1 Preface

Enterprise-wide information processing today comprises a large number of systems and applications. These are often no longer concentrated in a computer center but distributed over a number of different sites.

1.1 Central monitoring of decentralized systems via SNMP

Whereas the decentralized installation of servers used to be associated mainly with distributing operational responsibility, nowadays operational responsibility is increasingly becoming centralized. This presupposes that the distributed systems and applications can be monitored and controlled centrally over communication links and that on-site management of components (bridges, hubs, routers, servers etc.) "on site" can generally be replaced. The latter function is performed by central management platforms, which use "agents" to obtain up-to-date information on the components to be monitored. An agent is a piece of software which runs on the component being monitored to supply the necessary information on that component. The management platform and the agent communicate with each other using a defined protocol. The protocol which has emerged as a de facto standard is SNMP (Simple Network Management Protocol). SNMP enables a very heterogeneous inventory from different IT vendors to be integrated into a homogeneous network management system.

1.2 SNMP management for BS2000/OSD

With its SNMP management products for BS2000/OSD, Fujitsu Siemens Computers makes it possible for BS2000/OSD systems to also be included in this homogeneous network management system.

It offers the following components:

- Relating to SNMP agents:
  - SNMP Basic Agent
  - Numerous product-specific agents
- Relating to management platforms:
  - The applications: Console and Application Monitor, Performance Monitor, BCAM Monitor and Cluster Monitor.
  - A package for integrating the BS2000/OSD management system into Unicenter (Computer Associates).

1.3 Target group

This manual is aimed at network planners, administrators and operators as well as system administrators who integrate the BS2000/OSD systems into an SNMP-based network, system and application management, or those who wish to operate such a system. Prior knowledge of the BS2000/OSD operating system and the basic TCP/IP terms is assumed.

1.4 Structure of the manual

The present manual is structured in the following way:

- Chapter 2: An overview of SNMP

  This chapter provides an introduction to the SNMP architecture and an overview of the structure of the areas of application of SNMP and the functions it offers. It outlines the SNMP agent in BS2000/OSD and illustrates the product structure of the SNMP management system for BS2000/OSD which reflects the master-subagent principle of the SNMP agent. The chapter concludes with an overview of the user interfaces in the SNMP management system in BS2000/OSD.

- Chapter 3: Integrating BS2000/OSD into SNMP

  This chapter details the software requirements for the installation of SNMP management products in BS2000/OSD and describes how SNMP agents are installed.

- Chapter 4: SNMP Basic Agents for BS2000/OSD

  This chapter describes the areas of application, functionality, configuration and start and stop commands of the individual SNMP Basic Agents in BS2000/OSD.

- Chapter 5: Product-specific agents - functional enhancements in SNMP V6.0

  This chapter contains a complete description of the new HIPLEX subagent. It also describes the functional updates to the *open*UTM subagent since SSA-OUTM-BS2 V5.0A.

- Chapter 6: SNMP management

  This chapter outlines the three alternatives available for accessing management information, distinguished by the various system requirements at the management platform:

  – Access to the SNMP agent via the World Wide Web

  – Management applications

  – Integration in management platforms, such as Unicenter (Computer Associates)

- Chapter 7: Security considerations when using SNMP

  This chapter describes the security issues you need to be aware of when using SNMP.

1.5 Updates since the previous version

The following new features and functional extensions have been introduced with version V6.0 of the SNMP management system for BS2000/OSD:

- New subagents:
  - Event subagent
  - Scheduler subagent
  - HIPLEX subagent
- Functional extensions to the following subagents:
  - Console Monitor subagent
  - Application Monitor subagent
  - Subagent for *open*UTM
- New management application: Cluster Monitor
- Revision and functional extension of the following management applications:
  - Console Monitor, extended to include Application Monitor: Console and Application Monitor
  - Performance Monitor
- Further integration into CA Unicenter
  - Adapting to Unicenter NSM V3.0
  - Addition of a DSM policy to the subagent for *open*UTM
- Increased security

  The SNMP agents and the management applications Console and Application Monitor, Performance Monitor and Cluster Monitor all support the security concepts of SNMPv3, including authentication, authorization and access control for all queries or changes to management objects.

1.6 Notational conventions

This manual uses the following symbols and formatting to emphasize particularly important sections of text:

for general information

WARNING!
for warnings

Italics
: for file names, names of management windows and parameters, menu titles and menu items, as well as commands and variables included in continuous text.

<angled brackets>
: designate variables which have to be replaced by current values.

`fixed-width text`
: for the representation of system inputs and outputs and file names in examples.

command
: In the syntax description of commands, those parts that must be input unchanged (names of commands and parameters) are shown bold.

1.7 README file

Please see the product-specific README file for functional changes and updates to the current product version, if necessary. This file is stored on your BS2000/OSD computer under the file name `SYSRME.SBA-BS2.060.E`. Please ask your responsible system administrator for the user ID of the README file. The README file can be viewed with the `/SHOW-FILE` command or with an editor, or it can be printed out to a standard printer with the following command:

`/PRINT-DOCUMENT` *filename*`,LINE-SPACING=*BY-EBCDIC-CONTROL`

2 An overview of SNMP

SNMP stands for **S**imple **N**etwork **M**anagement **P**rotocol and was developed as a protocol for network management services in the TCP/IP internet. Originally, SNMP was only used to monitor and manage LAN components, such as bridges, routers and hubs, in heterogeneous networks with TCP/IP protocols. SNMP's range of application has since been extended to include system management, application management and even management of middleware products such as databases and transaction monitors.

Similarly to TCP/IP, where the protocol name does not just refer to the protocols themselves but to the infrastructure and framework of the whole TCP/IP network, i.e. the internet, the name SNMP does not just stand for the protocol but for the entire management system which is based on SNMP.

Benefits of SNMP

SNMP is no longer merely one management protocol among many, it is now *the* management protocol in TCP/IP networks. The reasons for this include:

– SNMP is a standard
– SNMP enjoys widespread popularity
– SNMP allows differentiated access
– SNMP is easy to implement

2.1 SNMP management architecture

SNMP employs a client /server architecture, where the management platform is the client and the management agents are the servers (see figure 1).

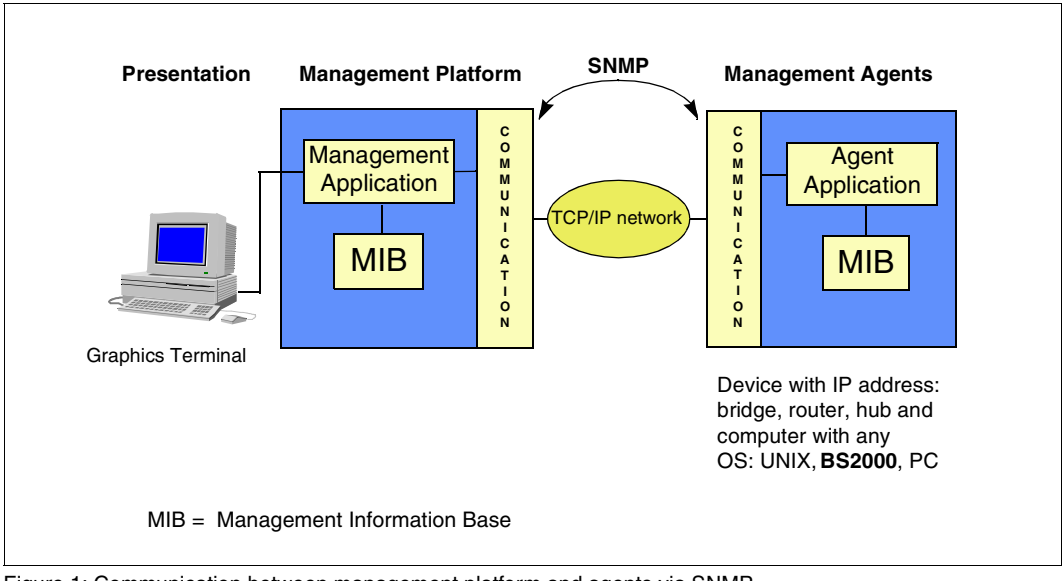

Figure 1: Communication between management platform and agents via SNMP

Management platform

The central component of an SNMP installation is the management platform. The management platform is a control center with graphics terminals, which enables a well-structured display of the components it manages and offers easy operation. The network and all of its components, systems and applications can be monitored and controlled from the management platform.

SNMP managers, also known as management applications, reside on the management platform. These communicate with the SNMP agents via SNMP over a TCP/IP network. Each managed component has an SNMP agent that provides the SNMP manager with current information about the component - on request or spontaneously. The initiative for controlling the activities mainly rests with the SNMP manager, which ensures that the components to be managed only have to cope with a small volume of management tasks.

SNMP manager

The SNMP manager is the software which generates the requests to the individual agents and sends them via SNMP to the corresponding agents.

SNMP managers receive two types of messages from the agents:

– Responses to their requests

– Traps. Traps are asynchronous messages which the agent sends to the SNMP manager in defined situations, unsolicited by the SNMP manager.

The SNMP manager displays the information received from the agent and can react to this information with its own actions. The display options range from the output of the values in a simple table to a visual representation of the monitored systems and applications in a network map with event reporting and specific alarm management.

SNMP agent

An SNMP agent is the software which receives, executes and responds to the requests sent by the SNMP manager. The agent has direct access to the part of the system or the component being monitored. In defined situations, the SNMP agents also send unsolicited asynchronous messages (traps) to the manager.

Modern SNMP agents, such as the agent used in BS2000/OSD, are structured according to a master-subagent principle (see section “Product structure” on page 12). In contrast to a monolithic SNMP agent, this makes it easier to start up and shut down subfunctions (subagents).

Management Information Base (MIB)

Every component which is to be managed and thus every subagent requires a separate MIB. In the MIB, the management-specific objects of the relevant component are defined and the object attributes are described. Object attributes include: object name, syntax, access rights and status.

The following types of MIBs are available:

– Standard MIBs, i.e. MIBs which have been adopted by standardization committees, especially internet committees. A typical example is the internet standard 17, the MIB-II (RFC1213) for TCP/IP networks.

– MIBs which represent a de-facto standard.

– Private MIBs which contain enhancements specific to the manufacturer.

The manufacturer supplies specific MIBs for many hardware and software components. For further information on the MIB, see the manual "SNMP Management V5.0".

Security mechanisms

Authorization for read or write access on the part of the SNMP manager is controlled by a "community name" (community string). The community name is included in every SNMP message and identifies the sender of the message as a member of a specific group, or community. Manager and agents may only communicate with each other if they belong to the same community.

