

Take Action Status attribute group

The Take Action Status attribute group contains the status of actions that the agent processed.

This attribute group is event-based and contains information about each attribute in the Take Action Status attribute group:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent.

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The time that is collected from the agent system, when the data row was built and sent from the agent to the Tivoli Enterprise Monitoring Server. Or stored for historical purposes. It represents the local time zone of the agent system.

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

Action Name attribute

Description
The name of the action that was run

Type
String

Names

Attribute name
Action_Name

Column name
TSKNAME

Action Status attribute

Description
The status of the action.

Type
Integer with enumerated values. The values are: OK (0), NOT_APPLICABLE (1), GENERAL_ERROR (2), WARNING (3), NOT_RUNNING (4), DEPENDENT_NOT_RUNNING (5), ALREADY_RUNNING (6), PREREQ_NOT_RUNNING (7), TIMED_OUT (8), DOESNT_EXIST (9), UNKNOWN (10), DEPENDENT_STILL_RUNNING (11), INSUFFICIENT_USER_AUTHORITY (12)

Names

Attribute name
Action_Status

Column name
TSKSTAT

Action Application Return Code attribute

Description
The return code of the application the action started.

Type
Integer

Names

Attribute name
Action_App_Return_Code

Column name
TSKAPRC

Action Message attribute

Description
The message that is associated with the return code of the action.

Type
String

Names

Attribute name
Action_Message

Column name
TSKMSGE

Action Instance attribute

Description
The instance that is associated with the output produced by running the action. If the action is a system command, the instance is the line number of the output of the command.

Type
String

Names

Attribute name
Action_Instance

Column name
TSKINST

Action Results attribute

Description
The output that is produced by running the action.

Type
String

Names

Attribute name
Action_Results

Column name
TSKOUTP

Action Command attribute

Description
The command that was run by the action.

Type
String

Names

Attribute name
Action_Command

Column name
TSKCMND

Action Node attribute

Description
The node where the action ran.

Type
String

Names

Attribute name
Action_Node

Column name
TSKORGN

Action Subnode attribute

Description
The subnode where the action ran.

Type
String

Names

Attribute name
Action_Subnode

Column name
TSKSBND

Action ID attribute

Description
The ID of the action.

Type
Integer

Names

Attribute name
Action_ID

Column name
TSKID

Action Type attribute

Description
The type of the action.

Type
Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal, the warehouse, and queries return the numbers. The defined values are: UNKNOWN (0), AUTOMATION (1).

Names

Attribute name
Action_Type

Column name
TSKTYPE

Action Owner attribute

Description
The name of the situation or user that initiated the action.

Type
String

Names

Attribute name
Action_Owner

Column name
TSKOWNR

Log File Status attribute group

The Log File Status attribute group contains information that reflects the status of log files this agent is monitoring.

The Log File Status attribute group is included if you have a log attribute group and the agent is at the default minimum Tivoli Monitoring version of 6.2.1 or later. The Log File Status attribute group includes two attributes that are defined as 64-bit numbers so that they can handle large files. 64-bit numeric attribute support is provided by Tivoli Monitoring version 6.2.1 or later.

The following list contains information about each attribute in the Log File Status attribute group:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent.

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The value is the time that is collected from the agent system, when the data row was built and sent from the agent to the Tivoli Enterprise Monitoring Server. Or stored for historical purposes. It represents the local time zone of the agent system.

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

Table Name attribute - This attribute is a key attribute

Description
The name of the table in which this log is being monitored

Type
String

Names

Attribute name
Table_Name

Column name
TBLNAME

File Name attribute - This attribute is a key attribute

Description
The name of the file that is being monitored

Type
String

Names

Attribute name
File_Name

Column name
FILNAME

RegEx Pattern attribute - This attribute is a key attribute

Description
The regular expression pattern (if any) that caused this file to be monitored

Type
String

Names

Attribute name
RegEx_Pattern

Column name
REPATRN

File Type attribute

Description
The type of this file (regular file or pipe)

Type
Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The defined values are UNKNOWN(0), REGULAR FILE(1), PIPE(2)

Names

Attribute name
File_Type

Column name
FILTYPE

File Status attribute

Description
The status of the file that is being monitored

Type
Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. The defined values are: OK(0), PERMISSION DENIED(1), FILE DOES NOT EXIST(2), INTERRUPTED SYSTEM CALL(4), I/O ERROR(5), NO SUCH DEVICE(6), BAD FILE NUMBER(9), OUT OF MEMORY(12), ACCESS DENIED(13), RESOURCE BUSY(16), NOT A DIRECTORY(20), IS A DIRECTORY(21), INVALID ARGUMENT(22), FILE TABLE OVERFLOW(23), TOO MANY OPEN FILES(24), TEXT FILE BUSY(26), FILE TOO LARGE(27), NO SPACE LEFT ON DEVICE(28), ILLEGAL SEEK ON PIPE(29), READ-ONLY FILE SYSTEM(30), TOO MANY LINKS(31), BROKEN PIPE(32)

Names

Attribute name
File_Status

Column name
FILSTAT

Num Records Matched attribute

Description
The number of processed records from this log which matched one of the specified patterns

Type
Integer

Names

Attribute name
Num_Records_Matched

Column name
RECMTCH

Num Records Not Matched attribute

Description
The number of processed records sent to the UnmatchLog; did not match any patterns

Type
Integer

Names

Attribute name
Num_Records_Not_Matched

Column name
RECUNMT

Num Records Processed attribute

Description
The number of records that are processed from this log since agent start (including ones that are not matches/events)

Type
Integer

Names

Attribute name
Num_Records_Processed

Column name
RECPROC

Current File Position attribute

Description
The current position in bytes into the monitored file. Data up to this position is processed, data after this position is not processed. Not applicable to pipes.

Type
Integer

Names

Attribute name
Current_File_Position

Column name
OFFSET

Current® File Size attribute

Description
The current size of the monitored file. Not applicable to pipes.

Type
Integer

Names

Attribute name
Current_File_Size

Column name
FILESIZE

Last Modification Time attribute

Description
The time when the monitored file was last written to. Not applicable to pipes.

Type
Timestamp

Names

Attribute name
Last_Modification_Time

Column name
LASTMOD

Codepage attribute

Description
The language codepage of the monitored file

Type
String

Names

Attribute name
Codepage

Column name
CODEPG

Log File RegEx Statistics attribute group

The Log File RegEx Statistics attribute group contains information that shows the statistics of the log file regular expression search expressions.

Regular expressions can be used to filter records or to define records. This attribute group shows information about both types. When the Result Type attribute contains either `INCLUDE` or `EXCLUDE`, the filter is use to filter records. If the Result Type attribute contains `BEGIN` or `END`, the filter is used to define records. The CPU measurements are approximations that are based on the granularity of the data that is exposed by the operating system. These measurements can result in values of `0.00` when a regular expression takes a small time to evaluate. Use the CPU times to determine the relative cost of regular expressions and to optimize the behavior of specific regular expressions.

The Log File RegEx Statistics attribute group is included if you have a log attribute group and the agent is at Tivoli Monitoring version of 6.2.1 or later. The minimum Tivoli Monitoring Version is selected on the **Agent Information** page. For more information, see ("Naming and configuring the agent" on page 1149). The Log File RegEx Statistics attribute group includes attributes that are defined as 64-bit numbers so that they can handle long durations. Support for 64-bit numeric attributes is provided by Tivoli Monitoring version 6.2.1 or later.

The following list contains information about each attribute in the Log File RegEx Statistics attribute group:

Node attribute - This attribute is a key attribute

Description
The managed system name of the agent.

Type
String

Names

Attribute name
Node

Column name
ORIGINNODE

Timestamp attribute

Description
The local time at the agent when the data was collected.

Type
Time

Names

Attribute name
Timestamp

Column name
TIMESTAMP

Table Name attribute - This attribute is a key attribute

Description
The name of the log file attribute group.

Type
String

Names

Attribute name
Table_Name

Column name
TBLNAME

Attribute Name attribute - This attribute is a key attribute

Description
The name of the attribute to which this filter is applied.

Type
String

Names

Attribute name
Attribute_Name

Column name
ATRNAME

Filter Number

Description
The sequence number, starting at zero, of the filter that is being used for the attribute.

Type
Integer (Numeric Property)

Names

Attribute name
Filter_Number

Column name
FLTRNUM

Result Type attribute

Description
The result type can be INCLUDE or EXCLUDE to accept or reject the attribute if the filter matches. The result type can be BEGIN or END to specify the start or end of a record for multi-line records.

Type
Integer with enumerated values. The strings are displayed in the Tivoli Enterprise Portal. If the filter is used to filter records, the defined values are `INCLUDE(1)` or `EXCLUDE(2)`. If the filter is used to define records, the defined values are `BEGIN(3)` or `END(4)`.

Names

Attribute name
Result_Type

Column name
RSTTYPE

Average Processor Time attribute

Description
The average number of processor seconds used to process the filter for this attribute. The average processor time is the total processor seconds divided by the filter count.

Type
Integer (Gauge)

Names

Attribute name
Average_Processor_Time

Column name
CPUTAVG

Processor Time attribute

Description
The total number of processor seconds used to process the filter for this attribute. The processor time is cumulative and is truncated, not rounded. Similar to the Linux `/proc/<pid>/task/`*`thread`*`/stat file`.

Type
Integer (Counter)

Names

Attribute name
Processor_Time

Column name
CPUTIME

Max Processor Time attribute

Description
The maximum number of processor seconds used for a single filter processing. It is possible that the maximum is zero if the filter was never used or if each of the filter processing took less than 0.01 seconds.

Type
Integer (Gauge)

Names

Attribute name
Max_Processor_Time

Column name
CPUTMAX

Min Processor Time attribute

Description
The minimum number of processor seconds used for a single filter processing. It is possible that the minimum is zero if a filter processing took less than 0.01 seconds.

Type
Integer (Gauge)

Names

Attribute name
Min_Processor_Time

Column name
CPUTMIN

Filter Count attribute

Description
The number of times the filter is run. Used with the total processor time to compute the average processor time.

Type
Integer (Counter)

Names

Attribute name
Filter_Count

Column name
COUNT

Filter Count Matched attribute

Description
The number of times the filter is run and the attribute matched.

Type
Integer (Counter)

Names

Attribute name
Filter_Count_Matched

Column name
COUNTMA

Filter Count Unmatched attribute

Description
The number of times the filter is run and the attribute did not match.

Type
Integer (Counter)

Names

Attribute name
Filter_Count_Unmatched

Column name
COUNTUN

RegEx Pattern attribute - This attribute is a key attribute

Description
The regular expression that is used for the match.

Type
String

Names

Attribute name
RegEx_Pattern

Column name
REGXPAT

Last Matched Time attribute

Description
The last time the filter was used and the result matched.

Type
Time

Names

Attribute name
Last_Matched_Time

Column name
LASTMAT

Last Unmatched Time attribute

Description
The last time the filter was used and the result was unmatched.

Type
Time

Names

Attribute name
Last_Unmatched_Time

Column name
LASTUMA

Creating application support extensions for existing agents

For the IBM Tivoli Monitoring environment, you can build an installable package to distribute custom workspaces, situations, queries, and Take Action commands that you created, as an application support extension for an existing agent.

Before you begin

For more information about how to create custom situations, workspaces, Take Action commands, and queries, see ("Creating workspaces, Take Action commands, and situations" on page 1337).

About this task

Important: This task is not how you add application support to an agent that you are building. To add application support to an agent that you are building see ("Importing application support files" on page 1371).

Procedure

1. From the Agent Builder, select **File** > **New** > **Other**.
2. Select **Agent Builder Application Support Extension** under **Agent Builder**.
3. Click **Next** to get to the welcome page for the **IBM Tivoli Monitoring Application Support Extension** wizard.
4. Click **Next** on the welcome page.
5. Enter a name for the project and click **Finish**

Creating an Application Support Extension project

Create an Application Support Extension project by using Agent builder.

Procedure

1. From the Agent Builder, select **File** > **New** > **Other**.
2. Select **Agent Builder Application Support Extension** under **Agent Builder**.
3. Click **Next** to get to the welcome page for the **IBM Tivoli Monitoring Application Support Extension Wizard**.
4. Click **Next** on the welcome page.
5. Enter a name for the project and click **Finish**

Adding support files to a project

Add your support files to an Application Support Extension project

Before you begin

Create an Application Support Extension project. For more information, see "Creating an Application Support Extension project" on page 1438.

Procedure

1. Right-click an Application Support Extension project and select **IBM Tivoli** > **Import Application Support Extensions**
2. In the **Import Information** window, select the name of the host where the Tivoli Enterprise Portal Server is located or click **Add** to add one.
3. In the **Application** field, enter the agent product code.
4. Enter the affinity of the agent for which you are creating custom application support.