This relatively simple model has been developed into a more comprehensive security concept with SNMPv3. This allows you, even if you are using SNMPv1 as a protocol, to use important SNMPv3 functions in the SNMP products for BS2000/OSD, such as:

– Selectively assigning access rights to MIB variables
– Defining access rights for a group of management platforms
– Detailed trap sending

2.2 SNMP agent in BS2000/OSD

Modern SNMP agents, like the ones used in BS2000/OSD, are structured according to a master-subagent principle. The functionality of the agent is distributed in the following way across a master agent and one or more subagents:

– The master agent performs the basic activities, such as handling the SNMP protocol, security functions, job assignment etc., centrally

– Each subagent is only responsible for a specific area of the monitored component. It only communicates with the SNMP master agent which performs the communication from/to the SNMP manager.

The subagents are independent and can be started and stopped at any point. This optimizes the performance, reliability and scalability of the whole SNMP system.

The master-subagent structure of the SNMP agent is reflected in the product structure of the SNMP management system for BS2000/OSD (see next section “Product structure”).

2.3 Product structure

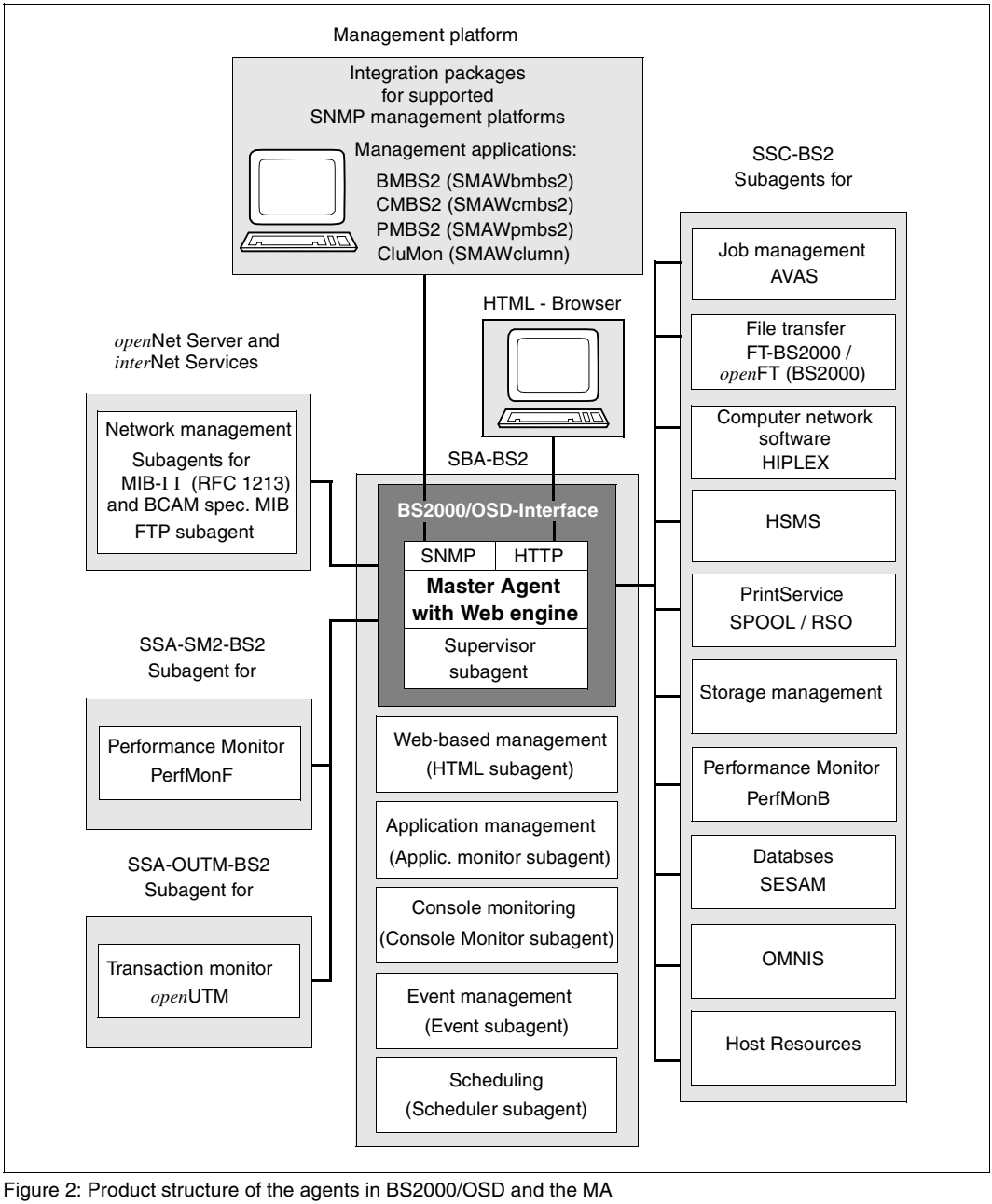

Figure 2: Product structure of the agents in BS2000/OSD and the MA

The SNMP agents are supplied in the following selectable units:

- SBA-BS2 (SNMP-Basic-Agent BS2000)
- SSC-BS2 (SNMP-Standard-Collection BS2000)
- SSA-SM2-BS2 (SNMP subagent for the Performance Monitor SM2)
- SSA-OUTM-BS2 (SNMP subagent for *open*UTM in BS2000/OSD)
- SNMP subagents in the products *open*Net Server and *inter*Net Services
- SNMP agent for the HNC (Highspeed Net Connector)

SBA-BS2 (SNMP-Basic-Agent BS2000)

The selectable unit SBA-BS2 V6.0 contains the basic agents:

– SNMP master agent
– Supervisor subagent
– Application Monitor subagent
– Console Monitor subagent
– Event subagent
– Scheduler subagent
– HTML subagent

Below is a brief description of the SNMP basic agents. For more detailed information on the functionality, configuration and operation of the individual basic agents, see chapter “SNMP Basic Agents for BS2000/OSD” on page 27. For a description of HTML subagents, see the manual "SNMP Management V5.0".

- The **master agent** is the BS2000/OSD communication partner for the management platform, which handles the SNMP protocol. It also controls the communication with the subagents and offers access to the SNMP group of the MIB-II (RFC1213) and the objects of other standardized SNMP MIBs (RFC 2272 - RFC 2275), thereby allowing monitoring of the system and the values relevant to SNMP. Furthermore, the master agent provides web access to information from the MIBs.
- The **Supervisor subagent** monitors the other subagents and the events notified by them.
- User applications, BCAM applications, DCAM Applications, tasks, job variables, BS2000/OSD subsystems are monitored by the **Application Monitor subagent**. It also monitors the log files in BS2000/OSD, POSIX and NFS. DCAM applications can be monitored cyclically. Logically associated objects in a business process can be grouped together by the Application Monitor Subagent and monitored either separately or as a group.

- The **Console Monitor subagent** is used for console monitoring. On the one hand, it allows you to forward console messages as traps and precisely define the quantity of the messages to be recorded. On the other, you can also issue BS2000/OSD commands from the management platform and query the results.

  The associated management application enables the display of the console messages and easy console operations for all integrated BS2000/OSD systems.

- The **Event subagent** provides a mechanism which periodically performs SNMP requests (SNMP GetRequests) on MIB objects of other subagents and triggers simple actions if specific conditions have been satisfied, e.g. that objects exist or that they have gone above or below certain threshold values. Both the current individual value and the difference from the last request can be evaluated.

- The **Scheduler subagent** provides a mechanism which makes changes to SNMP objects (SNMP SetRequests) periodically or at specific times. For periodic operations, the number of seconds between SNMP set operations is specified. Times are set by the month, day, weekday, hour and minute. It is thus possible, for instance, to perform a request every Monday at 6:00 or every last Friday of the month at 22:00.

- The **HTML subagent** enables the definition of custom pages for web access to management information of BS2000/OSD.

The selectable unit SBA-BS2 additionally contains three SDF commands for sending traps (see the manual "SNMP Management V5.0").

SSC-BS2 (SNMP-Standard-Collection BS2000)

A collection of subagents for BS2000/OSD-specific management tasks is supplied with SSC-BS2 V6.0.

The agents in the SNMP standard collection for BS2000 are briefly described below. For more detailed information on the functionality, configuration and operation of the individual agents, see the manual "SNMP Management V5.0". The HIPLEX subagent is described in detail on page 78 in the chapter “Product-specific agents - Functional enhancements in SNMP V6.0”.

- The **AVAS subagent** monitors the overall status of AVAS, the central processes and schedules, as well as the job networks and structure elements.
- The ***open*FT (BS2000)** subagent supplies information about FT system parameters and statistics of the session. It also has the additional functions of starting and stopping the FT, diagnosis control, changing the public key for encryption and changing the status of an FT partner.
- The **subagent for HIPLEX** supplies information about the current configuration in the HIPLEX cluster, as well as the status of the systems and switchover units involved in the cluster, and reports all relevant changes. The HIPLEX subagent sends traps on status changes.
- The **HSMS subagent** allows you to read and modify global HSMS data. It also supplies detailed information on HSMS tasks. The scope of the tasks can be restricted by the selection criteria "state" and "origin".
- The **subagent for spool and print services** monitors the SPOOL and RSO devices and supplies information about print jobs.
- The **subagent for storage management** supplies information about pubsets and disks. The subagent can also monitor selected or all pubsets and disks.
- The **Host Resources subagent** supplies information about the host, devices, pubsets, file systems and the installed software and notifies changes.
- The **OMNIS subagent** monitors data terminals, partners and applications and enables administration of OMNIS itself.
- The **subagent for managing SESAM/SQL databases** supplies information on SESAM/SQL databases and SESAM/SQL DBHs with which these databases are processed (RDBMS MIB according to RFC1697).
- The **subagent for basic performance monitoring with SM2 (PerfMonB)** supplies average values for monitoring CPU utilization and I/O rates.

SSA-SM2-BS2 (SNMP subagent for the Performance Monitor SM2)

The SM2-based SSA-SM2 performance subagent supplies basic information on SM2 itself, i.e. subsystem status, version and measurement interval and sample cycle sizes. The actual measurement values correspond to the familiar SM2 report groups and provide information on

- CPU utilization
- I/O activities
- main memory and virtual address space utilization
- main memory occupation by the four standard task categories
- input/output operations to peripheral devices during a measurement interval
- application-specific data from UTM applications
- resource utilization values of separate tasks

The display of the returned measurement values on the management platform can be supported by the management application PMBS2 which is on the TransView CD-ROM; these also enable the simultaneous monitoring of several BS2000/OSD systems.

For further information on the functionality, configuration and operation of the SNMP subagent for the Performance Monitor SM2, see the manual "SNMP Management V5.0".

SSA-OUTM-BS2 (SNMP subagent for *open*UTM in BS2000/OSD)

The *open*UTM subagent SSA-OUTM-BS2, which is also an additive subagent, offers the following services:

- monitoring and control of selected *open*UTM applications
- information on system parameters, physical and logical terminals, terminal pools, transaction codes, transaction classes, user data, connections and statistic data
- modifying application properties and system parameters
- locking and unlocking of UTM data terminals
- terminating an *open*UTM application

For *open*UTM in Reliant UNIX, the *open*UTM subagent in the product SSA-OUTM-SX is provided.

For detailed information on the functionality, configuration and operation of the SNMP subagent for *open*UTM in BS2000/OSD, see the manual "SNMP Management V5.0".

For information on the functional enhancements to the SNMP subagent for *open*UTM in BS2000/OSD, see page 85 onwards, in the chapter “Product-specific agents - Functional enhancements in SNMP V6.0”.

SNMP subagents for *open*Net Server and *inter*Net Services

A **MIB-II subagent in accordance with RFC 1213** is available for network management here.

The following are also offered:

– **BCAM subagent** (returns information on BCAM-specific settings and values)

– **FTP subagent** (SNMP subagent for the FTP server)

For further information on the functionality, configuration and operation of the SNMP subagent for *open*Net Server and *inter*Net Services, see the manual "SNMP Management for *open*NetServer and *inter*Net Services".

The following products are offered as supplements:

– TransView SNMP proxy agent for BS2000/PDN (TV-SPBP)
– HNC with integrated SNMP agent

For further information, see the manual "SNMP Management V5.0" and the corresponding product-specific manuals.

2.4 User interfaces

With the standard protocol SNMP, BS2000/OSD systems can be connected to any management platform which supports SNMP. This is the case for all management platforms commonly available on the market. The management platforms of the various manufacturers offer a variety of features. The strategic management platform Unicenter offered by Computer Associates (CA) is recommended by Fujitsu Siemens Computers and is aligned universally and equipped with a sophisticated alarm management system with numerous options for linking reactions to events.

Integration packages

Fujitsu Siemens Computers offers integration packages (SMBS2 and SMAWsmbs2) for CA Unicenter, which enable BS2000/OSD to automatically be integrated into these management platforms. This integration package includes extensions to the interface.

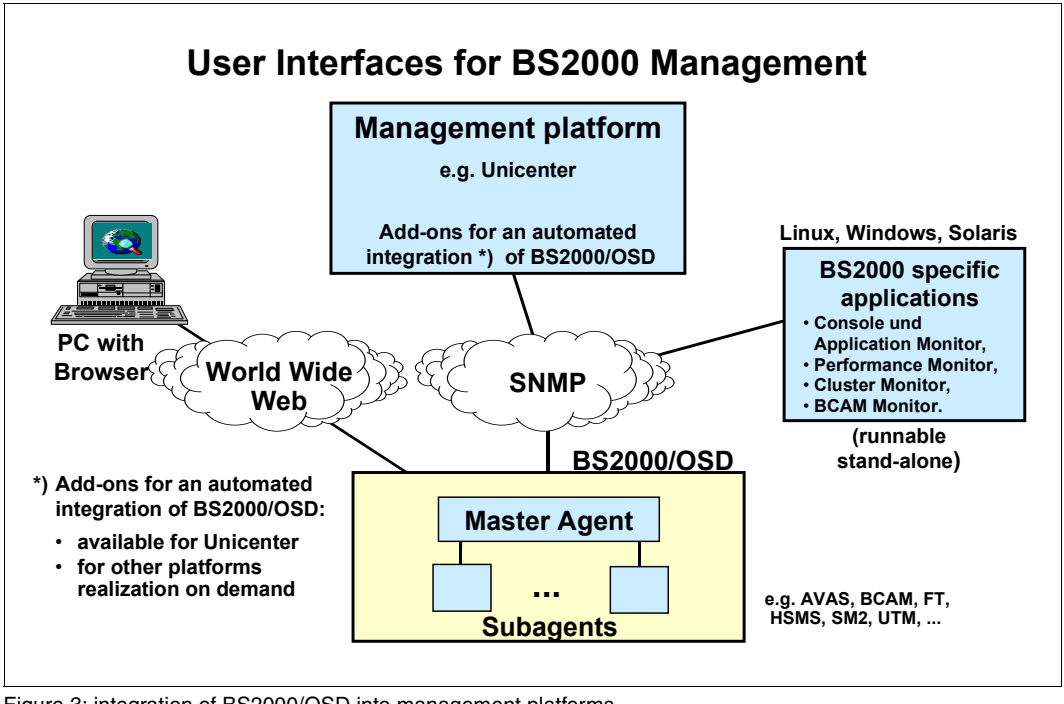

Figure 3: integration of BS2000/OSD into management platforms

Overview of integration packages

The following integration packages are offered for the strategically supported management platform CA Unicenter:

Integration packages	Related agents	Operating system
SMAWsmbs2	all subagents	Solaris
SMBS2	all subagents	Windows

Management applications

Besides the integration packages, Fujitsu Siemens Computers offers separate management applications for special subagents tailored to the specific attributes and tasks of the particular subagent, in the products BS2-SNMP-SO, BS2-SNMP-LX and BS2-SNMP-WIN. These management applications augment and enhance the representation and handling of the existing management platform. They can also be used separately on a Linux, Solaris or Windows system.

Overview of management applications

The following special management applications are offered:

Management applications	Package name	Related agents	Management platform	Operating system
BCAM Monitor BMBS2	BMBS2 for Solaris: SMAWbmbs2	BCAM subagent, MIB-II subagent *open*Net Server	standalone / integrated	Solaris / Linux / Windows
Console and Application Monitor CMBS2	CMBS2 for Solaris: SMAWcmbs2	Console Monitor subagent (SBA-BS2) Application Monitor subagent (SBA-BS2)	standalone / integrated	Solaris / Linux / Windows
Performance Monitor PMBS2	PMBS2 for Solaris: SMAWpmbs2	Performance Monitor subagent (SSA-SM2-BS2)	standalone / integrated	Solaris / Linux / Windows
Cluster Monitor CluMon	CluMon for Solaris: SMAWclumn	HIPLEX subagent (SSC-BS2)	standalone / integrated	Solaris / Linux / Windows

Web access to management information

Besides access to traditional SNMP management applications, the master agent provides access to management information via web browser on World Wide Web (WWW). For further information on web access, see the section “Web access to the BS2000/OSD management” on page 92 and the manual "SNMP Management V5.0".

3 Integrating BS2000/OSD into SNMP

BS2000/OSD-SNMP management consists of the following products for use in BS2000/OSD:

– SBA-BS2 V6.0

– SSC-BS2 V6.0

– SSA-SM2-BS2 V5.0

– SSA-OUTM-BS2 V5.0B

BS2000/OSD-SNMP management also includes packages for the management side, which are supplied on a separate CD-ROM along with the SBA-BS2 product, or can be downloaded from the Internet.

The SNMP agents are hardware-independent. They run on all central processing units (including RISC- and SPARC-based models) supported by BS2000/OSD as of V2.0 or by OSD-SVP as of V2.0.

3.1 Software requirements

Software requirements for SBA-BS2

SNMP-Basic-Agent-BS2000 V5.0 requires the following software:

– BS2000/OSD-BC ≥ V2.0 or OSD-SVP ≥ V2.0
– POSIX-BC ≥ V1.0*
– SOCKETS(POSIX) ≥ 1.0*
– IMON ≥ V 2.0*
– SDF-P-BASYS V2.0B*
– JV ≥ V11.2 (optional)

Components marked with an asterisk (*) are included in BS2000/OSD-BC.

Software requirements for SSC-BS2

SNMP-Standard-Collection V6.0 requires the following software:

– BS2000/OSD-BC ≥ V2.0 or. OSD-SVP ≥ V2.0
– SBA-BS2 V6.0
– AVAS ≥ V3.0
– FT-BS2000 V6.2 bzw. *open*FT (BS2000) ≥ V6.0
– SPOOL ≥ V3.0*
– RSO ≥ V2.4
– HSMS ≥ V3.1
– OMNIS ≥ V8.1
– SDF-P-BASYS ≥ V2.0B*/**
– SESAM/SQL-Server ≥ V2.1B 0***
– SM2 ≥ V11.2
– JV ≥ V11.2
– HIPLEX-MSCF ≥ V1.0, HIPLEX-AF ≥ V3.0

Components marked with an asterisk (*) are included in BS2000/OSD-BC.
Components marked with two asterisks (**) are required for the PrintService subagent.
The marking *** means that if a host is to monitor several DBHs, use of SESDCN is also required.

Software requirements for SSA-SM2-BS2

SSA-SM2-BS2 requires SBA-BS2 V5.0 or V6.0 and SM2 as of V11.2 in BS2000/OSD-BC ≥ V2.0 or OSD-SVP ≥ V2.0.

Software requirements for SSA-OUTM-BS2

SSA-OUTM-BS2 requires *open*UTM ≥ V3.3 and the corresponding version of UTM-D-SP. BS2000/OSD-BC ≥ V2.0 and SBA-BS2 V5.0 or V6.0 are also required.

Software requirements for the subagents for *open*Net Server and *inter*Net Services

To use the MIB II subagent, SBA-BS2 as of V3.1 and DCAM as of V13.0 or *open*Net Server V1.0 are required. The BCAM subagent (private MIB) runs as of DCAM V14.0.

Software requirements for the integration packages SMBS2 and SMAWsmbs2

The software requirements for the integration packages SMBS2 and SMAWsmbs2 are described in the section “Requirements for integration” on page 108.

3.2 Installation of SNMP agents in BS2000/OSD

The products SBA-BS2 and SSC-BS2 are installed on the BS2000/OSD host, as are the additive subagents SSA-SM2, SSA-OUTM-BS2 and the subagents for *open*Net Server and *inter*Net Services.

SBA-BS2, SSC-BS2, SSA-SM2-BS2 and SSA-OUTM-BS2 are installed by the software delivery and information system SOLIS2. SOLIS2 installation includes, where necessary, BS2000/OSD-specific tasks, such as subsystem catalog entries etc.

It must be noted that an entry for the SNMP subsystem is generated in the system catalog.

Please ensure that the internal communication between master and subagents is carried out via port number 3161. The BCAM dynamic port number assignment should, in particular, start with a higher value. The default BCAM value is 4096.

Deleting the SINLIB after installation leads to errors as the agents also require the SINLIB during operation.

The following sections describe the relevant installation steps for the agent side. The installation procedures for the SNMP manager and the management applications are described in the chapter “SNMP management” on page 91.