   The agent affinity is a Tivoli Monitoring internal identifier that associates workspaces, queries, and other items, with the agent. It must be unique in the Tivoli Monitoring installation. Click **Browse** to open the **Node Types** window and select this information from a list rather than typing it.
5. When you are satisfied with the import information, click **Finish**.
6. In the **Situations** window, select the situations that you want to import from the Available Situations list.

   Click **<<** to add them to the Selected Situations list and click **OK**. A new folder is created under the project, and it contains the necessary files to install the workspaces, situations, and queries.
7. In the **Queries** window, select the queries that you want to import from the Available Queries list.

   Click **<<** to add them to the Selected Queries list and click **OK**.
8. In the **Take Actions** window, Choose the Take Action commands that you want to import from the Available Take Actions list.

   Click **<<** to add them to the Selected Take Actions list and click **OK**. The support files for the agent are put in the project under the appropriate folder.

What to do next

You can repeat this process for as many different agents as you want. The Agent Builder creates a single installation image from all of the support files in the Application Support Extension project.

Generating the Application Support Extension installation image

Generate an Application Support Extension installation image.

Procedure

1. Right-click on the Application Support Extension project and select **IBM Tivoli** > **Create Application Support Extension Install Image**.
2. In the **Application Support Extension Information** window, enter the directory where the image is to be placed.
3. Your Application Support Extension must have its own product code. Enter the registered product code for your new agent. You can use one of the product codes that are reserved for use with the Agent Builder. The allowed values are `K00-K99`, `K{0-2}{A-Z}`, and `K{4-9}{A-Z}`.

   Note: These values are for internal use only and are not intended for agents that are to be shared or sold. If you are creating an agent to be shared with others, you must send a note to toolkit@us.ibm.com to reserve a product code. The request for a product code must include a description of the agent to be built. A product code is then assigned, registered, and returned to you. When you receive the three-letter product code, you are told how to enable the Agent Builder to use the assigned product code.
4. Enter the name of the Application Support Extension.
5. Enter a description of the Application Support Extension.
6. Enter a version for the Application Support Extension in the VVRRMMFF format where `vv` = version number; `rr` = release number; `mm` = modification number (fix pack number); and `ff` = interim fix number.
7. Click **Finish**.

Installing your Application Support Extension

Install your Application Support Extension

Procedure

1. Transfer your image to your Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server servers.
2. To install the Tivoli Enterprise Monitoring Server support, run one of the following commands:
   - On Windows: `install`*KXX*`TEMSSupport.bat`
   - On UNIX: `install`*KXX*`TEMSSupport.sh`

   The format for the command is as follows:

   ```
   installKXXTEMSSupport[.bat | .sh] <ITM Install Directory> [-s tems_host]
    [-u tems_user]  \[-p tems_password]
   ```
3. To install the Tivoli Enterprise Portal Server support, run one of the following commands:
   - On Windows: `install`*KXX*`TEPSSupport.bat`
   - On UNIX: `install`*KXX*`TEPSSupport.sh`

   The format for the command is as follows:

   ```
    installKXXTEPSSupport[.bat | .sh] <ITM Install Directory> [-r]
   ```

   where `-r` indicates that the Tivoli Enterprise Portal Server must be restarted after installation

Converting a Solution Install Project to an Application Support Extension project

Convert an existing **Solution Install Project** to an Application Support Extension project

About this task

If you have an existing **Solution Install Project** that you want to convert to an Application Support Extension project, complete the following steps:

Note: In the **Solution Install Project** only Support files are migrated.

Procedure

1. Right-click on the **Solution Install project** and select **IBM Tivoli** > **Convert Solution Install Project**.
2. Enter the name of a new Application Support Extension project or select an existing one from the list
3. Click **Finish**.

Cognos data model generation

Agent Builder can generate a Cognos data model for each agent. Use the data model to import agent information into the Cognos Framework Manager for report creation.

This Cognos data model can be opened and viewed in the Framework Manager, which builds a model package to be published into Tivoli Common Reporting. The data model can also be customized or modified within the Framework Manager before publication.

When a report is created, Agent Builder also allows for a final report package to be imported into the Agent Builder project. This feature enables future agent projects to be generated with the reports that are already part of the agent package. The reports that are packaged as part of the agent installation image can be imported into Tivoli Common Reporting in your production environment.

Note: In this documentation, note the following convention:

- K*xx* or k*xx* refers to the product code given to the agent, for example, k99.
- *dbType* refers to the database that is being used by the Tivoli Data Warehouse, for example, DB2.

Prerequisites to generating a Cognos data model

Complete these tasks before you generate a Cognos data model

About this task

Note:

- These steps must be completed only one time, as all future data models generated with Agent Builder will use this environment.
- It is advisable to create an isolated development environment for agent testing and report creation.

Procedure

1. Install and configure a ("Tivoli Data Warehouse" on page 1441).
2. Create tables and Procedures in the Tivoli Data Warehouse.
   a) "Create tables and Procedures in the Tivoli Data Warehouse" on page 1441.
   b) "Populating the Tivoli Data Warehouse with the Tivoli Reporting and Analytics Model" on page 1443.
3. Install and configure ("Tivoli Common Reporting" on page 1444).
4. Install and configure the ("Framework Manager" on page 1444).

Tivoli Data Warehouse

About Tivoli Data Warehouse.

To create reports, you need a Tivoli Data Warehouse, a Warehouse Proxy agent, and a Summarization and Pruning agent, to be installed and configured in your environment. For more information, see the *IBM Tivoli Monitoring Installation and Setup Guide*.

Create tables and Procedures in the Tivoli Data Warehouse

Create or alter the `ManagedSystem` Table and Stored Procedure in the Tivoli Data Warehouse

About this task

The generated Cognos data model includes a `ManagedSystem` table which is used to define a `ManagedSystem` dimension. The `ManagedSystem` dimension allows reports to be created that can correlate managed systems. For example, if the agent is a subnode agent, the dimension can be used to determine the subnodes that exist for a specific agent instance.

The `ManagedSystem` table is not created by the Tivoli Data Warehouse. Therefore, when an agent is generated in Agent Builder, SQL scripts are generated for each database platform that will:

- Create the `ManagedSystem` table. Use this script if the table does not exist in the Tivoli Data Warehouse.
- Edit the `ManagedSystem` table. Use this script if the table exists in the Tivoli Data Warehouse. Other reporting products can create the `ManagedSystem` table, but they do not create it with all of the required columns.
- Create a stored procedure that populates the `ManagedSystem` table from tables in the Tivoli Data Warehouse.

Run these scripts one time only.

Running DB2 Scripts to Create tables and Procedures in the Tivoli Data Warehouse

For a DB2 database, use these scripts to create tables in the Tivoli Data Warehouse

Before you begin

The scripts for DB2 are in the following directory:

```
reports/db2/Kxx/reports/cognos_reports/itmkxx/db_scripts
```

Procedure

1. The generated scripts (`create_table.sql`, `alter_table.sql`, and `create_procedure.sql`) all use *`itmuser`* as the Tivoli Data Warehouse user ID. If *`itmuser`* is not the Tivoli Data Warehouse user ID in your environment, change all occurrences of *`itmuser`* to the correct user ID.
2. Connect to the Tivoli Data Warehouse as the Tivoli Data Warehouse User:

   ```
   db2 connect to <Tivoli Data Warehouse alias name> user
   <Tivoli Data Warehouse user id> using  <password>
   ```

3. Determine whether the ManagedSystem table exists:

   ```
   db2 "select count(*) from sysibm.systables where name = 'MANAGEDSYSTEM'
   and creator=upper ('<Tivoli Data Warehouse user id>')"
   ```

4. Create or alter the table.

   - If the query returns 1, the table exists. Run the alter script:

     ```
     db2 -tvf alter_table.sql
     ```

- If the query returns 0, the table does not exist. Run the create script:

  ```
  db2 -tvf create_table.sql
  ```

5. Run the script to create the stored procedure:

   ```
   db2 -td@ -f create_procedure.sql
   ```

Running Oracle Scripts to Create tables and Procedures in the Tivoli Data Warehouse
For an Oracle database, use these scripts to create tables in the Tivoli Data Warehouse

Before you begin

The scripts for Oracle are in the following directory:

```
reports/oracle/Kxx/reports/cognos_reports/itmkxx/db_scripts
```

Procedure

1. The generated scripts (`create_table.sql`, `alter_table.sql`, and `create_procedure.sql`) all use *itmuser* as the Tivoli Data Warehouse user ID. If *itmuser* is not the Tivoli Data Warehouse user ID in your environment, change all occurrences of *itmuser* to the correct user ID.
2. Start `sqlplus`:

   ```
   sqlplus <IBM Tivoli Monitoring user ID>/<password>@
   <Tivoli Data Warehouse SID>
   ```

3. Determine whether the ManagedSystem table exists:

   ```
   select count(*) from user_tables where table_name = 'MANAGEDSYSTEM';
   ```

4. Create or alter the table.
   - If the query returns 1, the table exists. Run the alter script:

     ```
     @<path to alter_table.sql>;
     ```

   - If the query returns 0, the table does not exist. Run the create script:

     ```
     @<path to create_table.sql>;
     ```

5. Run the script to create the stored procedure:

   ```
   @<path to create_procedure.sql>;
   ```

Running SQL Server 2005 and 2008 Scripts to Create tables and Procedures in the Tivoli Data Warehouse

Before you begin

The scripts for SQL Server are in the following directory:

```
reports/mssql/Kxx/reports/cognos_reports/itmkxx/db_scripts
```

Procedure

1. The generated scripts (`create_table.sql`, `alter_table.sql`, and `create_procedure.sql`) all use *itmuser* as the Tivoli Data Warehouse user ID. If *itmuser* is not the Tivoli Data Warehouse user ID in your environment, change all occurrences of *itmuser* to the correct user ID.
2. Determine whether the `ManagedSystem` table exists:

```
osql -S <Server> -U <Tivoli Data Warehouse user ID> -P <password> -d
<Tivoli Data Warehouse database name> -Q "Select count(*)
from INFORMATION_SCHEMA.TABLES where table_name = 'ManagedSystem'"
```

3. Create or alter the table.

   - If the query returns 1, the table exists. Run the alter script:

     ```
     osql -S <Server> -U <Tivoli Data Warehouse user ID> -P <password> -d
     <Tivoli Data Warehouse database name> -I -n -i <path to alter_table.sql>
     ```

   - If the query returns 0, the table does not exist. Run the create script:

     ```
     osql -S <Server> -U <Tivoli Data Warehouse user ID> -P <password> -d
     <Tivoli Data Warehouse database name> -I -n -i <path to create_table.sql>
     ```

4. Run the script to create the stored procedure:

   ```
   osql -S <Server> -U <Tivoli Data Warehouse user ID> -P
   <password> -d <Tivoli Data Warehouse database name>
   -I -n -i <path to create_procedure.sql>
   ```

Populating the Tivoli Data Warehouse with the Tivoli Reporting and Analytics Model

Use provided database scripts to populate the Tivoli Data Warehouse

About this task

Tivoli Reporting and Analytics Model (TRAM) contains the base-set of knowledge that is common to all reporting packages. TRAM is installed by a set of scripts unique to each database. The necessary scripts for populating each supported database are included in the agent installation image, within the reports directory. Use the following procedure to create Tivoli Reporting and Analytics Model Common Dimensions in Tivoli Data Warehouse.

Procedure

1. Browse to the Tivoli Reporting and Analytics Model database scripts.
2. Extract the agent package.

   - On Windows systems, agent package is `k`*`xx`*`.zip`.
   - On Linux and UNIX systems, agent package is `k`*`xx`*`.tgz`.

3. Go to the appropriate database scripts.

   - DB2 scripts are in the Agent package at:

     ```
     reports/db2/Kxx/reports/cognos_reports/itmkxx/db_scripts
     ```

   - Oracle scripts are in the Agent package at:

     ```
     reports/oracle/Kxx/reports/cognos_reports/itmkxx/db_scripts
     ```

   - Microsoft SQL Server scripts are in the Agent package at:

     ```
     reports/mssql/Kxx/reports/cognos_reports/itmkxx/db_scripts
     ```

4. Run the database scripts to generate the common dimensions within the Tivoli Data Warehouse. Each script set provides a readme file for usage instructions.
5. Verify that the scripts added the following tables to the Tivoli Data Warehouse:

   ```
   "Computer System", WEEKDAY_LOOKUP, MONTH_LOOKUP, TIMEZONE_DIMENSION, TIME_DIMENSION
   ```

Tivoli Common Reporting
Tivoli Common Reporting contains the Cognos Business Intelligence engine, which contains elements to assist with the creation of agent reports.