3.2.1 Installing SBA-BS2 and SSC-BS2

The POSIX subsystem must be running. The executable agents of SBA-BS2 are in SINLIB.SBA-BS2.060. This also contains all elements that must be installed in the UFS. Installation is carried out under the SYSROOT or TSOS ID (UID=0,GID=0) with the POSIX installation tool:

```
/CALL-PROCEDURE *LIB(LIB=$TSOS.SINPRC.POSIX-BC.<version>,ELEMENT=POSINST)
```

Function: Install POSIX program package
Product name: SBA-BS2
Product version: 060

SINLIB.SSC-BS2.060 contains the executable agents of SSC-BS2 and all elements which must be installed in the UFS. Installation is carried out under the SYSROOT or TSOS ID (UID=0,GID=0) with the POSIX installation tool:

```
/CALL-PROCEDURE *LIB(LIB=$TSOS.SINPRC.POSIX-BC.<version>,ELEMENT=POSINST)
```

Function: Install POSIX program package
Product name: SSC-BS2
Product version: 060

3.2.2 Installing SSA-SM2-BS2

Installation is carried out under the SYSROOT or TSOS ID (UID=0, GID=0) with the POSIX installation tool:

```
/CALL-PROCEDURE *LIB(LIB=$TSOS.SINPRC.POSIX-BC.<version>,ELEMENT=POSINST)
```

Function: Install POSIX program package
Product name: SSA-SM2-BS2
Product version: 050

3.2.3 Installing SSA-OUTM-BS2

Installation is carried out under the SYSROOT or TSOS ID (UID=0, GID=0) with the POSIX installation tool:

```
/CALL-PROCEDURE *LIB(LIB=$TSOS.SINPRC.POSIX-BC.<version>,ELEMENT=POSINST)
```

Function: Install POSIX program package
Product name: SSA-OUTM-BS2
Product version: 050

3.2.4 Version upgrading

The following notes on version upgrading supplement the information contained in the preceding sections on installation.

Upgrading from an older version of SBA-BS2 to V6.0

Version upgrade installations are also carried out with IMON or by making the SYSSII file (IMON V2.0). Libraries can be read into the desired ID without problems occurring as different version designations preclude conflicts with the previous versions.

The previous version of the syntax file must be replaced with that of Version 6.0. The agents should be terminated when performing this task, since the agents of the previous version cannot be terminated using the STOP command as the version of the agent must match that of the associated command program.

The file *snmpd.cnf* in */etc/snmp/agt* must be extended to include the customized entries. The master agent must be stopped for this as it overwrites the configuration file when it is terminated.

3.2.5 Deinstallation

Deinstallation is also carried out under the SYSROOT or TSOS ID (UID=0, GID=0) with the POSIX installation tool:

```
/CALL-PROCEDURE *LIB(LIB=$TSOS.SINPRC.POSIX-BC.<version>,ELEMENT=POSINST)
```

Function: Deinstall POSIX program package
Product name: see corresponding in section “Installing ...” on page 25.
Product version: <prod-version>

<prod-version> refers to the version number of the program package to be deinstalled.

4 SNMP Basic Agents for BS2000/OSD

The basic product SNMP-Basic-Agent BS2000, which contains the SNMP Basic Agents, is a prerequisite for SNMP management in BS2000/OSD.

The following SNMP Basic Agents exist:

– Master agent
– Supervisor subagent
– Application Monitor subagent
– Console Monitor subagent
– Event subagent
– Scheduler subagent

The SNMP Basic Agent with its SNMP subagents in BS2000/OSD can in principle be connected to all management platforms with the SNMP protocol.
The standardized and the BS2000/OSD-specific MIBs in ASN.1 format are the basis for this. These are described in detail in the manual "SNMP Management V5.0", chapter "Functions of the BASIC AGENT".

Supplementing this are special products which facilitate integration (see chapter “SNMP management” on page 91).

4.1 Master agent

The master agent forms the SNMP agent's interface to the network and thus to the management platforms.

4.1.1 Functionality of the master agent

The master agent performs the following functions:

- Handling the SNMP protocol and communicating over a TCP/IP network with the SNMP manager on the management platform
- Checking access authorization
- Forwarding requests from the SNMP manager to the responsible subagents
- Forwarding the responses and traps of the subagents to the SNMP manager

The master agent also enables management information to be accessed via the World Wide Web (WWW). Information provided by the subagents can thus be queried and modified both by traditional SNMP management applications and web browsers.

User-specific subagents can also be connected to the master agents. In its functions as a central management entity and SNMP protocol machine, the master agent implements objects in the system group, the MIB-II SNMP group and the SNMP framework.

The following information belongs to the system group and the MIB-II SNMP group:

– Run time of the agent

– Name and type of the system

– Number of incoming and outgoing packages

– Number of different protocol errors

– Number of security infringements (e.g. attempts to query an agent with an incorrect community name)

4.1.2 Configuring the master agent

The configuration file for the master agent is the file *snmpd.cnf*. This is located in the POSIX file system in the directory */etc/snmp/agt*. Besides the parameters for the security configuration (see manual "SNMP management V5.0"), the configuration file *snmpd.cnf* also contains the Initial System Group and optional start statement for the Supervisor subagent.

The *snmpd.cnf* file should only be edited when the master agent has been stopped, since the master agent overwrites the configuration file when terminated.

Initial System Group

sysDescr	
sysLocation	Fujitsu Siemens Computers Mch-P *
sysContact	Help Desk *
sysObjectID	1.3.6.1.4.1.231.1.6
MAX_PDU_TIME	Master agent wait time for a response from the subagent before it rejects the request.
MAX_THREADS	Specifies the maximum number of threads which can be processed simultaneously. It is recommended to select a number which is double the number of subagents as the subagents can each only process one request.
MAX_OUTPUT_WAITING	Specifies the number of bytes which can be stored as messages from the master before an overflow occurs.
MAX_SUBAGENTS	Defines the maximum number of subagents which may connect to the master agent.
RETRY_INTERVAL	RETRY_INTERVAL is currently not used.
snmpEnableAuthenTraps	2 : no authentication failure traps are sent 1 : authentication failure traps are sent
subagent	If the supervisor subagent is to be started, the name of the library must be entered here: [:<catid>:] [$<userid>.]SYSLNK.SBA-BS2.060 or for RISC machines: [:<catid>:] [$<userid>.]SRMLNK.SBA-BS2.060 or for SPARC: [:<catid>:] [$<userid>.]SPMLNK.SBA-BS2.060

Initial System Group default setting

* Please modify only the sysLocation and sysContact values to suit your requirements, the sysObjectID value should remain unchanged.

4.1.3 Starting / stopping the master agent

Before the master agent is started for the first time, the */etc/srconf/agt/snmpd.cnf* file in BS2000/OSD must be matched to the configuration used (see page 29).

As with all other agents, the master agent should be started in the background, otherwise the shell will be blocked.

Starting the master agent in BS2000/OSD:

/START-SNMP-MASTER
VERSION=***STD** / <product-version> , **MONJV**=***NONE** / <filename 1 .. 54 without-gen-vers> , **CPU-LIMIT**=***STD** / <integer 1 .. 32767> / ***NO** , **JOB-CLASS**=***STD** / <name 1 .. 8> , **TIMER-INTERVAL**= 5 / <integer 1 .. 32767>

or in the POSIX shell with:

```
snmpdm
```

Stopping the master agent in BS2000/OSD:

/STOP-SNMP-MASTER
VERSION=***STD** / <product-version>

or in the POSIX shell with:

```
snmpdmcmd T
```

Description of operands:

VERSION=*STD / <product-version>
Defines the version of the agent to be started or stopped. This parameter is currently not used.

MONJV=*NONE / <filename 1 .. 54 without-gen-vers>
Name of the job variable that is to monitor the agent. If *NONE is specified (default value), there is no monitoring per job variable.

CPU-LIMIT=*STD / <integer 1 .. 32767> / *NO
Maximum CPU runtime in seconds. If *STD is specified, the generated default value is used.

JOB-CLASS=*STD / <name 1 .. 8>
Job class with which the agent is started. If *STD is specified, the generated standard job class is used.

TIMER-INTERVAL=5 / <integer 1 .. 32767>
Interval at which the agent checks for requests from the command program. The default interval is set to five seconds.
The supervisor subagent uses the interval to check its subagent table. If no message is received from the subagent in the last five minutes, the supervisor checks this subagent with a query.

4.2 Supervisor subagent

The task of the Supervisor subagent is to monitor all subagents which are connected to the master agent. The supervisor subagent is linked particularly closely to the master agent and registers all events which the master agent receives from the other subagents.

4.2.1 Functionality of the Supervisor subagent

The Supervisor subagent checks at regular intervals whether the other subagents can be reached. It sends a trap to the SNMP manager in the following cases:

– If a subagent logs on or off at the master agent
– If a subagent can no longer be reached

Only a single instance needs to be polled per BS2000/OSD system. If the master agent with the Supervisor subagent is active, the Supervisor subagent returns the status of all other subagents. This significantly reduces the network load arising from active monitoring by the SNMP manager.

The Supervisor subagent also returns the following values for every subagent which is logged on:

– Status of the subagent (*active*, *disconnected*, *undefined*)
– Time of logon
– Time of last communication
– Number of requests answered
– Number of traps sent
– A supported OID (Object Identifier)

4.2.2 Configuring the Supervisor subagent

When a subagent logs on at the master agent, it informs the master agent of all the Object Identifiers (OID) it supports. The Supervisor subagent then identifies the newly logged on subagent via one of these OIDs.

This OID, and a name for the subagent, can be defined in the file */etc/snmp/agt/supervis.cnf*.

The entry is structured as follows:

<name> <oid>

<name>
Name of the subagent to be monitored by the Supervisor subagent.

<oid>
OID of the subagent to be monitored by the Supervisor subagent.

If the file */etc/snmp/agt/supervis.cnf* does not exist, the smallest OID which is supported is used.

Example

```
AppMon      1.3.6.1.4.1.231.2.23.5.3.0
```

4.2.3 Starting / stopping the Supervisor subagent

The Supervisor subagent is always started or stopped at the same time as the master agent. There is therefore no separate start or stop command for the Supervisor subagent. The Supervisor subagent is started by a corresponding entry in the file */etc/snmp/agt/snmpd.cnf*. As long as this entry exists, the Supervisor subagent is always automatically started and stopped in conjunction with the master agent.

4.3 Application Monitor subagent

The Application Monitor subagent is a universal agent and not assigned to any specific BS2000/OSD component.

4.3.1 Functionality of the Application Monitor subagent

With the Application Monitor subagent, the following entities (objects) can be monitored, i.e. their status and attributes can be queried:

- User applications
- BCAM applications
- DCAM applications
- Subsystems
- Job variables
- Log files

The Application Monitor subagent can also send each change as an unsolicited trap to a management platform. In this way, user applications and tasks can be monitored. Entries in a specific file can also be sent as a trap. In addition, you can manage groups of related applications as an entity (object). You can control the type and scope of application monitoring with a configuration file. You inform the Application Monitor subagent of the name of the configuration file in the start command.

Both options, trap and request, allow a universal application monitoring system to be incorporated into the alarm management system of a management platform.

There is a separate management application for the Application Monitor subagent, the Console and Application Monitor (see page 99).

4.3.2 Configuring the Application Monitor subagent

4.3.2.1 Statements for the configuration file

The configuration file contains information as to which applications, tasks, subsystems, job variables and log files are to be monitored. Up to 256 user applications, BCAM applications, job variables and log files can be monitored, as well as 128 DCAM applications. The user and BCAM applications and tasks to be monitored must be started with job variables. There is no limit to the number of subsystems that can be monitored.

The entries in the configuration file are generated using SDF statements. The //REMARK can be used to store comments in the configuration file. The last statement in the file must always be //END. Statements that come after the END statement are ignored.

Monitoring	Statement	Page
Application	//ADD-APPLICATION-RECORD	39
DCAM application	//ADD-DCAM-APPLICATION-RECORD	40
Subsystem	//ADD-SUBSYSTEM-RECORD	42
Log file	//ADD-LOG-FILE-RECORD	43
Job variable	//ADD-JV-RECORD	45
Group of associated applications	//DEFINE-OBJECT	47
Trap format	//DEFINE-TRAP-FORMAT	49
Monitoring log	//SET-TIMER-OPTIONS	50

Example 1

The following example can also be found in the SINLIB.SBA-BS2.060 library:

```
//REMARK Application Monitor, SDF-Configuration File
//REMARK
//REMARK Trap Format
//DEFINE-TRAP-FORMAT TYPE = (*GENERIC, *TVCC)
//REMARK
//REMARK Application Monitoring, Type BCAM
//ADD-APPLICATION-RECORD  -
//        APPLICATION-NAME = ANW1  -
//       ,VERSION = V1.0  -
//       ,TYPE = *BCAM  -
//       ,JV-NAME = MONJV  -
//       ,TRAP-CONDITION = (A, R)  -
//       ,WEIGHT=10  -
//
//REMARK Application Monitoring, Type USER
//ADD-APPLICATION-RECORD  -
//        APPLICATION-NAME = Applikation1  -
//       ,VERSION = V01.0A00  -
//       ,TYPE = *USER  -
//       ,JV-NAME = MJV1  -
//       ,TRAP-CONDITION = A  -
//       ,WEIGHT=5   -
//       ,ACKNOWLEDGE = *YES -
//
//ADD-APPLICATION-RECORD  -
//        APPLICATION-NAME = Applikation2  -
//       ,TYPE = *USER  -
//       ,JV-NAME = MJV2  -
//       ,TRAP-CONDITION = (T, A)  -
//
//REMARK Subsystem Monitoring
//ADD-SUBSYSTEM-RECORD  -
//        NAME = EDT  -
//
//ADD-SUBSYSTEM-RECORD  -
//        NAME = MAREN  -
//       ,VERSION = 08.1  -
//       ,TRAP-CONDITION = *NONE  -
//
//REMARK File Monitoring
//ADD-LOG-FILE-RECORD  -
//        NAME = /tmp/logfile1  -
//       ,APPLICATION-NAME = Datei1  -
```

```
//       ,MONITORING = *NO  -
//       ,FORMAT = *EBCDIC  -
//
//ADD-LOG-FILE-RECORD  -
//        NAME = $HUGO.LOGFILE2  -
//       ,MONITORING = *NO  -
//       ,FORMAT = *EBCDIC  -
//       ,PATTERN = '*important*' -
//
//REMARK Jobvariables
//ADD-JV-RECORD  -
//        JV-NAME = JOBVAR  -
//       ,PATTERN = ('*terminated*', '[1-5]00*')  -
//
//REMARK DCAM Application
//ADD-DCAM-APP A-NAME=d3str10$,HOST=camilla2,KEEP-CONNECTION=*NO -
//
//ADD-DCAM-APP A-NAME=$CONSOLE,HOST=D017ZE00, -
//        MSG='@CONSOLE,TSOS,''@@@@@@@@'',V01' -
//       ,WEIGHT=99  -
//
//REMARK Object
//DEFINE-OBJECT OBJECT-NAME=OB1,BCAM-APPLICATION=ANW1,          -
//        LOG-FILE=(/tmp/logfile1),  -
//        MONITORING-TIME=*INTERVAL(START=3:00,STOP=18:11,EX=SUN)
//END
```

Example 2: Monitoring critical applications

You wish to monitor a specific application which is very important to you, so as to be kept informed at all times about potential failures. The application was started with the Monitor Job variable APPMONJV.

You enter your application into the configuration file of the Application Monitor subagent as below:

```
//ADD-APPLICATION-RECORD -
//   ,APPLICATION-NAMW=APP -
//   ,TYPE=*USER -
//   ,JV-NAME=APPMONJV -
//   ,TRAP-CONDITION=(A,R,T)
```

The Application Monitor subagent then registers any change to the Monitor Job variable APPMONJV. When the program is started, the Monitor Job variable is set at $R. The Application Monitor subagent forwards Monitor Job variable status change as a trap to the management platform.

Example 3: Monitoring a MAREN system

A MAREN system includes the following components:

– MAREN subsystem
– MARENCP control program
– MARENUCP automatic free tape allocation facility

Every VSN reserved by the automatic free tape allocation facility is stored in the TAPE.FILE.MAREN job variable.

The following definition of a "MAREN" object combines these components:

```
//DEFINE-OBJECT OBJECT-NAME = MAREN -
//   ,USER-APPLICATION = (MARENCP, MARENUCP) -
//   ,SUBSYSTEM = MAREN -
//   ,JV = TAPE.FILE.YES
```

4.3.2.2 Changing the configuration file during the current session

Changes to the current configuration file during a session can be made by the Application Monitor either by setting the *appMonConfFile* object or using the command**:**

```
/START-APPMONCMD
       x "readConfig <filename>"
```

POSIX:

```
appmoncmd x "readConfig <filename>"
```

If there are syntax errors in *appMonConfFile*, the original configuration is retained.

ADD-APPLICATION-RECORD

The //ADD-APPLICATION-RECORD statement states the BCAM and user applications to be monitored. Applications are taken to be mean programs or tasks.

//ADD-APPLICATION-RECORD
APPLICATION-NAME = <composed-name_1 .. 54_with-underscore>
, **VERSION** = ***NONE** / <product-version>
, **TYPE** = ***BCAM** / ***USER**
, **JV-NAME** = <filename_1 .. 54>
, **TRAP-CONDITION** = **A** / list-poss (6) : <name_1 .. 1>
, **WEIGHT**= **0 /** <integer 0 .. 999>
, **ACKNOWLEDGE** = ***NO** / ***YES**

APPLICATION-NAME=<composed-name_1 .. 54_with-underscore>
Defines the application which the subagent is to monitor.

VERSION=*NONE / <product-version>
Version number of the application.
Default value: *NONE

TYPE=*BCAM / *USER
Type of application.

JV-NAME = <filename_1 .. 54>
Job variable (MONJV), which is used to monitor the application or task.

TRAP-CONDITION=A / list-poss (6) : <name_1 .. 1>
States for which a trap is to be generated.

WEIGHT= 0 / <integer 0 .. 999>
Weight of the traps specific to the Application Monitor subagents. When sending a (generic) trap, the Application Monitor subagent supplies the specified value for the trap object *appMonWeight* and the trap number (see manual "SNMP Management V5.0"). If various weights are to be used in an application for various events, the associated //ADD-APPLICATION-RECORD statement must be specified several times in the configuration file.
Default value: 0

ACKNOWLEDGE=*NO / *YES
Specifies whether or not the trap must be acknowledged. Only Application Monitor-specific traps can be acknowledged.
Default value: *NO

ADD-DCAM-APPLICATION-RECORD

The statement //ADD-DCAM-APPLICATION-RECORD identifies the DCAM applications which are to be monitored cyclically. The monitoring interval for DCAM applications is normally 60 times the value of the timer setting, therefore generally 5 minutes. With the statement //SET-TIMER-OPTIONS (see page 50), you can set the monitoring interval to any multiple of the timer setting.

A maximum of 128 DCAM applications can be monitored.

//ADD-DCAM-APPLICATION-RECORD
APPLICATION-NAME = <name_1 .. 8>
, **HOST**= ***OWN** / <name1 .. 8>>
, **KEEP-CONNECTION** = ***YES** / ***NO**
, **MSG**= ***NONE** / <c-string> / <x-string>
, **TRAP-CONDITION** = list-poss (2) : ***NOT-AVAILABLE** / ***AVAILABLE**
, **WEIGHT**= **0 / <integer 0 .. 999>**
, **ACKNOWLEDGE**= ***NO** / ***YES**

APPLICATION-NAME=<name_1 .. 8>
Defines the DCAM application which the subagent is to monitor.

HOST=*OWN / <name1 .. 8>
Host on which the DCAM application is running
Default value: *OWN

KEEP-CONNECTION=*YES / *NO
Defines whether the connection is to be cleared down
Default value: *YES

MSG= *NONE / <c-string> / <x-string>
Connection message
Default value: *NONE

TRAP-CONDITION=*NOT-AVAILABLE / *AVAILABLE
Conditions under which a trap is generated.
Default value: *NOT-AVAILABLE

WEIGHT= 0 / <integer 0 .. 999>
Weight of the traps specific to the Application Monitor subagents. When sending a (generic) trap, the Application Monitor subagent supplies the specified value for the trap object *appMonWeight* and the trap number (see manual "SNMP Management V5.0").
Default value: 0

ACKNOWLEDGE= *NO / *YES
Specifies whether or not the trap must be acknowledged. Only Application Monitor-specific traps can be acknowledged.
Default value: *NO

ADD-SUBSYSTEM-RECORD

The statement //ADD-SUBSYSTEM-RECORD defines the subsystems to be monitored. The monitoring interval is normally five times the value of the timer setting, therefore generally 25 seconds. With the statement //SET-TIMER-OPTIONS (see page 50), you can set the monitoring interval to any multiple of the timer setting.

//ADD-SUBSYSTEM-RECORD
NAME = <structured-name 1 .. 8> / ***ALL** , **VERSION** = ***NONE** / <product-version> , **TRAP-CONDITION** = ***NONE** / list-poss (8) : ***CREATED** / ***NOT-CREATED** / ***IN-DELETE** / ***IN-CREATE** / ***IN-RESUME** / ***IN-HOLD** / ***NOT-RESUMED** / ***LOCKED** , **WEIGHT= 0 / <integer 0 .. 999>** , **ACKNOWLEDGE** = ***NO** / ***YES**

NAME=<structured-name 1 .. 8> / *ALL
Defines the subsystem which the subagent is to monitor.

VERSION=*NONE / <product-version>
Version number of the subsystem
Default value: *NONE

TRAP-CONDITION=*NONE / list-poss (8) : *CREATED / *NOT-CREATED / *IN-DELETE / *IN-CREATE / *IN-RESUME / *IN-HOLD / *NOT-RESUMED / *LOCKED
States for which a trap is to be generated.
Default value: *NONE

CAUTION!
If NAME=*ALL is specified, you should use TRAP-CONDITION=*NONE as otherwise performance problems may arise.

WEIGHT= 0 / <integer 0 .. 999>
Weight of the traps specific to the Application Monitor subagents. When sending a (generic) trap, the Application Monitor subagent supplies the specified value for the trap object *appMonWeight* and the trap number (see manual "SNMP Management V5.0"). If various weights are to be used in an application for various events, the associated //ADD-SUBSYSTEM-RECORD statement must be specified several times in the configuration file.
Default value: 0

ACKNOWLEDGE=*NO / *YES
Specifies whether or not the trap must be acknowledged. Only Application Monitor-specific traps can be acknowledged.
Default value: *NO

ADD-LOG-FILE-RECORD

The statement //ADD-LOG-FILE-RECORD defines the log files to be monitored. The Application Monitor subagent generally sends a trap for every change to a file (log file). It is possible to filter the traps and entries. With the statement //SET-TIMER-OPTIONS (see page 50), you can set the monitoring interval to any multiple of the timer setting.