Tivoli Common Reporting must be installed and configured with a data source that connects to the Tivoli Data Warehouse.

Installing Tivoli Common Reporting

You must install Tivoli Common Reporting. Versions 1.3, 2.1, 2.1.1 or later are supported. For information about installing Tivoli Common Reporting, see Installing Tivoli Common Reporting.

Configuring Tivoli Common Reporting

You must configure Tivoli Common Reporting. For information about configuring Tivoli Common Reporting, see Configuring IBM Tivoli Common Reporting.

Create a data source between the Tivoli Data Warehouse and Tivoli Common Reporting. For more information, see Configuring database connection. Click the appropriate database type. Note the name that is given to the data source. The default is **TDW**.

Note: The data source name must match the name in the **Data source** field of the **Cognos Information** page. For more information about the **Cognos Information** page, see "Cognos information" on page 1164.

Framework Manager
Framework Manager is an application that ships with the Tivoli Common Reporting application, but must be installed and configured separately.

Framework Manager is used to view and modify data models and to publish data models to Tivoli Common Reporting

Installing Framework Manager

You must install Framework Manager. Versions 8.4, 8.4.1 or later are supported.

The Framework Manager ships with Tivoli Common Reporting, but must be manually installed. Tivoli Common Reporting 1.3 ships with Framework Manager 8.4. Tivoli Common Reporting 2.1 and 2.1.1 ships with Framework Manager 8.4.1. For information about installing Framework Manager, see Installing Framework Manager in the *Tivoli Common Reporting User's Guide*.

Configuring Framework Manager

You must configure Framework Manager. For information about configuring Framework Manager, see Configuring Framework Manager in the *Tivoli Common Reporting User's Guide*.

Creating reports

Use the Framework Manager to publish the agent model, and Report Studio to begin creating reports.

Before you begin

When the agent is completed, it must be installed into the Tivoli Monitoring environment. In addition, historical collection for the agent must be configured and the agent be run for at least one warehouse upload interval. Summarization must be configured, and the summarization setting choices that are made in Tivoli Monitoring must be identical to the summarization choices made in the Agent Builder. The Summarization and Pruning agent must run at least one time after the agent's data is uploaded to the warehouse.

1. Install, configure, and start your agent.
2. Create and distribute to the agent a historical collection for each attribute group you want to create a report for.

Note: The warehouse upload interval defaults to daily. However, you might want to shorten this interval.

For information about configuring historical collection, see Managing historical data in the *IBM Tivoli Monitoring Administrator's Guide*.

3. In Tivoli Monitoring, configure summarization for all of the attribute groups you created historical collections for in Step 2.

   Note: When you configure historical collection and summarization, you must wait enough time for data to end up in the summary tables.

   Note: By default, the Summarization and Pruning agent is configured to run one time a day at 2 a.m. You might want to change this setting. For example, you can configure it to run hourly. For information about configuring the Tivoli Data Warehouse, see Setting up data warehousing in the *IBM Tivoli Monitoring Installation and Setup Guide*.

About this task

Generating an agent in Agent Builder creates an entire Framework Manager project, which includes the data model and the Framework Manager project file. Framework Manager can open the project file directly, which opens the data model for modification, customization, or publication.

Procedure

Note: The generated data model for the agent contains all of the summary time dimensions for each attribute group: hourly, daily, weekly, monthly, quarterly, and yearly. The dimensions exist only in the Tivoli Data Warehouse for the agent if summarization and pruning is configured for the agent. And also if the dimensions are selected, and if the Summarization and Pruning agent, created, and populated the tables. Reports can be defined and published into Tivoli Common Reporting that use dimensions that do not exist. Such reports do not function until the summary tables are created by the Summarization and Pruning agent.

1. Open the Agent Data Model in Framework Manager:

   a) Open the Framework Manager.

   b) From the **Welcome** page, click **Open a project**.

      Tip: If you are in Framework Manager, click **Open** from the **File** menu.

   c) Browse to the Agent data model.

      - For DB2:

        ```
        reports/db2/Kxx/model/
        ```

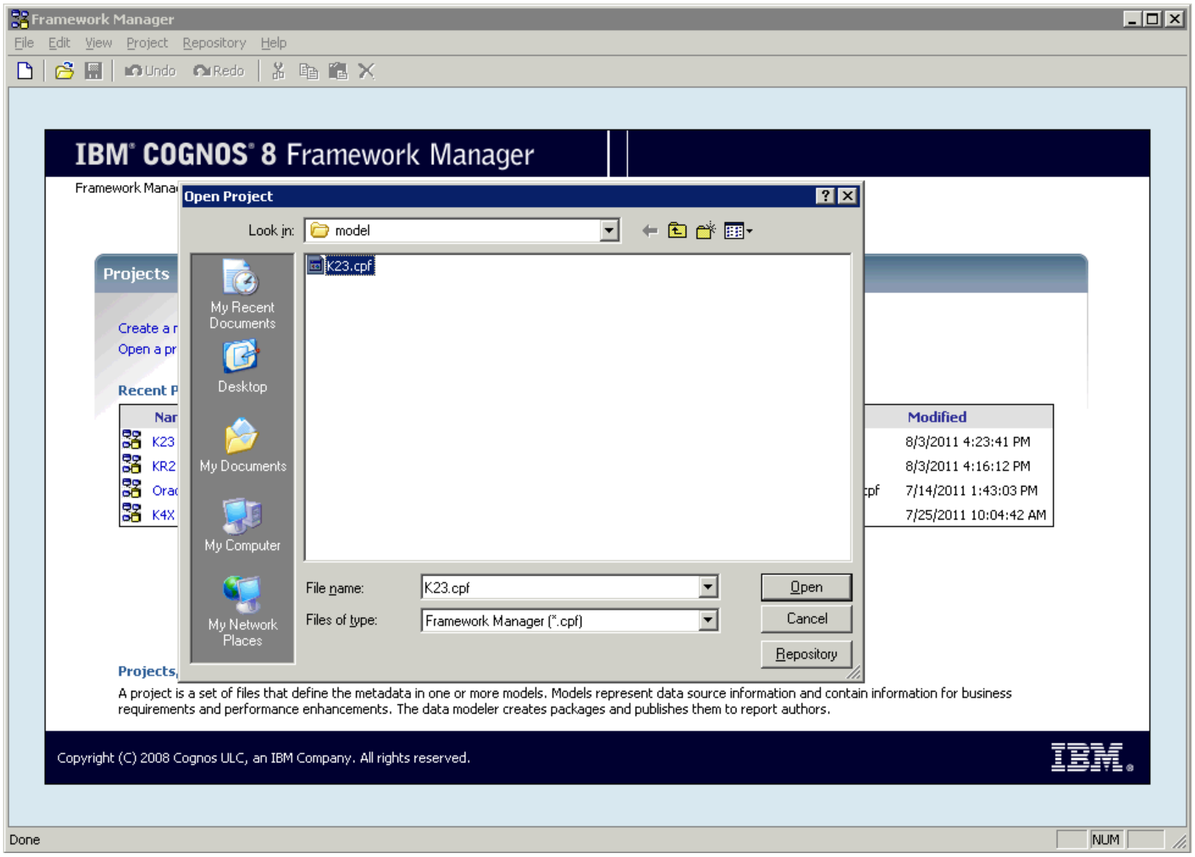
      - For Oracle:

        ```
        reports/oracle/Kxx/model/
        ```

      - For Microsoft SQL Server:

        ```
        reports/mssql/Kxx/model/
        ```

   d) Select the agent project file, K*xx*`.cpf`.



Look in: model

File name: K23.cpf

Files of type: Framework Manager (*.cpf)

Nar		Modified
K23		8/3/2011 4:23:41 PM
KR2		8/3/2011 4:16:12 PM
Orac	:pf	7/14/2011 1:43:03 PM
K4X		7/25/2011 10:04:42 AM

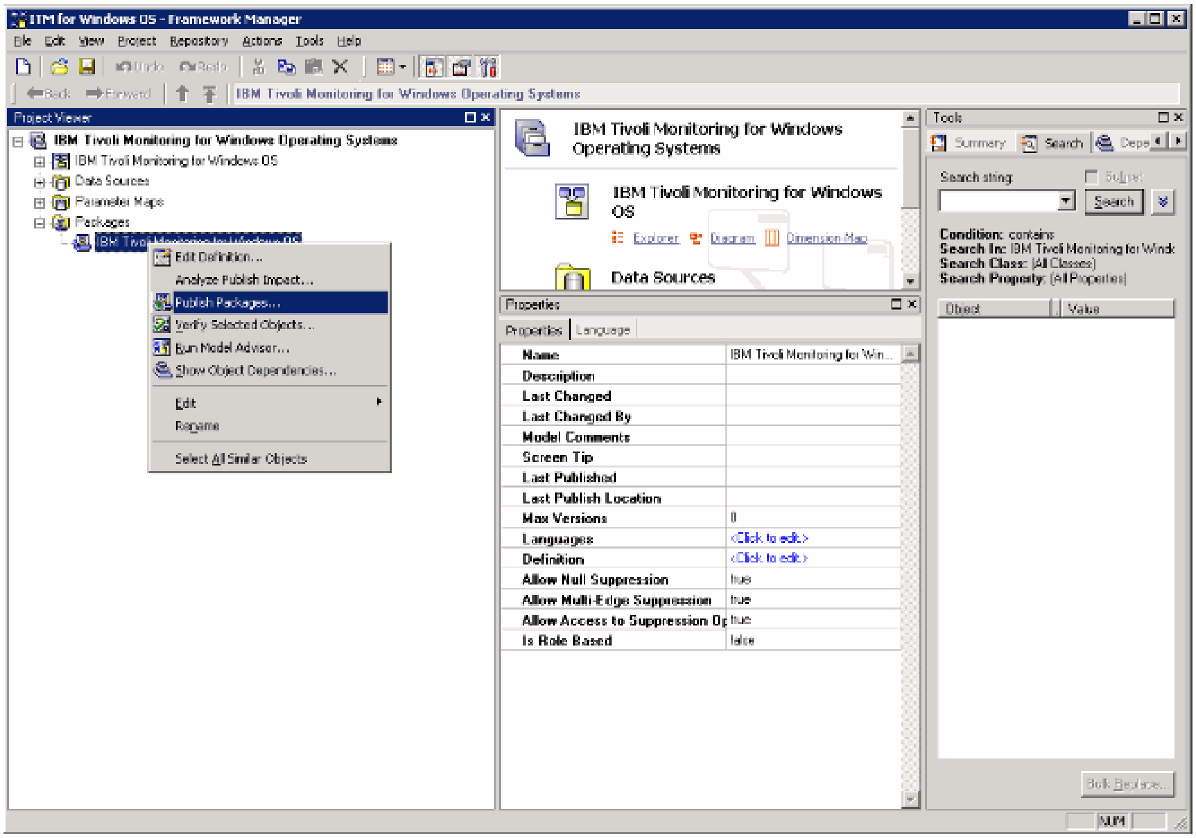
A project is a set of files that define the metadata in one or more models. Models represent data source information and contain information for business requirements and performance enhancements. The data modeler creates packages and publishes them to report authors.

Copyright (C) 2008 Cognos ULC, an IBM Company. All rights reserved.

Figure 80. Selecting agent project file

Note: When an agent project is opened in Framework Manager, the agent name is listed under the Recent Projects.

2. Populate the Managed System Table. For more information, see "Populating the ManagedSystem Table" on page 1450
3. Use the Framework Manager to publish the Agent Model to Tivoli Common Reporting
   a) Open the Framework Manager.
   b) Open the Agent project.
   c) Expand **Packages** in the navigation tree.
   d) Right-click the agent package and select **Publish Packages**.