//ADD-LOG-FILE-RECORD
NAME = <filename_1 .. 54> / <posix-pathname>
, **APPLICATION-NAME** = ***NONE** / <composed-name_1 .. 54_with-underscore>
, **MONITORING** = ***YES** / ***NO**
, **FORMAT** = ***EBCDIC** / ***ASCII**
, **PATTERN** = ***NONE** / list-poss (8) : <c-string_1 .. 256_with-lower-case>
, **WEIGHT**= **0 / <integer 0 .. 999>**
, **ACKNOWLEDGE** = ***NO** / ***YES**

NAME=<filename_1 .. 54> / <posix-pathname>
Defines the log file which the subagent is to monitor.

APPLICATION-NAME=*NONE / <composed-name_1 .. 54_with-underscore>
Name of the application.
Default value: *NONE

MONITORING=*YES / *NO
Specifies whether the log file is to be monitored.

FORMAT=*EBCDIC / *ASCII
Format of the log file.
Default value: *EBCDIC

PATTERN = *NONE / list-poss (8) : <c-string_1 .. 256_with-lower-case>
Specifies one or more search patterns. If no PATTERN is specified, all entries are recorded in a log file for each trap.
The following wildcards are permitted:

? : replaces any one character
* : replaces any character string
[s] : replaces precisely one character from the s string
[c1 - c2]: replaces any character from the range c1 to c2
The "\" character (backslash) must be used as the escape character for special characters.
A distinction is made between uppercase and lowercase letters.
Default value: *NONE

WEIGHT= 0 / <integer 0 .. 999>
Weight of the traps specific to the Application Monitor subagents. When sending a (generic) trap, the Application Monitor subagent supplies the specified value for the trap object *appMonWeight* and the trap number (see manual "SNMP Management V5.0").
Default value: 0

ACKNOWLEDGE=*NO / *YES
Specifies whether or not the trap must be acknowledged. Only Application Monitor-specific traps can be acknowledged.
Default value: *NO

ADD-JV-RECORD

The //ADD-JV-RECORD statement defines the job variables to be monitored. By default, the Application Monitor subagent sends each job variable modification as a trap. However, it is possible to filter the traps.

//ADD-JV-RECORD
JV-NAME = <filename_1 .. 54> , **APPLICATION-NAME** = ***NONE** / <composed-name_1 .. 54_with-underscore> , **PASSWORD** = ***NONE** / <c-string_1 .. 4 > / < x-string_1 .. 8> , **PATTERN** = ***NONE** / list-poss (8) : <c-string_1 .. 256_with-lower-case> , **WEIGHT**= **0 / <integer 0 .. 999>** , **ACKNOWLEDGE** = ***NO** / ***YES**

JV-NAME = <filename_1 .. 54>
Defines the job variable which the subagent is to monitor.

APPLICATION-NAME = *NONE / <composed-name_1 .. 54_with-underscore>
Name of the application.
Default value: *NONE

PASSWORD = *NONE / <c-string_1 .. 4 > / <x-string_1 .. 8>
Read password of the job variables.
Default value: *NONE

PATTERN = *NONE / list-poss (8) : <c-string_1 .. 256_with-lower-case>

Defines one or more search patterns. If no PATTERN is specified, all JV changes are notified per trap.
The following wildcards are permissible:

? : replaces any character
* : replaces any number of characters
[s] : replaces exactly one character in a string s
[c1 - c2]: replaces any character in the range c1 to c2
The backslash character "\" must be specified to invalidate special characters. A distinction is made between uppercase and lowercase.
Default value: *NONE

WEIGHT= 0 / <integer 0 .. 999>
Weight of the traps specific to the Application Monitor subagents. When sending a (generic) trap, the Application Monitor subagent supplies the specified value for the trap object *appMonWeight* and the trap number (see manual "SNMP Management V5.0").
Default value: 0

ACKNOWLEDGE = *NO / *YES
Specifies whether or not the trap must be acknowledged. Only Application Monitor-specific traps can be acknowledged.
Default value: *NO

DEFINE-OBJECT

Logically associated components in a process (applications, log files, subsystems and job variables) can be grouped together using the statement //DEFINE-OBJECT. All elements stated in the //DEFINE-OBJECT statement must also be defined in the configuration file with the corresponding //ADD... statements.

```
//DEFINE-OBJECT

OBJECT-NAME = <composed-name_1 .. 8_with-underscore>
, BCAM-APPLICATION = *NONE / list-poss(5): <composed_name_1 .. 54_with-underscore>
, USER-APPLICATION = *NONE / list-poss(5): <composed_name_1 .. 54_with-underscore>
, DCAM-APPLICATION = *NONE / list-poss(5): <name_1 .. 8>
, LOG-FILE = *NONE / list-poss(5): <filename_1 .. 54> / <posix-pathname>
, SUBSYSTEM = *NONE / list-poss(5): <structured-name_1 .. 8>
, JV = *NONE / list-poss(10): <filename_1 .. 54>
, MONITORING-TIME = *ALWAYS / *INTERVAL (...)
    *INTERVAL ( ...)
        , START-TIME = hh:mm
        , STOP-TIME = hh:mm
        , EXCEPT-DAYS = *NONE / list-poss(6): MON / TUE / WED / THU / FRI / SAT / SUN
, ACKNOWLEDGE= *NO / *YES
```

OBJECT-NAME = <composed-name_1 .. 8_with-underscore>
Name of the object.

BCAM-APPLICATION = *NONE / list-poss(5): <composed_name_1 .. 54_with-underscore>
BCAM applications which belong to this object.
Default value: *NONE

USER-APPLICATION = *NONE / list-poss(5): <composed_name_1 .. 54_with-underscore>
User applications which belong to this object.
Default value: *NONE

DCAM-APPLICATION = *NONE / list-poss(5): <name_1 .. 8>
DCAM applications which belong to this object.
Default value: *NONE

LOG-FILE = *NONE / list-poss(5): <filename_1 .. 54> / <posix-pathname>
Log files that belong to this object.
Default value: *NONE

SUBSYSTEM = *NONE / list-poss(5): <structured-name_1 .. 8>
Subsystems that belong to this object.
Default value: *NONE

JV = *NONE / list-poss(10): <filename_1 .. 54>
Job variables that belong to this object.

MONITORING-TIME = *ALWAYS / *INTERVAL (...)
Specifies the monitoring time.
Default value: *ALWAYS

***INTERVAL (...)**
Defines the monitoring interval. If STOP-TIME is greater than START-TIME, the hours after midnight are counted to the previous day when checking the EXCEPT-DAYS.

Example:

The monitoring time ranges from 20:00 to 3.00 hrs, except fro Saturday and Sunday. Monitoring therefore stops on Saturday at 3:00 in the morning and starts again on Monday at 20:00 in the evening.

START-TIME = HH:MM
Time when the object should be monitored

STOP-TIME = HH:MM
Time up to which the object should be monitored

EXCEPT-DAYS = *NONE / list-poss(6): MON / TUE / WED / THU / FRI / SAT / SUN
Weekdays on which the object is not to be monitored
Default value: *NONE

ACKNOWLEDGE=*NO / *YES
Specifies whether the trap must be confirmed.
Default value: *NO

DEFINE-TRAP-FORMAT

The //DEFINE-TRAP-FORMAT statement defines the trap format for the Application Monitor subagents.

//DEFINE-TRAP-FORMAT
TYPE = list-poss(2) ***GENERIC** / ***TVCC**

TYPE = list-poss(2) *GENERIC / *TVCC
Defines the trap format.
GENERIC: The Application Monitor-specific trap format is used.
TVCC: The TV-CC-specific trap format is used.
Default value: *GENERIC

SET-TIMER-OPTIONS

The Application Monitor subagent uses a timer. You specify the value for the timer in the START command of the Application Monitor subagent (see page 51). The statement SET-TIMER-OPTIONS defines the monitoring interval (polling factor). The monitoring interval defines the number of timer cycles which are to elapse before the next check is to be carried out.

//SET-TIMER-OPTIONS
FILES = **1** / <integer> , **SUBSYSTEMS** = **5** / <integer> , **DCAM-APPLICATIONS** = **60** / <integer>

FILES = 1 / <integer>
Defines the polling factor for files.
Default value: 1

SUBSYSTEMS = 5 / <integer>
Defines the polling factor for subsystems.
Default value: 5

DCAM-APPLICATIONS = 60 / <integer>
Defines the polling factor for DCAM applications.
Default value: 60

4.3.3 Starting / stopping the Application Monitor subagent

The Application Monitor subagent is a subagent that is started in the POSIX shell or in BS2000/OSD.

1. Starting in BS2000/OSD:

/START-SNMP-APPMON
VERSION=***STD** / <product-version> , **MONJV**=***NONE** / <filename 1 .. 54 without-gen-vers> , **CPU-LIMIT**=***STD** / <integer 1 .. 32767> / ***NO** , **JOB-CLASS**=***STD** / <name 1 .. 8> , **FILE-NAME**=***NONE** / <filename 1 .. 54 without-gen-vers> , **TIMER-INTERVAL** = 5 / <integer 1 .. 32767>

2. Starting in the POSIX shell:

```
appmonagt [-f <inputfile>]
          [-t <int>]
```

The Application Monitor subagent is terminated (independent of the environment in which it was started) in BS2000/OSD with:

/STOP-SNMP-APPMON
VERSION=***STD** / <product-version>

or in the POSIX shell with:

```
appmoncmd T
```

Description of operands:

VERSION=*STD / <product-version>
Defines the version of the agent to be started or stopped. This parameter is currently not checked.

MONJV=*NONE / <filename 1 .. 54 without-gen-vers>
Name of the job variable that is to monitor the agent. If *NONE is specified (default value), there is no monitoring per job variable.

CPU-LIMIT=*STD / <integer 1 .. 32767> / *NO
Maximum CPU runtime in seconds. If *STD is specified, the generated default value is used.

JOB-CLASS=*STD / <name 1 .. 8>
Job class with which the agent is started. If *STD is specified, the generated standard job class is used.

FILE-NAME=*NONE / <filename 1 .. 54 without-gen-vers>
You may specify a configuration file when you start the Application Monitor subagent (see page 35). If no configuration file is specified, all the subsystems recognized by BS2000/OSD when the Application Monitor subagent was started are monitored. The configuration file, which is defined by <filename> or <inputfile>, must be stored in the BS2000/OSD file system.

TIMER-INTERVAL=5 / <integer 1 .. 32767>
Interval at which the agent checks for requests from the command program. The default interval is set to five seconds.
File monitoring is carried out when the interval has expired.