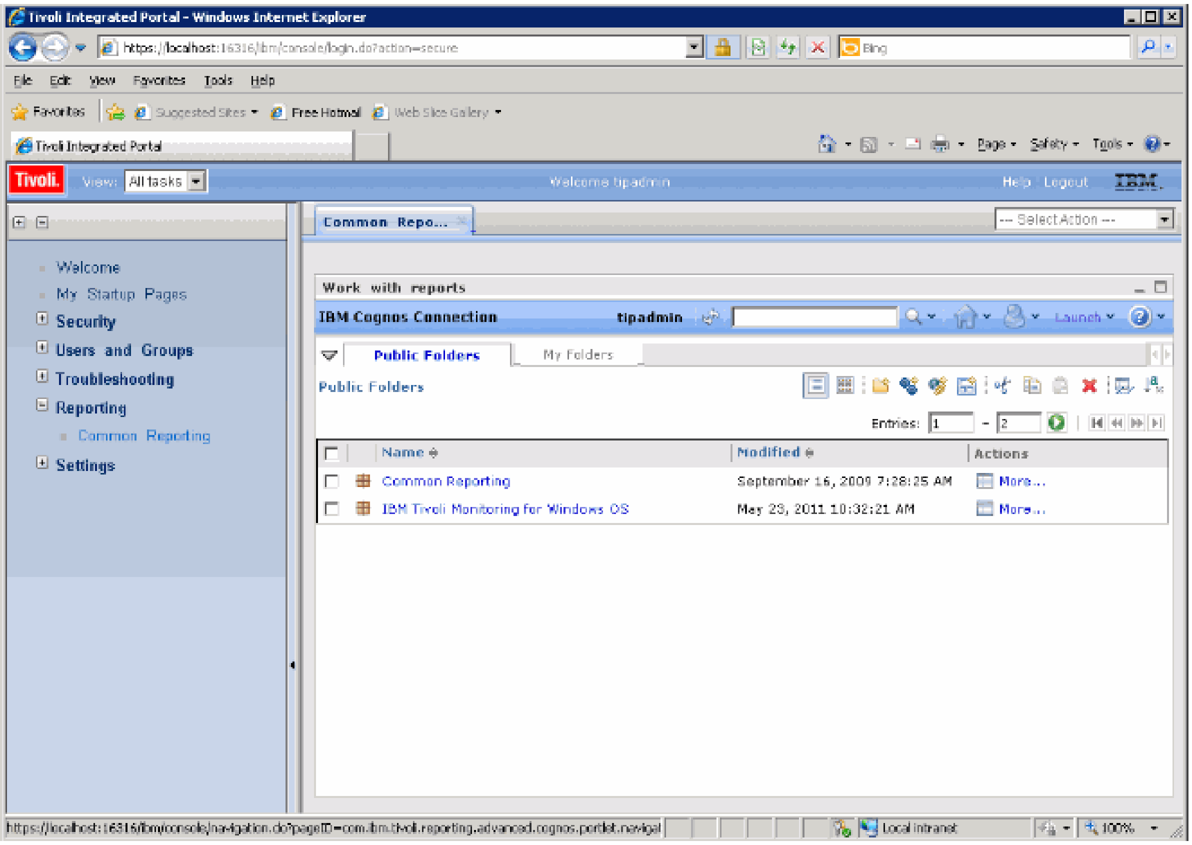

ITM for Windows OS - Framework Manager
File Edit View Project Repository Actions Tools Help
Undo Redo
Back Forward
IBM Tivoli Monitoring for Windows Operating Systems
Project Viewer
IBM Tivoli Monitoring for Windows Operating Systems
IBM Tivoli Monitoring for Windows OS
Data Sources
Parameter Maps
Packages
IBM Tivoli Monitoring for Windows OS
Edit Definition...
Analyze Publish Impact...
Publish Packages...
Verify Selected Objects...
Run Model Advisor...
Show Object Dependencies...
Edit
Rename
Select All Similar Objects
IBM Tivoli Monitoring for Windows Operating Systems
IBM Tivoli Monitoring for Windows OS
Explorer
Diagram
Dimension Map
Data Sources
Properties
Properties
Language
Tools
Summary
Search
Depe
Search string
Subtree
Search
Object
Value
Bulk Replace...
NUM

Name	IBM Tivoli Monitoring for Win...
Description	
Last Changed	
Last Changed By	
Model Comments	
Screen Tip	
Last Published	
Last Publish Location	
Max Versions	0
Languages	<Click to edit>
Definition	<Click to edit>
Allow Null Suppression	true
Allow Multi-Edge Suppression	true
Allow Access to Suppression Op	true
Is Role Based	false

Condition: contains
Search In: IBM Tivoli Monitoring for Wind
Search Class: (All Classes)
Search Property: (All Properties)

Figure 81. Selecting Publish Packages

4. Use Report Studio to create new reports or templates.
   a) Log on to Tivoli Common Reporting.
   b) Browse to Public Folders, expand **Reporting** in the navigation panel, and select **Common Reporting**.



	Name	Modified	Actions
□	Common Reporting	September 16, 2009 7:28:25 AM	More...
□	IBM Tivoli Monitoring for Windows OS	May 23, 2011 10:32:21 AM	More...

Figure 82. Selecting Common Reporting

c) Select your Tivoli Monitoring agent from the list provided.

d) Open the report creation tool, by clicking the Launch menu and selecting **Report Studio** or **Query Studio**.

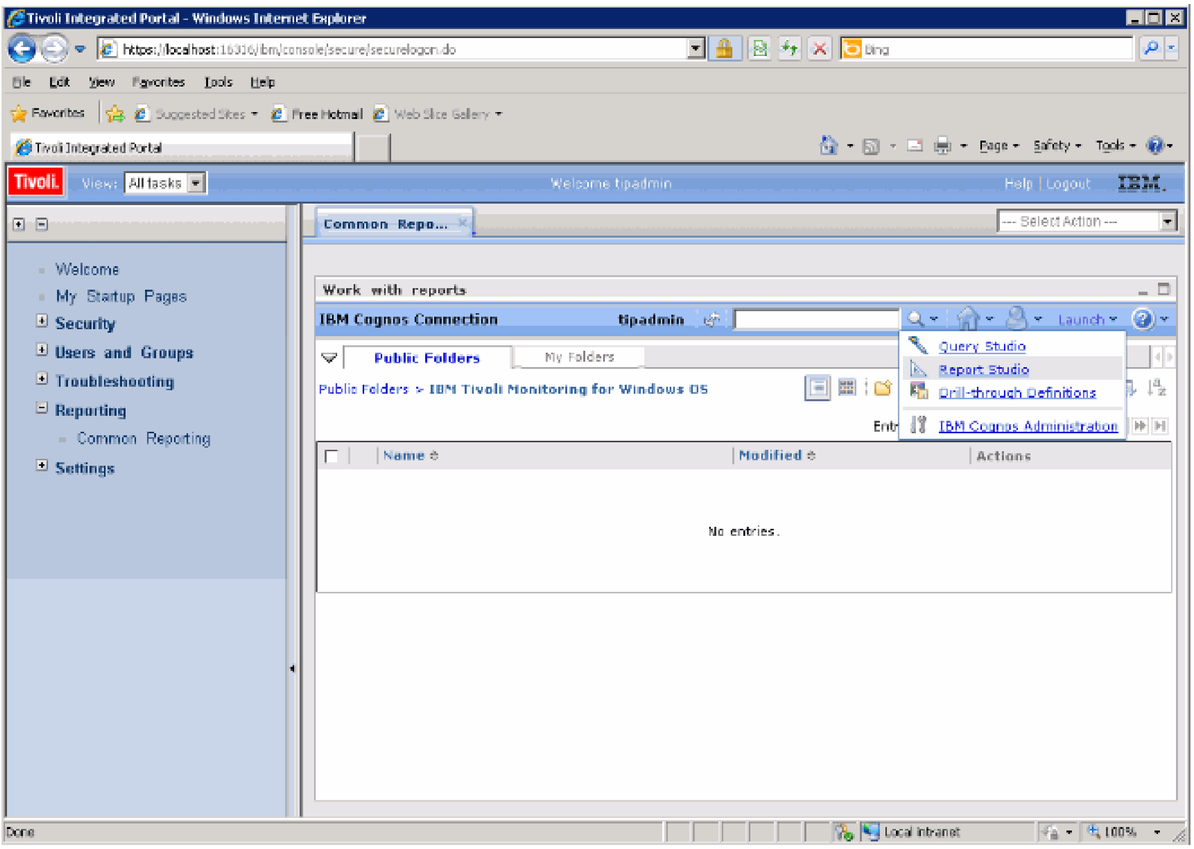
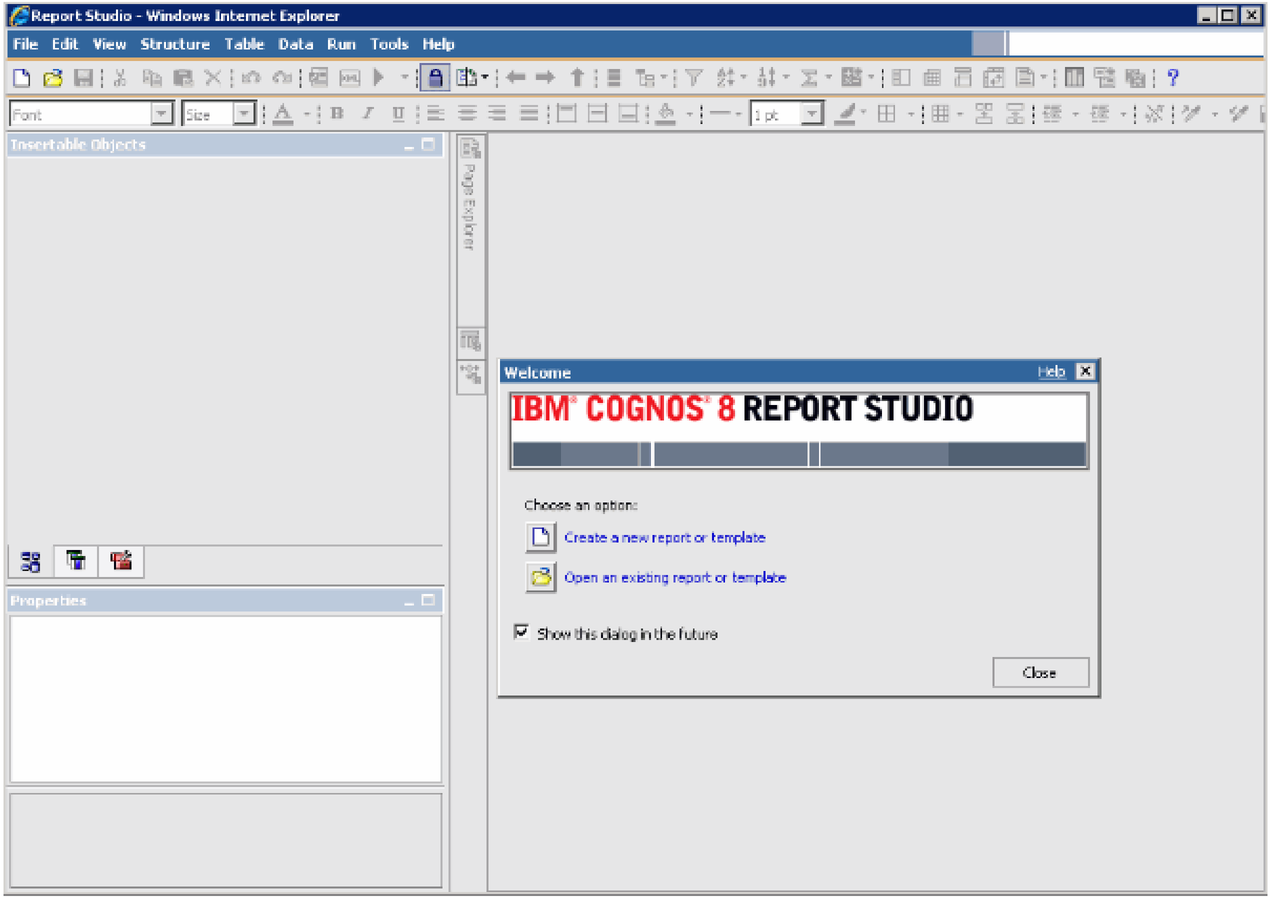


Figure 83. Selecting Report Studio

What to do next

You can use the Report Studio to create new reports or templates, or you can modify an existing report or template.



Report Studio - Windows Internet Explorer
File Edit View Structure Table Data Run Tools Help
Font
Size
1 pt
Insertable Objects
Page Explorer
Properties
Help

Welcome

IBM® COGNOS® 8 REPORT STUDIO

Choose an option:

Create a new report or template

Open an existing report or template

☑ Show this dialog in the future

Figure 84. Report Studio

For more information, see the Tivoli Common Reporting topic collection on IBM Knowledge Center.

Populating the ManagedSystem Table

The `ManagedSystem` table is populated by using the `kqz_populate_msn` stored procedure.

For more information, see "Running the DB2 stored procedure" on page 1451. This procedure must be run periodically so that the `ManagedSystem` table contains the current list of managed system names.

The stored procedure reads the following historical tables in the Tivoli Data Warehouse if they exist:

- The agent's Performance Object status table
- The agent's availability table. Agents that monitor processes or services have an availability table.
- The agent's discovery tables. Subnode agents create discovery tables.

Historical collection must be started on a particular set of attribute groups. A set of scripts is generated that creates and starts historical collection for these attribute groups. If you do not want to use the scripts, the list of attribute groups is listed in the comment header block of the script.

Sample scripts are created that show which tables must have historical collection enabled:

- `reports/configuretdw.sh`
- `reports/configuretdw.bat`

The following table describes the required arguments:

Note: You must specify either `-n` or `-m`, but not both.

Table 304. Required arguments

Argument	Description
-h *candle_home*	The Tivoli Monitoring installation path.
-u *teps_user*	The Tivoli Enterprise Portal Server user to log in as when you create the historical collections.
-n *tems_name*	The Tivoli Enterprise Monitoring Server where the collections must be started. More than one Tivoli Enterprise Monitoring Server can be specified by using a space separated list. If you specify more than one Tivoli Enterprise Monitoring Server, put the list in quotation marks. For example, `-n “tems1 tems2”`
-m *managed_system_group_or_managed_system*	The managed system group or managed system name against which the collection must be started. More than one managed system group or managed system can be specified by using a space separated list. If you specify more than one managed system group or managed system, put the list in quotation marks. For example, `-m “msg1 msg2”`

The following table describes the optional arguments:

Table 305. Optional arguments

Argument	Description
-s *teps_host*	The host name or IP address of the Tivoli Enterprise Portal Server. If not specified, the default is localhost.
-p *teps_password*	The password for the Tivoli Enterprise Portal Server user that is specified with the `-u` option. If not specified, the script prompts for the password
-c *historical_collection_interval*	The historical collection interval to use when you start the historical collections. If not specified, the default is `1h` (1 hour). The valid values are: `15m`, `30m`, `1h`, `12h`or `1d`, where `m` is minutes, h is hours and d is days.
-r *pruning_interval*	The pruning interval to use for the historical data. The historical data must be pruned so that the tables do not continue to grow in size. If not specified, the default is 2d(2 days). Use d for days, `m` for months, y for years.

After historical collection is started, the `kqz_populate_msn` stored procedure must be run periodically. The stored procedure is run periodically so that the `ManagedSystem` table contains the most current list of managed systems in the Tivoli Monitoring environment.

Running the DB2 stored procedure
Run a stored procedure on DB2.

About this task

Perform the following steps to run the stored procedure on DB2:

Procedure

1. Connect to the Tivoli Data Warehouse database as the warehouse user:

```
connect to <Tivoli Data Warehouse database alias> user
<Tivoli Data Warehouse user id> using <password>
```

2. Run the stored procedure:

```
db2 "call <Tivoli Data Warehouse schema>.kqz_populate_msn
('<three letter product code for the agent>')"
```

Running the Oracle stored procedure

Run a stored procedure on Oracle.

About this task

Perform the following steps to run the stored procedure on Oracle:

Procedure

1. Start `sqlplus`:

```
sqlplus <Tivoli Data Warehouse user id>/<password>@
<Oracle SID>
```

2. Run the stored procedure:

```
execute kqz_populate_msn('<three letter product code for the agent>');
```

Running the stored procedure on SQL Server 2005 and 2008

Run a stored procedure on SQL Server.

About this task

Perform the following steps to run the stored procedure on SQL Server 2005 and 2008:

Procedure

Run the stored procedure:

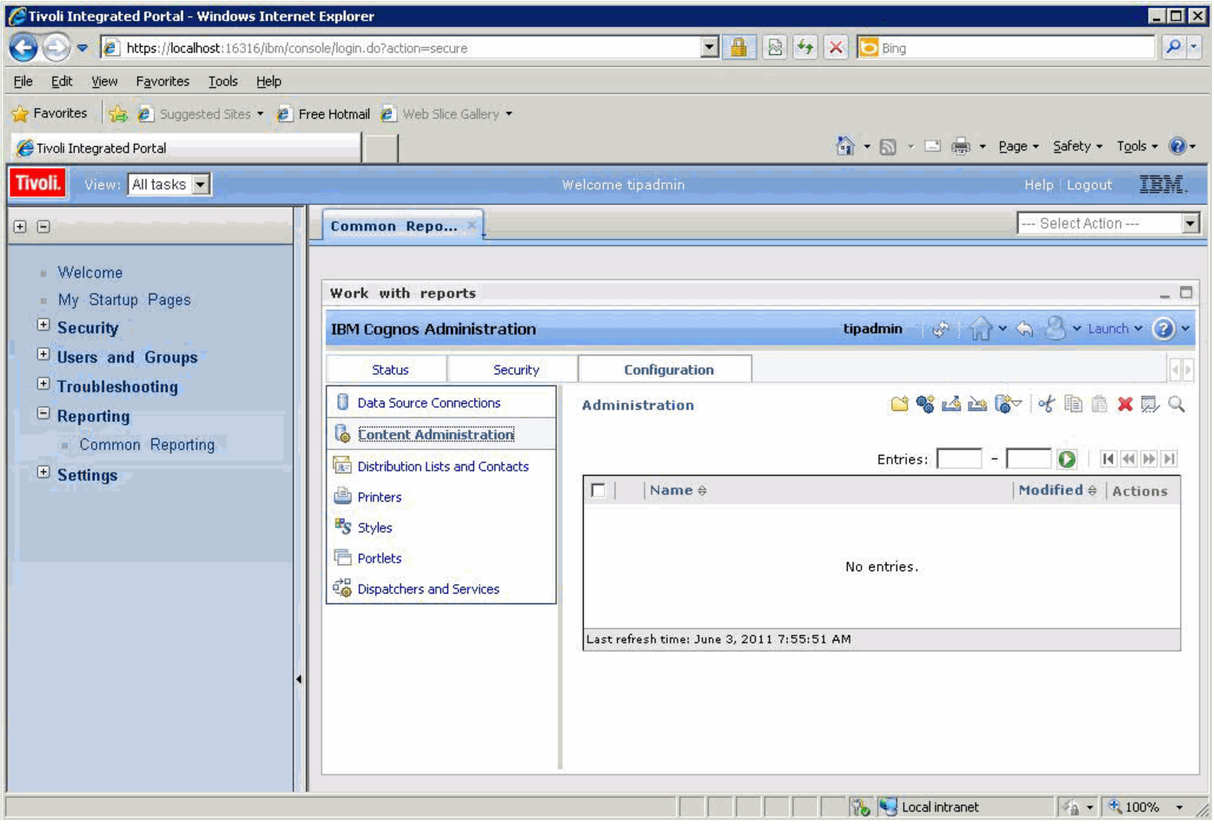
```
osql -S <server> -U <Tivoli Data Warehouse id> -P
<Tivoli Data Warehouse password> -d
<Tivoli Data Warehouse database name> -Q "EXEC
[<Tivoli Data Warehouse schema>].[kqz_populate_msn]
@pv_productcode = N'<three letter product code>'"
```

Exporting reports and data models from Tivoli Common Reporting

Export reports and data models from Tivoli Common Reporting.

Procedure

1. Log in to the Tivoli Common Reporting.
2. Go to Public Folders, and under **Reporting** in the navigation panel select **Common Reporting**.
3. In the Work with reports section, click the **Launch** menu and select **IBM Cognos Administration**.
4. Click the **Configuration** tab.
5. Click **Content Administration**.

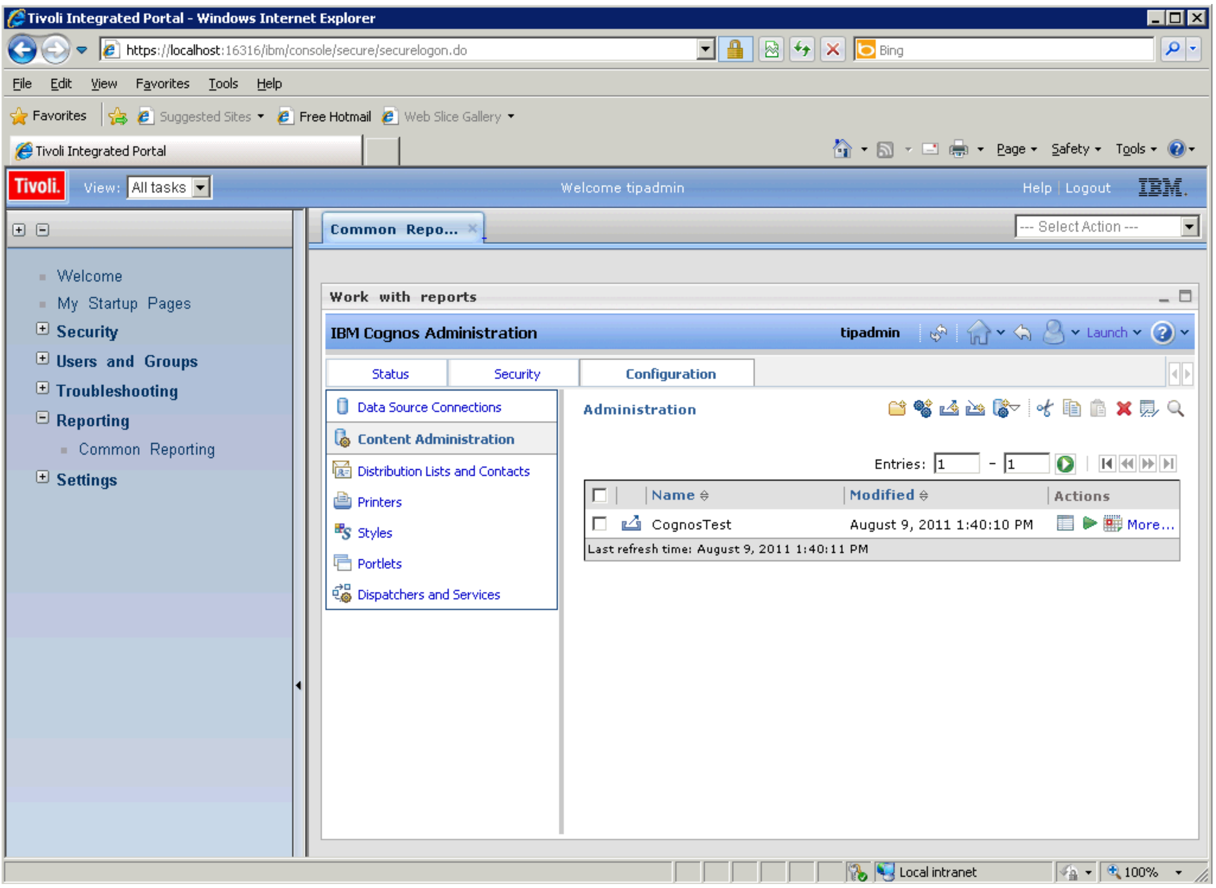


No entries.

Last refresh time: June 3, 2011 7:55:51 AM

Figure 85. The Content Administration tab

6. Click the **New Export** icon to export a new package.
7. Name the package. Optionally, you can add a screen tip and description.
8. Select **Select public folders and directory content** .
9. In the Public Folders dialog, click the **Add** link.
10. Move your agent package to **Selected entries**.
11. On the last page of the wizard, select **Save Only**. When the wizard completes, the report package is listed on the Content Administration tab.
12. On the Content Administration tab, click the green arrow (Run) to create the compressed `.zip` file.



Entries: 1 - 1

☐	Name	Modified	Actions
☐	CognosTest	August 9, 2011 1:40:10 PM	More...

Last refresh time: August 9, 2011 1:40:11 PM

Figure 86. The Content Administration tab with agent package listed

Results

The compressed `.zip` file that is created by the export process is placed in the deployment directory.

- The directory path for Tivoli Common Reporting version 1.3 is:

```
C:\IBM\tivoli\tip\products\tcr\Cognos\c8\deployment
```

- The directory path for Tivoli Common Reporting version 2.1 or later is:

```
C:\IBM\tivoli\tipv2Components\TCRComponent\cognos\deployment
```

What to do next

For more information about exporting reports, see Exporting Cognos report packages in the *Tivoli Common Reporting User's Guide*.

Importing reports into Agent Builder

When the report package is exported from Tivoli Common Reporting, it can be imported into the Agent Builder project. The report package can then be included in the agent installation image.

Procedure

1. Right-click on the agent project in the Agent Builder.
2. Select **IBM** > **Import Report Package**.
3. In the **Import Report Package** window, select the **Database Type** on which the report package was created.
4. Enter the fully qualified path to the report package, or click **Browse** to select it.
5. Click **OK**.

6. The report package is now shown in the agent project under the `reports/dbtype` directory.

   Note: If you create report packages that are database-specific you must import each package into the Agent Builder.