4.4 Console Monitor subagent

Just like the Application Monitor subagent, the Console Monitor subagent is also a universal agent and not assigned to any specific product. It communicates with the BS2000/OSD console.

4.4.1 Functionality of the Console Monitor subagent

With the Console Monitor subagent, BS2000/OSD console messages can be captured, filtered according to specific criteria to reduce network load and sent as a trap to the management platform. Conversely, the administrator can issue BS2000/OSD console commands at a management platform and have the results of the command execution displayed.

Some management platforms allow data to be automatically extracted from a message and inserted into specific console commands which are then returned to the BS2000/OSD system as an automatic action.

There is a separate management application for the Console Monitor subagent: the Console and Application Monitor (see page 99).

4.4.2 Configuring the Console Monitor subagent

The Console Monitor has access to the console commands of $CONSOLE via UCON.

The following preparation is required to enable the Console Monitor to access the BS2000/OSD console:

– Configure operator ID <operator-id>

– Grant access authorization for the operator ID

Configure operator ID <operator-id>

```
/ADD-USER USER-ID=<operator-id>,                              -
          PROTECTION-ATTRIBUTE=*PAR(LOGON-PASSWORD=<pass>),   -
          ACCOUNT-ATTRIBUTES=*PAR(ACCOUNT=<account-nr>)
```

The logon attributes defined here must be specified in the Console Monitor start statement (see START-SNMP-CONSMON on page 61).

Grant access authorization for the operator ID

For operating with SECOS, access rights must additionally be granted for the operator ID to $CONSOLE:

```
/MOD-LOGON-PROTECTION USER-IDENTIFICATION=<operator-id>,     -
                      OPERATOR-ACCESS-PROG=*YES(PASSWORD-CHECK=*YES)
```

The class 2 system parameter NBBAPRIV must be set to the default value N.

4.4.2.1 Defining message filters

The Console Monitor subagent uses two filter options for selecting messages:

– positive message filter
selects messages to be sent to the management platform.

– negative message filter
selects messages not to be sent to the management platform.

Positive message filter

The following two filter options are available for selection of messages to be sent to the management platform:

– Routing code (assigned to each console message)
– Message key (uniquely identifies each message)

Routing code selection criterion

Each message is assigned a specific routing code. Operator roles contain the routing codes of the messages to be sent to the management platform. The operator roles are specified in the Console Monitor start statement (see START-SNMP-CONSMON on page 61).The following statements show you how operator roles are created and assigned to the operator ID. The SECURITY ADMINISTRATION privilege, which the user ID SYSPRIV has as default, is required for issuing the following statements.

Create the operator role:

```
/CREATE-OPERATOR-ROLE OP-ROLE=<op-role-name>,                -
                      ROUTING-CODES=.......
```

Assign the operator role to the operator ID:

```
/MODIFY-OPERATOR-ATTR USER-ID=<operator-id>,                 -
                      ADD-OPERATOR-ROLE=(<op-role-name1>,...,<op-role-namex>)
```

The operator ID must additionally be assigned the OPERATING privilege if SECOS is used:

```
/SET-PRIVILEGE PRIV=OPERATING,USER-ID=<operator-id>
```

Message code selection criterion

The codes of messages to be sent to the management platform are stored in the positive message filter file.

Three filter options are available with the following statements:
– *msgid*
– *QUESTION*
– *TYPIO*

The name of the message filter file is made known to the Console Monitor when it is started via the MSG-FILTER entry. The file name can be entered in the MIB *consMonMsgFilter* object during a session.

If no message filter file is specified when the Console Monitor subagent is started, all messages are output for which the routing code is specified in the operator role.

If the message filter file contains no key, or no valid key, no traps are sent to the management platform. It only makes sense to create an empty message filter file when you are using the HIPLEX OP agent to monitor the BS2000/OSD console messages but at the same time still wish to enter console commands with the aid of the Console Monitor.

The following name conventions apply to the message filter file:

/BS2/<file>	BS2000/OSD file
[:<catid>:]$<userid>.<file>	BS2000/OSD file
*POSIX(<file>)	POSIX file
/<path>/<file>	POSIX file
<file>	The deciding factor in this case is the environment in which the subagent was started.

Structure of the positive message filter

msgid

```
<msgid [wgt] [SOURCE=src] [DEVICE=dev] [PATTERN=/pat1[/..patx]] [ACKNOWLEDGE=YES]>
```

msgid Specifies a message code.

The following wildcards are permitted for message code entries:
? : replaces any character
* : replaces any number of characters
[s] : replaces exactly one character in a string s
[c1 - c2]: replaces any character in the range c1 to c2

The backslash character "\" must be specified for special characters.
A distinction is made between uppercase and lowercase.

wgt Specifies a message weight. A weight can be assigned to the message codes. The weight is prefixed to the actual message in the trap string. This allows the user to set the importance of messages himself and transmit this to the management platform. The message code is assigned the value 0 as default if no weight is specified.

The entry is expected as an integer in the range 0 - 999.

src Specifies a source name. The source is supplied with BS2-<source> in the trap string. The default value *BS2Console* is used if no value is specified. You can set an alarm to a specific object in the network map with this entry. The entry is alphanumeric in the range 1 - 12 (see manual "SNMP Management V5.0").

pat Specifies one or more search patterns (pattern).

? : replaces any character
* : replaces a sequence of characters of any length
[s] : replaces exactly one character from the string s
[c1 - c2]: replaces any character from the range c1 through c2

The character "\" (backslash) must be specified to invalidate special characters.
A distinction is made between upper case and lower case.

dev If DEVICE is specified, the Console Monitor subagent sends this trap with the DEVICE entry as Community (see manual "SNMP Management V5.0").

ACKNOWLEDGE=YES

If you specify ACKNOWLEDGE=YES, the subagent is informed that this trap must be acknowledged.

QUESTION

Question filters all messages that contain a question, i.e. expect an answer. If a question is encountered, the Console Monitor first checks whether a pattern of QUESTION entries matches. If not, the MSGID entries or the TYPIO entries are searched for the relevant message type.

```
<QUESTION [wgt] [SOURCE=src] [DEVICE=dev] [PATTERN=/pat1[/..patx]]
[ACKNOWLEDGE=YES]>
```

QUESTION Message key of a console query

wgt	see above
src	see above
dev	see above
pat	see above
ACKNOWLEDGE=YES	see above

Example:

`<QUESTION PATTERN=[0-9]*>` Selection of all questions that start with a digit.

TYPIO

TYPE I/O messages are a special case. These include, for example, messages sent to the BS2000/OSD console with /SEND-MSG. Their reception is also controlled via the message filter file. The entry for a TYPE I/O message is as follows:

```
<TYPIO [wgt] [SOURCE=src] [DEVICE=dev] [PATTERN=/pat1[/..patx]] [ACKNOWLEDGE=YES]>
```

wgt	see above
src	see above
dev	see above
pat	see above
ACKNOWLEDGE=YES	see above

Example:

```
<TYPIO PATTERN=/*abc*/xyz>
<TYPIO PATTERN=/Hello*>
<TYPIO PATTERN=/?\?*>
```

All TYPE I/O messages that contrain the string "abc", are only made up of "xyz", start with "Hello" or have a question mark as their second character, are sent to the management platform as a trap.

An example of a message filter can be found in the SINLIB.SBA-BS2.060 library.

Negative message filter

A negative message filter is also provided as of Console Monitor subagent. The message code of the console messages, which are not to be forwarded to the management platform are stored in the negative message filter file. Questions cannot be suppressed. Every message filter can optionally be extended by one or more search patterns. The MIB object *consMonNegMsgFilter* refers to the name of the negative message filter file. The name of the negative message filter file is defined with the SUPPRESS-MSG-FILE operand when the Console Monitor is started. This definition can only be modified when the Console Monitor is started, not during a session.

The length of the entry must not exceed 179 characters.

```
<msgid [PATTERN=/pat1[/..patx]]> [<msgid [PATTERN=/pat1[/..patx]]>] ...
```

pat see above

Trap format

The trap format is additionally defined in the message filter file:

```
TRAP-FORMAT=GENERIC / TVCC / ALL
```

GENERIC
: Only the trap relevant to the Application Monitor is used.
GENERIC is the default value

TVCC
: Only the TVCC trap format is used.

ALL
: Both trap formats are used.

4.4.2.2 Modifying the configuration file during operation

It is possible to modify the current message filter file during operation using the Console Monitor application either by setting the *consMonMsgfilter* object or by command.

```
/START-CONSMONCMD

       x "readConfig <filename>"
```

POSIX:

```
consmoncmd x "readConfig <filename>"
```

If the *consMonMsgFilter* file contains syntax errors, processing continues with the original message filter file.

Example: filtering console messages

The message EXC0858 should only be sent to the management platform if it contains neither the string "CLAQ" nor the string "TEST". The trap should be sent with the trap number 99 and have "Hardware" entered as its source.

This is done as follows:

- In the positive message filter, enter: `<EXC0858 99 SOURCE=Hardware>`
- In the negative message filter, enter: `<EXC0858 PATTERN = *CLAQ* / *TEST*>`

4.4.3 Starting / stopping the Console Monitor subagent

The Console Monitor subagent is started in the POSIX shell or in BS2000/OSD.

1. Starting in BS2000/OSD:

/START-SNMP-CONSMON
VERSION=***STD** / <product-version> , **MONJV**=***NONE** / <filename 1 .. 54> , **CPU-LIMIT**=***STD** / <integer 1 .. 32767> / ***NO** , **JOB-CLASS**=***STD** / <name 1 .. 8> , **OPERATOR-ID**= <name 1 .. 8> , **PASSWORD**=***NONE** / <c-string 1 .. 8> / ***SECRET** , **OPERATOR-ROLE**= **list-poss(10)** <name 1 .. 8> , **MSG-FILTER**=***NONE** / <filename 1 .. 54> / <posix-pathname> , **SUPPRESS-MSG-FILE** = ***NONE** / <filename 1 .. 54> / <posix-pathname> , **TIMER-INTERVAL** = 5 / <integer 1 .. 32767>

2. Starting in the POSIX shell:

```
consmonagt  -o <operid>
            [-t <int>]
            [-p <password>]
            [-f <msg-filter>]
            [-n <negative-msg-filter>]
            <op-role1> [,<op-role2>, ...., <op-role10>]
```

The Console Monitor is terminated (independent of the environment in which it was started) in BS2000/OSD with:

/STOP-SNMP-CONSMON
VERSION=***STD** / <product-version>

or in the POSIX shell:

```
consmoncmd T
```

Description of operands:

VERSION=*STD / <product-version>
Defines the version of the agent to be started or stopped. This parameter is currently not used.

MONJV=*NONE / <filename 1 .. 54 without-gen-vers>
Name of the job variable that is to monitor the agent.
If *NONE is specified (default value), there is no monitoring per job variable.

CPU-LIMIT=*STD / <integer 1 .. 32767> / *NO
Maximum CPU runtime in seconds.
If *STD is specified, the generated default value is used.