Installing reports from an agent package into Tivoli Common Reporting

Import a report package from your agent to Tivoli Common Reporting

Procedure

1. Follow the steps in the wizard to import a new package from your agent image.

   In the agent image, the reports are found in: `reports/dbType/Kxx/reports/cognos_reports/itmkxx/packages`
2. Copy the reports compressed `zip` file into the Tivoli Common Reporting deployment directory.
   - The directory path for Tivoli Common Reporting version 1.3 is: `C:\IBM\tivoli\tip\products\tcr\Cognos\c8\deployment`
   - The directory path for Tivoli Common Reporting version 2.1 or later is: `C:\IBM\tivoli\tipv2Components\TCRComponent\cognos\deployment`
3. Log in to the Tivoli Common Reporting.
4. Go to Public Folders, and under **Reporting** in the navigation panel select **Common Reporting**.
5. In the Work with reports section, click the **Launch** menu and select **IBM Cognos Administration**.
6. Go to the **Configuration** tab, and open the **Content Administration** section.
7. Click **New Import** to create a package import.
8. Select the agent's reports package.
9. Select the public folders that you want to import.
10. Select save.
11. Click the green (run) arrow to import.

Results

For more information, see Logging in to the reporting interface in the *Tivoli Common Reporting User's Guide*.

ICU regular expressions

A description of the specifics of the ICU regular expression implementation.

This reference content is extracted from the *ICU User Guide*. The content describes the specifics of the ICU regular expression implementation. This information is essential if you are using the Agent Builder regular expression feature because different programming languages implement regular expressions in slightly different ways.

Table 306. Regular expression metacharacters

Character	Description
\a	Match a BELL, \u0007
\A	Match at the beginning of the input. Differs from ^ in that \A does not match after a new line within the input.
\b, outside of a [Set]	Match if the current position is a word boundary. Boundaries occur at the transitions between word (`\w`) and non-word (`\W`) characters, with combining marks ignored. For more information about word boundaries, see ICU Boundary Analysis.

Table 306. Regular expression metacharacters (continued)

Character	**Description**
\b, within a [Set]	Match a BACKSPACE, `\u0008`.
`\B`	Match if the current position is not a word boundary.
`\cX`	Match a `Ctrl-X` character.
`\d`	Match any character with the Unicode General Category of Nd (Number, Decimal Digit.)
`\D`	Match any character that is not a decimal digit.
`\e`	Match an ESCAPE, `\u001B`.
`\E`	Terminates a `\Q ... \E` quoted sequence.
`\f`	Match a FORM FEED, `\u000C`.
`\G`	Match if the current position is at the end of the previous match.
`\n`	Match a LINE FEED, `\u000A`.
`\N{UNICODE CHARACTER NAME}`	Match the named character.
`\p{UNICODE PROPERTY NAME}`	Match any character with the specified Unicode Property.
`\P{UNICODE PROPERTY NAME}`	Match any character not having the specified Unicode Property.
`\Q`	Place quotation marks around all following characters until `\E`.
`\r`	Match a CARRIAGE RETURN, `\u000D`.
`\s`	Match a white space character. White space is defined as `[\t\n\f\r\p{Z}]`.
`\S`	Match a non-white space character.
`\t`	Match a HORIZONTAL TABULATION, `\u0009`.
`\uhhhh`	Match the character with the hex value hhhh.
`\Uhhhhhhhh`	Match the character with the hex value hhhhhhhh. Exactly eight hex digits must be provided, even though the largest Unicode code point is `\U0010ffff`.
`\w`	Match a word character. Word characters are `[\p{Ll}\p{Lu}\p{Lt}\p{Lo}\p{Nd}]`.
`\W`	Match a non-word character.
`\x{hhhh}`	Match the character with hex value hhhh. From one to 6 hex digits can be supplied.
`\xhh`	Match the character with 2 digit hex value hh.
`\X`	Match a Grapheme Cluster.
`\Z\`	Match if the current position is at the end of input, but before the final line terminator, if one exists.

Table 306. Regular expression metacharacters (continued)

Character	Description
`\z`	Match if the current position is at the end of input.
`\n`	Back Reference. Match whatever the `nth` capturing group matched. n must be a number `>` 1 and < total number of capture groups in the pattern. **Note:** Octal escapes, such as `\012`, are not supported in ICU regular expressions.
`[pattern]`	Match any 1 character from the set. See `UnicodeSet` for a full description of what can appear in the pattern
`.`	Match any character.
`^`	Match at the beginning of a line.
`$`	Match at the end of a line.
`\`	Place quotation marks around the following character. Characters that must have surrounding quotation marks to be treated as literals are `* ? + [ ( ) { } ^ $ | \ . /`

Table 307. Regular expression operators

Operator	Description
`|`	Alternation. A|B matches either A or B.
`*`	Match 0 or more times. Match as many times as possible.
`+`	Match 1 or more times. Match as many times as possible.
`?`	Match zero or 1 time. Prefer one.
`{n}`	Match exactly n times
`{n,}`	Match at least n times. Match as many times as possible.
`{n,m}`	Match between n and m times. Match as many times as possible, but not more than m.
`*?`	Match 0 or more times. Match as few times as possible.
`+?`	Match 1 or more times. Match as few times as possible.
`??`	Match zero or 1 time. Prefer zero.
`{n}?`	Match exactly n times
`{n,}?`	Match at least n times, but no more than required for an overall pattern match
`{n,m}?`	Match between n and m times. Match as few times as possible, but not less than n.

Table 307. Regular expression operators (continued)	
Operator	**Description**
`*+`	Match 0 or more times. Match as many times as possible when first encountered, do not retry with fewer even if overall match fails (Possessive Match)
`++`	Match 1 or more times. Possessive match.
`?+`	Match zero or 1 time. Possessive match.
`{n}+`	Match exactly n times
`{n,}+`	Match at least n times. Possessive Match.
`{n,m}+`	Match between n and m times. Possessive Match.
`( ... )`	Capturing parentheses. Range of input that matched the parenthesized subexpression is available after the match.
`(?: ... )`	Non-capturing parentheses. Groups the included pattern, but does not provide capturing of matching text. More efficient than capturing parentheses.
`(?> ... )`	Atomic-match parentheses. First match of the parenthesized subexpression is the only one tried. If it does not lead to an overall pattern match, back up the search for a match to a position before the `"(?>"`.
`(?# ... )`	Free-format comment `(?# comment )`.
`(?= ... )`	Look-ahead assertion. True if the parenthesized pattern matches at the current input position, but does not advance the input position.
`(?! ... )`	Negative look-ahead assertion. True if the parenthesized pattern does not match at the current input position. Does not advance the input position.
`(?<= ... )`	Look-behind assertion. True if the parenthesized pattern matches text that precedes the current input position. The last character of the match is the input character just before the current position. Does not alter the input position. The length of possible strings that is matched by the look-behind pattern must not be unbounded (no * or + operators.)
`(?<!...)`	Negative Look-behind assertion. True if the parenthesized pattern does not match text that precedes preceding the current input position. The last character of the match is the input character just before the current position. Does not alter the input position. The length of possible strings that is matched by the look-behind pattern must not be unbounded (no * or + operators.)

Table 307. Regular expression operators (continued)

Operator	Description
`(?ismx-ismx: ... )`	Flag settings. Evaluate the parenthesized expression with the specified flags enabled or disabled.
`(?ismx-ismx)`	Flag settings. Change the flag settings. Changes apply to the portion of the pattern that follows the setting. For example, `(?i)` changes to a not case-sensitive match.

Replacement text

The replacement text for find-and-replace operations can contain references to capture-group text from the find. References are of the form $n, where n is the number of the capture group.

Table 308. Replacement text characters

Character	Description
$n	The text of the positional capture group n is substituted for `$n.` n must be `>= 0`, and not greater than the number of capture groups. A $ not followed by a digit has no special meaning, and is displayed in the substitution text as itself, a $.
\	Treat this character as a literal, suppressing any special meaning. Backslash escaping in substitution text is required only for `'$'` and `'\'`, but can be used on any other character without adverse effects.
$@n	The text of capture group n is substituted for the regular expression that matched capture group n. n must be `>= 0`, and not greater than the number of capture groups. A $@ not followed by a digit has no special meaning, and is displayed in the substitution text as itself, a $@.
$#n	The text of the matched capture group n is substituted for $#n. n must be `>= 0`, and not greater than the number of matched capture groups. A $# not followed by a digit has no special meaning, and is displayed in the substitution text as itself, a $#.

Flag options

The following flags control various aspects of regular expression matching. The flag values can be specified at the time that an expression is compiled into a RegexPattern object. Or, they can be specified within the pattern itself using the `(?ismx-ismx)` pattern options.

Table 309. Flag options

Flag (pattern)	Flag (API constant)	Description
`i`	`UREGEX_CASE_INSENSITIVE`	If set, matching take place in a case-insensitive manner.

Table 309. Flag options (continued)

Flag (pattern)	Flag (API constant)	Description
x	UREGEX_COMMENTS	If set, white space and #comments can be used within patterns.
s	UREGEX_DOTALL	If set, a "." in a pattern matches a line terminator in the input text. By default, it does not. A carriage-return / line-feed pair in text behaves as a single-line terminator, and matches a single "." in a RE pattern
m	UREGEX_MULTILINE	Control the behavior of "^" and "$" in a pattern. By default these patterns match only at the start and end, respectively, of the input text. If this flag is set, "^" and "$" also match at the start and end of each line within the input text.

Creating Non-agent file bundles

You can create file bundles that can be placed in the Tivoli Monitoring depot. These file bundles can then be deployed to target systems in your environment.

About this task

With this function, you can remotely configure products for which there is no remote configuration option. To use this function, you place pre-populated configuration files into the depot and send them out to the wanted systems.

Procedure

1. From the Agent Builder, select **File** > **New** > **Other**.
2. Under **Agent Builder**, select **Non-Agent Remote Deploy Bundle**.
3. Click **Next**.
4. In the **Project name** field, enter a name for your project.
5. Click **Next**.
6. Complete the information in the **Remote Deploy Bundle Information** window:
   a) In the **Bundle identifier** field, type an identifier that is a unique alphanumeric string of 3 - 31 characters. This string can contain a hyphen. The string must start with a letter, but it cannot start with a K or a hyphen.
   b) In the **Bundle description** field, type a description of the bundle.
   c) In the **Version** field, type a version for the bundle in the VVRRMMFFF format. Where vv= version number; rr= release number; mm= modification number (fix pack number); and fff = interim fix number.
7. In the **Operating Systems** area, select the operating systems to which the bundle can be deployed.
8. Click **Finish** to create a project in the workspace and open the **Remote Deploy Bundle Editor**.

Remote Deploy Bundle Editor

The Remote Deploy Bundle Editor is used to generate commands to help deploy your file bundle.

The Remote Deploy Bundle Editor provides information about the bundle for a project.

The **Bundle Identification Information** section contains the following information:

Bundle identifier
Unique ID for the bundle

Bundle description
Description for the bundle

Bundle version
Version of the bundle

Build
Build identifier for the bundle. Enter a build number here. If no build number is specified, a number is generated from the date and time when the bundle is generated.

Create copy commands for the files in the bundle check box
Click the check box to generate a set of default copy commands that run when the bundle is deployed. The files are copied to the location specified in the **Copy location** text box. The default location is *`INSTALLDIR`*. Specify this remote deployment variable from the command-line deployment by setting `KDY.`*`INSTALLDIR`*`=...`

The **Operating Systems** section shows the operating systems to which the bundle can be deployed.

The **Commands** section shows the commands to run when the bundle is deployed.

Prerequisite Bundles section shows the bundles that must be present for this bundle to work.

Use the Remote Deploy Bundle Editor to opt for a set of default copy commands that copy the files in your bundle to a set location. If this option is selected, then a copy command is generated for each file in your bundle project. The default copy location is *`INSTALLDIR`*. A special remote deployment variable that, if not set on the deployment command line, defaults to *`CANDLEHOME`*. To change the location that is specified by *`INSTALLDIR`*, specify the **`KDY.INSTALLDIR`** property when you run the **`addSystem`** command.

The same directory structure that is specified in your bundle project is replicated in *`INSTALLDIR`*. For example, if there is a folder named `config` in your bundle project with a file named `myprod.config`, then the generated copy command copies the file to *`INSTALLDIR`*`/config/myprod.config`when the bundle is deployed.

Adding commands to the bundle

You can specify more commands to run during the deployment.

About this task

You can specify more commands to run during the deployment by using the **Remote Deploy Bundle Editor**.

Procedure

1. To specify more commands to run during the deployment, click **Add** in the **Commands** section of the **Remote Deploy Bundle Editor**.
2. In the **Command** window, select the type of command **Preinstall**, **Install**, **Post-Install**, or **Uninstall**and then specify the command to run.

   You must specify the fully qualified path to the command you want to run. For convenience, remote deployment provides a set of predefined variables. To reference the variable for a command, surround the variable with vertical bars, for example, `|DEPLOYDIR|`. For more information about predefined variables for commands, see (Table 310 on page 1462).

Table 310. Predefined Variables for Commands

Variable	Description
DEPLOYDIR	The temporary directory on the endpoint where the bundle is stored during the deployment. For instance, if you want to run `myscript.sh`, a script that is included in your bundle, you specify the following command: `|DEPLOYDIR|/ myscript.sh`
INSTALLDIR	Either *CANDLEHOME* or the value of *KDY.INSTALLDIR* if specified on the **addSystem** command.
CANDLEHOME	The Tivoli Monitoring installation directory.

3. Finally, select the **Operating Systems** on which the command is to run.

Adding prerequisites to the bundle

Use the **Remote Deploy Bundle Editor** to specify prerequisites for the bundle.

Procedure

1. To add a prerequisite, click **Add** in the **Prerequisite Bundles** section of the **Remote Deploy Bundle Editor**, **Bundle Information** page.
2. In the **New Prerequisite** window, enter the bundle identifier on which this bundle depends and the minimum version required.
3. Select the operating systems for which this prerequisite is required.
4. Click **OK** to complete and exit.

Adding files to the bundle

Add files to a file bundle by using the **Remote Deploy Bundle Editor**.

Procedure

1. To add files to the remote deployment bundle, do one of the following procedures:
   - In the Bundle Editor, click **Add files to the bundle**.
   - Right-click the project in the Navigator tree, then click **IBM Tivoli Monitoring Remote Deploy** > **Add Files to Bundle**

   Both of these actions display the **Import Bundle Files** window:
2. Specify individual files or directories that contain files in **File Information** area.
3. Click **Finish**.

   The files or directories that are specified are copied into the project directory. The directory structure in the project is maintained when you build the remote deployment bundle. If you want Agent Builder to generate default copy commands, ensure that the files are in the correct directory structure for deployment.

Generating the bundle

Use Agent Builder to generate a bundle for remote deployment of an agent.

Procedure

1. To generate the remote deployment bundle, use one of the following procedures to display the **Generate Final Remote Deploy Bundle** window

    - In the **Remote Deploy Bundle Editor**, click **generate the final Remote deploy bundle**.
    - Right-click the project in the Navigator tree, then click **IBM Tivoli Monitoring Remote Deploy** > **Generate Remote Deploy Bundle**

2. You can now generate the bundle in two ways:
    - If there is a Tivoli Enterprise Monitoring Server on the system where you are running the Agent Builder, click **Install the Remote Deploy bundle into a local TEMS depot**.

      The Agent Builder attempts to determine the Tivoli Monitoring installation location and enter it into the **Directory** field. If *`CANDLE_HOME`* is not set, the default location of `C:\IBM\ITM` or `/opt/IBM/ITM` is used. Ensure that the installation location is correct before you continue.

      You must provide Tivoli Enterprise Monitoring Server login information to install the bundle.
    - To generate the bundle to a directory on your system, click **Generate the Remote Deploy bundle in a local directory**

      After the process is complete, you must transfer this directory to a Tivoli Enterprise Monitoring Server system and use the `tacmd addbundles` command to add the bundle to the depot.

What to do next

When you deploy the bundle, you must use the `tacmd addSystem` command. For example:

```
tacmd addsystem -t MONITORINGCOLLECTION -n Primary:ITMAGT:NT
```

Where `-t` (type) is the Product Code as returned by the `tacmd viewDepot` command:

```
>tacmd viewDepot
Product Code : MONITORINGCOLLECTION
Version : 010000003
Description : MonitoringCollectionScripts
Host Type : WINNT
Host Version : WINNT
Prerequisites:
```

Note: You cannot deploy remotely from the Tivoli Enterprise Portal Desktop or Browser. Deploy remotely from the Tivoli Enterprise Portal Desktop or Browser results in the KFWITM219E message.

See the Tivoli Monitoring documentation for more details.

Creating deployable bundles for Tivoli Netcool/OMNIbus probes

You can use the Agent Builder to create package and configuration bundles that can be used to deploy Tivoli Netcool/OMNIbus probes to remote computers.

About this task

To support the remote deployment of probes, you can also create Tivoli Netcool/OMNIbus bundles that can be deployed to the remote computers before you deploy the probes.

Procedure

1. From the Agent Builder, select **File** > **New** > **Other**.
2. Under **IBM Tivoli OMNIbus Wizards**, select **Package Bundle**.
3. Click **Next**.

What to do next

Next, use the **OMNIbus Install Bundle** wizard to create the bundles. For information about using this wizard, see the Tivoli Netcool/OMNIbus documentation.

Dynamic file name support

Use dynamic file name support to specify a file name pattern instead of an actual file name.

Some application programs create an output file name that is subject to change. The name changes based on specific criteria such as the current day, month, year, or a file name that includes an incrementing sequence number. In these cases, you can specify the file name pattern instead of the actual file name. There are two pattern formats that are recognized when you specify the file name pattern:

- Regular Expressions (preferred).
- IBM Tivoli Universal Agent dynamic file name syntax (deprecated).

Regular expression file name patterns

To specify file name patterns, you can use regular expressions according to the International Components for Unicode (ICU) syntax that is documented in ("ICU regular expressions" on page 1455). To use this capability, you must select the **File names match regular expression** check box on the **Advanced Log File Attribute Group Information** page. When you specify regular expression patterns, you must also select an option from the **When Multiple Files Match** list on the **Advanced Log File Attribute Group Information**page to specify the guidelines for selecting the most current matching file.

Note: Regular expressions is the preferred method to specify file name patterns.

For more information about how to configure advanced log file attribute group properties, see ("Monitoring a log file" on page 1235), Step ("6" on page 1236). For example, if you specified a file name pattern:

```
d:\program files\logs\tivoli.*
```

This pattern searches for file names that start with `tivoli` in the `d:\program files\logs` directory. Regular expressions can be specified only for the file name portion, and not the path name.

Dynamic file name syntax

With the dynamic file name syntax, only one file at a time can be monitored. The File Data Provider inspects all files in the designated path location, seeking files that match the defined pattern. The File Data Provider always monitors the most current matching file that is based on whichever matching file name has the highest number or date-time value. The appropriate file to monitor is determined by file name, instead of by file creation or other criteria.

Patterns can be specified for file names with any number of parts. For example, `Log{###}` matches on one-part file names such as `Log010` or `Log456`. In multi-part file names, pattern characters can be specified in any part of the file name or in multiple parts. For example, `aaa.bbb{???}.ccc` is a valid pattern, and `aaa.bbb{???}.ccc{###}` is also valid.

Note: Regular expressions rather than dynamic file name syntax is the preferred method to specify file name patterns, for more about regular expressions, see "Regular expression file name patterns" on page 1464

The following examples illustrate file name pattern specification:

`{########}.abc`
Matches numeric file names of length 8 and the file extension *.abc*, such as `10252006.abc` or `10262006.abc`. File `10262006.abc` is monitored because `10262006` is greater than `10252006`.

`{########}.*`
Matches numeric file names of length 8 and ignores the file extension. Examples include `20061025.log`, `20061101.log`, and `10252006.abc`. File `20061101.log` is monitored because `20061101` is the largest number.

`{######??}.abc`
: Matches numeric file names of length 8 and file extension `.abc`, and ignores the last two positions in the name portion. Examples include `02110199.abc`, `02110200.abc`, and `021101AZ.abc`. File `02110200.abc` is monitored because `021102` is the largest number.

`Console.{######}`
: Matches file names that contain *`Console`* in the name portion and a six-digit number in the extension portion. Examples include `Console.000133`, `Console.000201`, and `Console.000134`. File `Console.000201`is monitored.

`IN{######}.log`
: Matches file names that start with `IN` followed by six numerals and the file extension `.log`. Examples include `IN021001.log`, `IN021002.log`, and `IN021004.log`. File `IN021004.log` is monitored.

`PS{###}FTP.txt`
: Matches file names that start with PS followed by three numerals, followed by FTP, and the extension `.txt`. Examples include `PS001FTP.txt`, `PS005FTP.txt`, and `PS010FTP.txt`. File `PS010FTP.txt`is monitored.

Follow these guidelines to establish file name patterns:

- Use braces {} to enclose pattern characters in a file name. The presence of pattern characters inside braces indicates that a file name pattern is being used.
- Use an asterisk (*) as a wildcard to ignore file extensions or any trailing characters in the file name. For example, `Myapp{###}.log*` specifies that any file name that starts with `Myapp`, followed by three digits, and followed by `.log`, is a match, regardless of what comes after.

  The asterisk must be specified after the curly braces `({ })` and cannot be used at the beginning of a file name. When you use the asterisk in a file name extension, the asterisk must be used by itself.

  Examples of correct wildcard `(*)`usage:

  `err{??}.*`

  `error{$}.*`

  Examples of incorrect wildcard `(*)` usage:

  `error.20*`
  : No curly braces precede the asterisk `(*)`.

  `error*.{###}`
  : The asterisk is not used at the end of the file name.

  `error.*`
  : No curly braces precede the asterisk `(*)`.
- If a specific file extension is defined, then only files with the same extension are considered.
- Use a number sign to indicate each numeric element of a file name.
- Use a question mark to exclude each element of the naming convention that does not serve as search criteria in determining the appropriate file name.
- Use a dollar sign `($)` to represent either any character or no character. For example, if you want to match on two files named `Log` and `LogA`, specify `Log{$}`. The dollar sign has several usage restrictions. When you use one or more dollar signs to prefix a file name as in `{$$$$$$}_abc.log`, the number of dollar signs must exactly match the number of characters in that position in the file name. Also, you cannot specify dollar signs in multiple locations in a file name pattern, for example, `{$$$}b{$$$}.log` does not match `abc.log`. Given these dollar sign restrictions, use regular expression file name patterns if there are an indeterminate number of characters in the file names.
- The total number of number signs and question marks that are enclosed in braces is significant. It must match the portion of file name exactly. For example, the pattern `AA{####}` instructs the File Data Provider to look for files such as `AA0001`. File names, such as `AA001` or `AA00001`, are not considered.
- The exact file name pattern, the constant, and the numeric parts, must match the file name exactly. For example, the pattern `AA{###}` instructs the File Data Provider to check file `AA101`. File names, such as `XAA101`, `AA222X` and `AA55555`, are not considered.

- Use the reserved pattern string {TIVOLILOGTIME}to substitute for the hex timestamp and file sequence number in a Tivoli Monitoring agent or server log file. This pattern string is useful when you do self-monitoring of Tivoli Monitoring components. For example, if you want to monitor the latest monitoring server log in the /opt/IBM/ITM/logsdirectory, can specify a file name pattern:

  ```
  /opt/IBM/ITM/logs/Host1_ms_{TIVOLILOGTIME}.log
  ```

  If Host1_ms_452053c0-01.log, Host1_ms_451f11f4-01.log, Host1_ms_45205946-01.log, and Host1_ms_451f11f4-02.log are present in the /logsdirectory, the Host1_ms_45205946-01.log file is selected for monitoring.

To precisely specify a file name that consists of date components (year, month, and day), use the capital letters Y, M, and D. These letters must be specified within braces; otherwise they are treated as literal characters in the file name.

See the following examples:

{YYYYMMDD}.log
Specifies file names such as 20060930.log or 20061015.log.

{MMDDYY}.log
Specifies file names such as 101106.log or 110106.log.

{DDMMYYYY}.log
Specifies file names such as 01092006.log or 15082006.log.

{DDMMMYY}.log
Specifies file names such as 24Jan07 or 13Sep06.

{MM-DD-YY}.log
Specifies file names such as 11-02-06 or 04-29-07. The (-)separator character is ignored in the date field and does not require a question mark pattern character to skip over it.

MY{YYDDD}.log
Specifies file names such as MY06202.log, MY06010.log, or MY04350.log.

Complex cases exist, where a date field is embedded within a longer file name, and the date patterns in the previous examples are not sufficient. For complex cases, create patterns that mix number signs and question marks and still perform numeric comparisons that select the most current file for monitoring. For example, the pattern ABC{?####?##?##?##?##?##?}XYZ.TXT can be used for file names such as ABC 2006-04-20 11_22_33 XYZ.TXT. In this example, you are interested in only the #- marked digits and question marks serve as placeholders that ignore other characters in the file name.

The File Data Provider periodically checks for new files that match the defined file pattern in the target path location. When a newer file that matches the pattern is detected, the File Data Provider automatically switches application monitoring to the new file. The File Data Provider searches for the best matching file when:

- The File Data Provider first starts.
- The currently monitored file no longer exists because of possible renaming or deleting.
- The existing file contents, changed because of possible rewriting.
- The check interval expired. The default interval is 10 minutes. You can change the interval to a longer or shorter interval value by specifying the environment variable

  ```
  KUMP_DP_FILE_SWITCH_CHECK_INTERVAL=number-of-seconds
  ```

SNMP trap configuration

Description of the configuration file that is used by the SNMP Data Provider to render trap information in a more easily readable form. The file is also used to assign categories, severities, status, and source IDs to traps.

Also contains instructions for modifying the default file or substituting your own configuration file.

SNMP trap configuration file, `trapcnfg`

At startup, the SNMP Data Provider reads a configuration file named `trapcnfg`. One purpose of this file is to translate SNMP trap information into a more readable form. Another is to assign categories, severities, status, and source IDs to specific traps, since these categories are not defined by SNMP.

You can modify the `trapcnfg` file to suit your site-specific needs by adding new trap or enterprise definitions or changing the existing ones. You can also use your own configuration file.

Use the HP OpenView `trapd.conf` file

The `trapcnfg` file is similar in format, but not identical, to the HP OpenView Network Node Manager trap configuration file `trapd.conf`. You can copy the OpenView file and reuse many of the definition statements if necessary.

Types of records

`trapcnfg` contains three types of records or record blocks:

comments
: Comment records begin with a number sign (#).

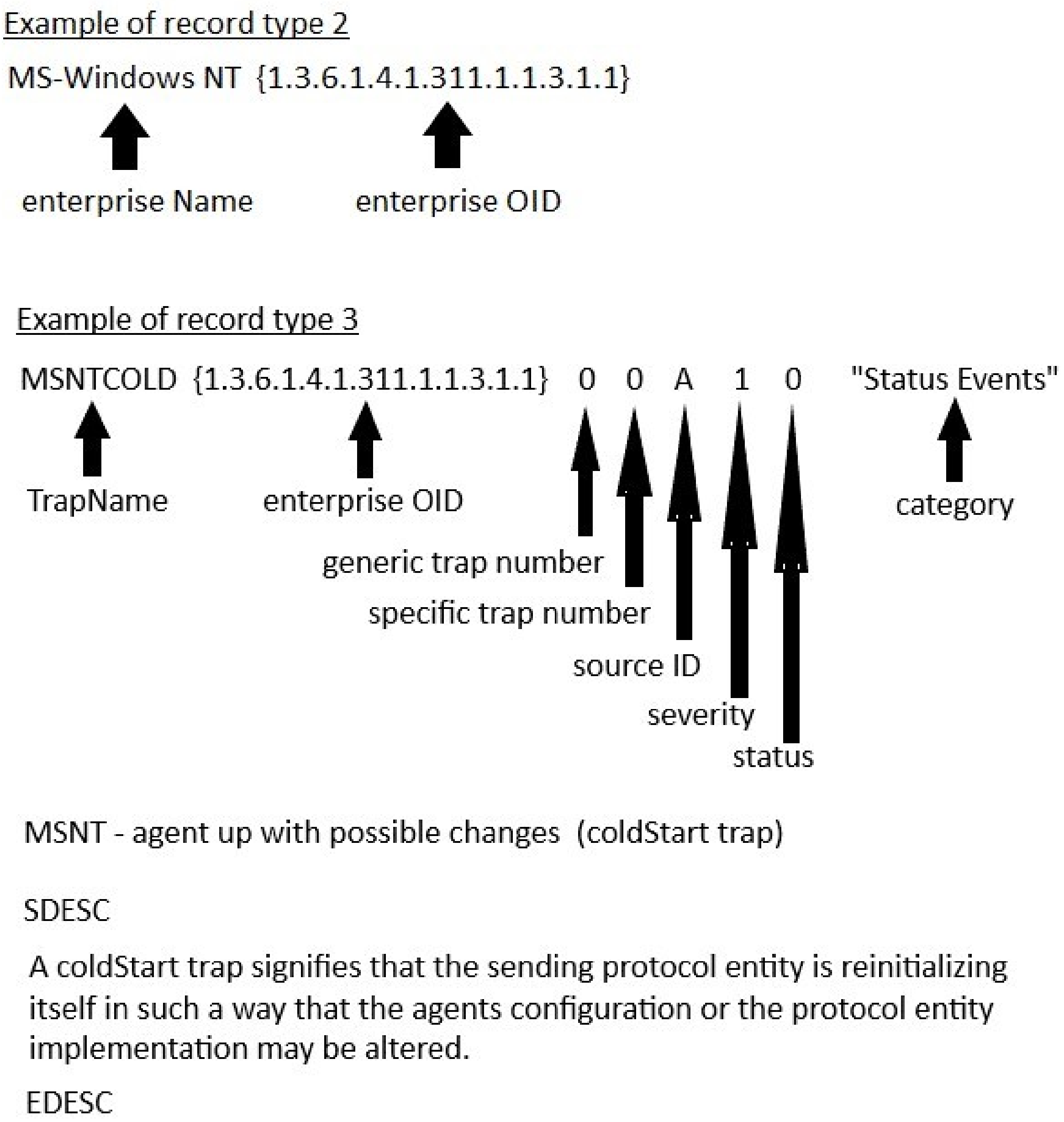
enterprise definitions
: Enterprise definitions consist of two blank-delimited tokens, where the first token is a name and the second is an object identifier (OID) surrounded by curly brackets ({ }).

trap definitions
: Trap definitions consist of eight blank-delimited tokens. Trap definitions are block records because each definition might consist of multiple records.

The first type is self-explanatory. (Figure 87 on page 1468) shows examples of the second and third types.

The first example in Figure 87 on page 1468 shows an enterprise definition record which defines enterprise OID 1.3.6.1.4.1.311.1.1.3.1.1 as being Microsoft Windows NT.

The second example shows a trap definition record that defines `trapName MSNTCOLD` as being associated with enterprise OID `1.3.6.1.4.1.311.1.1.3.1.1`, generic trap number 0, and specific trap number 0. Notice that the severity is in decimal form whereas the category is in textual form. Severities are translated into their textual form before they are displayed. The next record in the type 3 record block is the short description, which the Agent Builder does not use. The Agent Builder uses the long description that is enclosed within the delimiters SDESC and EDESC.



Example of record type 2

```
MS-Windows NT  {1.3.6.1.4.1.311.1.1.3.1.1}
```

Example of record type 3

```
MSNTCOLD  {1.3.6.1.4.1.311.1.1.3.1.1}  0  0  A  1  0   "Status Events"
```

```
MSNT - agent up with possible changes  (coldStart trap)

SDESC

A coldStart trap signifies that the sending protocol entity is reinitializing
itself in such a way that the agents configuration or the protocol entity
implementation may be altered.

EDESC
```

Figure 87. Examples of configuration record types 2 and 3

Defaults for the `trapcnfg` file

Tables that list the defaults that are supported by the SNMP Data Provider.

Supported categories

(Table 311 on page 1468) shows the categories that are supported by the Agent Builder.

Table 311. Categories supported by the SNMP Data Provider

Category	Textual representation
0	`Threshold Events`
1	`Network Topology Events`

Table 311. Categories supported by the SNMP Data Provider (continued)

Category	Textual representation
2	`Error Events`
3	`Status Events`
4	`Node Configuration Events`
5	`Application Alert Events`
6	`All Category Events`
7	`Log Only Events`
8	`Map Events`
9	`Ignore Events`

(Table 312 on page 1469) lists the severities that are supported by the Agent Builder.

Table 312. Severities supported by the SNMP Data Provider

Severity	Textual representation
0	`Clear`
1	`Indeterminate`
2	`Warning`
3	`Minor Error`
4	`Critical`
5	`Major Error`

Supported statuses

(Table 313 on page 1469) shows the statuses that are defined in the Agent Builder configuration file.

Table 313. Statuses supported by the SNMP Data Provider

Status	Textual representation
0	`Unchanged`
1	`Unknown`
2	`Up`
3	`Marginal`
4	`Down`
5	`Unmanaged`
6	`Acknowledge`
7	`User1`
8	`User2`

Supported source IDs

(Table 314 on page 1470) lists the source IDs supported by `trapcnfg`.

Table 314. Source IDs supported by the SNMP Data Provider

Source ID	Description
a	Application
A	Agent
C	Xnmcollect
d	Demo
D	Data Collector
E	Nvevents
I	Ipmap
L	LoadMIB
m	Shpmon
M	IP topology
n	netmon related
N	netmon-generated traps
O	OSI SA
P	Non-IP traps
r	Tralertd
s	Spappld
S	Security Agent
t	Xnmtrap
T	Trapd
V	Vendor related
?	Unknown

Take Action commands reference

An overview of Take Action commands, references about Take Action commands, and descriptions of special Take Action commands.

About Take Action commands

Take Action commands can be included in an Agent Builder monitoring agent. Take Action commands can be run from the portal client or included in a situation or a policy. When included in a situation, the command runs when the situation becomes true. A Take Action command in a situation is also known as reflex automation. When you enable a Take Action command in a situation, you automate a response to system conditions. For example, you can use a Take Action command to send a command to restart a process on the managed system. You can also use a Take Action command to send a text message to a cell phone.

Advanced automation uses policies to run actions, schedule work, and automate manual tasks. A policy comprises a series of automated steps that are called activities that are connected to create a workflow. After an activity is completed, the Tivoli Enterprise Portal receives return code feedback, and advanced automation logic responds with subsequent activities prescribed by the feedback.

A basic Take Action command displays the return code of the operation in a message box or a log file that is displayed after action completion. After you close this window, no further information is available for this action.

More information about Take Action commands

For more information about working with Take Action commands, see the *Tivoli Enterprise Portal User's Guide*.

For a list and description of the Take Action commands for this monitoring agent, see ("Special Take Action commands" on page 1471). See also the information in that section for each individual command.

Special Take Action commands

An Agent Builder monitoring agent can recognize and do special processing for a set of Take Action commands:

- SSHEXEC

For more information about creating these commands and including them in an Agent Builder monitoring agent project, see ("Creating workspaces, Take Action commands, and situations" on page 1337).

SSHEXEC action

Before you begin

For more information about Take Action commands, see ("Take Action commands reference" on page 1470).

About this task

The SSHEXEC action is recognized for a monitored application that has at least one SSH Script attribute group. It indicates that the command that follows the SSHEXEC keyword is remotely started on the SSH target system. The command is started with the credentials and privileges of the user that is configured to monitor the SSH target system. The command is run on the remote system that is represented by the Managed System Name.

Procedure

To include the Take Action command in a situation or workflow policy, use the following syntax for the system command:

```
SSHEXEC [Command]
```

For example:

```
SSHEXEC [ls &path]
```

Note: You can customize the command or portions of the command during invocation of the Take Action by using the Take Action arguments option with the *Command*.

Note: If the *Command* includes multiple arguments, then consider including the bracket parenthesis to enable invocation of the Take Action command with the **tacmd** command-line interface.

Accessibility features

Accessibility features assist users who have a disability, such as restricted mobility or limited vision, to use information technology content successfully.

Accessibility features

The web-based interface of IBM Cloud Application Performance Management is the Cloud APM console. The console includes the following major accessibility features:

- Enables users to use assistive technologies, such as screen-reader software and digital speech synthesizer, to hear what is displayed on the screen. Consult the product documentation of the assistive technology for details on using those technologies with this product.
- Enables users to operate specific or equivalent features using only the keyboard.
- Communicates all information independently of color.[1]

The Cloud APM console uses the latest W3C Standard, WAI-ARIA 1.0 (http://www.w3.org/TR/wai-aria/), to ensure compliance with US Section 508 (http://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/section-508-standards), and Web Content Accessibility Guidelines (WCAG) 2.0 . To take advantage of accessibility features, use the latest release of your screen reader in combination with the latest web browser that is supported by this product.

The Cloud APM console online product documentation in IBM Knowledge Center is enabled for accessibility. The accessibility features of IBM Knowledge Center are described at IBM Knowledge Center release notes .

Keyboard navigation

This product uses standard navigation keys.

Interface information

The Cloud APM console web user interface does not rely on cascading style sheets to render content properly and to provide a usable experience. However, the product documentation does rely on cascading style sheets. IBM Knowledge Center provides an equivalent way for low-vision users to use their custom display settings, including high-contrast mode. You can control font size by using the device or browser settings.

The Cloud APM console web user interface includes WAI-ARIA navigational landmarks that you can use to quickly navigate to functional areas in the application.

The Cloud APM console user interface does not have content that flashes 2 - 55 times per second.

Related accessibility information

In addition to standard IBM help desk and support websites, IBM has established a TTY telephone service for use by deaf or hard of hearing customers to access sales and support services:

TTY service 800-IBM-3383 (800-426-3383) (within North America)

IBM and accessibility

For more information about the commitment that IBM has to accessibility, see IBM Accessibility (www.ibm.com/able).

[1] Exceptions include some **Agent Configuration** pages of the Performance Management console.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBMproducts. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work
must include a copyright
notice as follows:
© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. 2014, 2015.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth in the following paragraphs.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's user name for purposes of session management, authentication, and single sign-on configuration. These cookies can be disabled, but disabling them will also likely eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.