JOB-CLASS=*STD / <name 1 .. 8>
Job class with which the agent is started. If *STD is specified, the generated standard job is used.

OPERATOR-ID=<name 1 .. 8>
User ID with which the subagent logs on to $CONSOLE.

PASSWORD=*NONE / <c-string 1 .. 8> / *SECRET
Definition of the password which authorizes the subagent to access $CONSOLE.
The default value *NONE specifies that no password is required. *SECRET causes the field for password input to be blanked out.

OPERATOR-ROLE=list-poss(10) <name 1 .. 8>
Name of the operator role containing the relevant routing code for console monitoring.

MSG-FILTER=*NONE / <filename 1 .. 54> / <posix-pathname>
Name of the file (<filename> or <posix-pathname>) containing the relevant message code.
*NONE (default) means that no message code file is assigned.

SUPPRESS-MSG-FILE=*NONE / <filename 1 .. 54> / <posix-pathname>
The file defined by <filename> or <posix-pathname> contains the console message code to be suppressed.
*NONE (default) means that no file, containing message code to be suppressed, is assigned.

TIMER-INTERVAL=5 / <integer 1 .. 32767>
Interval at which the agent checks for requests from the command program. The default interval is set to five seconds.

4.5 Event subagent

With the Event subagent you can monitor MIB objects of other subagents with periodic SNMP queries and carry out simple actions, as soon as specific conditions (trigger tests) have been satisfied. Actions can be: sending traps or carrying out SNMP set operations.

The interaction of the master agent and Event subagent when a subagent is monitored is illustrated in figure 4.

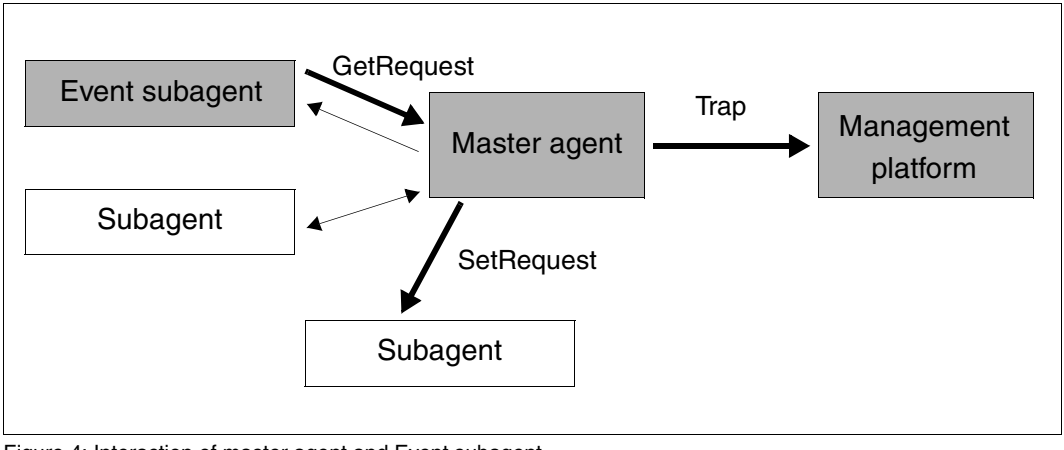

Figure 4: Interaction of master agent and Event subagent

4.5.1 Functionality of the Event subagent

The Event subagent implements the Event MIB (RFC2981). The Event MIB is divided into the following two sections, which define the Event subagent's range of functions:

– Trigger section
– Event section

These sections are configured with tables in a configuration file.

Trigger section

The trigger section defines the MIB objects to be monitored and the conditions (trigger tests) with which an event is triggered, e.g.

- **TriggerValueID** specifies the OID of the MIB object to be checked.
- **TriggerTest** specifies the conditions to be tested, e.g.
  - existence: whether the MIC object specified in TriggerValueID exists.
  - boolean: whether the value of this MIB object matches a value defined in the BooleanTable (TriggerBooleanValue).
  - threshold: whether this value is above or below a threshold value defined in the ThresholdTable.
- **TriggerSampleType** specifies whether the reference value is to be interpreted as an absolute value (absolute) or as a difference (delta) from a value determined during a previous query.
- **TriggerFrequency** specifies the time interval (in seconds) between two contiguous queries.

Depending on the type of test, entries may require a further table:

- **ExistenceTable:** Object exists / disappears / changes value.
- **BooleanTable:** Benchmark test of the object or delta value with the reference value.
- **ThresholdTable:** Object or delta value falls above / below threshold value.

If the test result is positive, a search is made in the EventTable for an appropriate entry for the action to be carried out.

Event section

The Event section defines which action - SNMP trap and/or SNMP SetRequest - is to be triggered in response to a successful trigger test, in the object EventAction:

- **notification** (SNMP trap): The OID of the trap to be sent is defined in the EventNotificationTable.
- **set** (SNMP SetRequest): The OID of the MIB object and the value to be set are defined in the EventSetTable.

Notifications

The Event MIB offers the following traps, which can be sent in response to triggered events.

- **TriggerSenseAlarm** reports that the trigger monitoring an object has been triggered.
- **TriggerRaisingAlarm** reports that the threshold value has been exceeded.
- **TriggerFallingAlarm** reports that the value has fallen below the threshold value.

Objects passed with a notification are entered in the **Objects Table** and specified in the corresponding entry in the:

– Trigger Table
– ExistenceTable / Boolean Table / Threshold Table and / or
– Notification Table

4.5.2 Configuring the Event subagent

You will find a description of how to configure the Event subagent as a commentary in the Event subagent configuration file sample which is supplied.

Examples

With MIB-I I : send a trap if:

- rUtilization = ((delta(iflnOctets)*8) / (ifSpeed * delta(seconds)) >x
- rErrorRate = ((delta(iflnErrors)) / (delta(seconds)) > x)
- comSecurit = (delta(snmplBadCommunityNames) > 0)

With Sar MIB: send a trap if:

- idleTime = (sm“TimeIOmachTabldleTime < x)

The first example requires the following configuration:

```
rUtilization = ((delta(iflnOctets)*8) / (ifSpeed * delta(seconds)) > 20%


mteTriggerEntry
         owner1                    TriggerOwner
         tInUtil                   Triggername
         "Utilization if1"         Comment
         20                        Test: threshold
         2                         Sampletype: delta
         1.3.6.1.2.1.2.2.1.10.1    OID to be monitored: ifInOctets of interface 1
         –                         –
         –                         –
         –                         –
         –                         –
         100                       Frequency: 100 sec
         –                         –
         –                         –
         1                         trigger enabled
         1                         active

mteTriggerThresholdEntry
(extends the trigger table via shared index 'owner1 tInutil')
          1                        StartUp: rising
          –                        –
          –                        –
          25000000                 DeltaRising: 25.000.000 InOctets/100 sec.
          –                        –
          –                        –
          –                        –
          –                        –
```

```
          -                          -
          -                          -
          -                          -
          owner1                     DeltaRisingEventOwner (Index in EventTable)
          evInUtil                   DeltaRisingEvent      (Index in EventTable)
          -                          -
          -                          -
          owner1                     TriggerOwner  (Index in TriggerTable)
          tInUtil                    TriggerName   (Index in TriggerTable)

mteEventEntry
          evInUtil                   Name        (Index in this EventTable)
          "Utilization if1"          Comment
          80                         Actions: notification
          1                          true
          1                          active
          owner1                     Owner       (Index in this EventTable)


mteEventNotificationEntry
(extends the EventTable via shared index 'owner1 evInUtil')
          1.3.6.1.2.1.88.2.0.2       Notification: mteTriggerRising
          owner1                     ObjectsOwner (Index in ObjectTable)
          oInUtil                    ObjectsName  (Index in ObjectTable)
          owner1                     EventOwner   (Index in EventTable)
          evInUtil                   EventName    (Index in EventTable)

mteObjectsEntry
          oInUtil                    Name      (Index in this ObjectTable)
          1                          Subindex  (Index in this ObjectTable)
          1.3.6.1.2.1.2.2.1.10.1     OID: ifInOctets of interface 1
          2                          -
          1                          active
          owner1                     Owner     (Index in this ObjectTable)

mteObjectsEntry
          oInUtil                    Name      (Index in this ObjectTable)
          2                          Subindex  (Index in this ObjectTable)
          1.3.6.1.2.1.2.2.1.5.1      OID: ifSpeed of interface 1
          2                          -
          1                          active
          owner1                     Owner     (Index in this ObjectTable)
```

4.5.3 Starting / stopping the Event subagent

The Event subagent is started in the POSIX shell or in BS2000/OSD.

To start the Event subagent, the privilege NET - ADMINISTRATION is required.

1. Starting in BS2000/OSD:

/START-SNMP-EVENTAGT
VERSION=***STD** / <product-version> , **MONJV**=***NONE** / <filename 1 .. 54 without-gen-vers> , **CPU-LIMIT**=***STD** / <integer 1 .. 32767> / ***NO** , **JOB-CLASS**=***STD** / <name 1 .. 8> **, FILE-NAME**=***STD** / <posix-pathname_1 .. 1023>

2. Starting in the POSIX shell:

```
eventagt  [-f <config-file>]
```

The Event subagent is stopped in BS2000/OSD (irrespective of the environment in which it was started) with:

/STOP-SNMP-EVENTAGT
VERSION=***STD** / <product-version>

or in the POSIX shell with:

```
eventcmd T
```

Description of operands:

VERSION=*STD / <product-version>
Defines the version of the agent to be started or stopped. This parameter is currently not used.

MONJV=*NONE / <filename 1 .. 54 without-gen-vers>
Name of the job variable that is to monitor the agent. If *NONE is specified (default value), there is no monitoring by job variable.

CPU-LIMIT=*STD / <integer 1 .. 32767> / *NO
Maximum CPU runtime in seconds.
If *STD is specified, the generated default value is used.

JOB-CLASS=*STD / <name 1 .. 8>
Job class with which the agent is started.
If *STD is specified, the generated default job class is used.

FILE-NAME=*STD / <posix-pathname_1 .. 1023>
Name of the configuration file.
If *STD is specified, the file */etc/snmp/agt/event.cnf* is used.

4.6 Scheduler subagent

The Scheduler subagent enables modifications to be made to objects periodically or at defined times. It is thus possible, for instance, to issue an SNMP set operation every Monday at 6:00 A.M. or even every last Friday in the month at 10:00 P.M. This type of scheduling can be activated or deactivated by modifying a control object. This makes a preconfigured scheduling possible which can be activated or deactivated by other management functions.

A typical application for the SNMP set operations controlled by the Scheduler subagent is changing the administration status of MIB instances at specific times, e.g.:

– Status of an interface, by setting *ifAdminStatus*

– Monitoring status in the Event subagent, by setting *mteTriggerEnabled*

– Status of application monitoring by the Application Monitor subagent, by setting *appMonLogFState*

Figure 5 illustrates how the Scheduler subagent works.

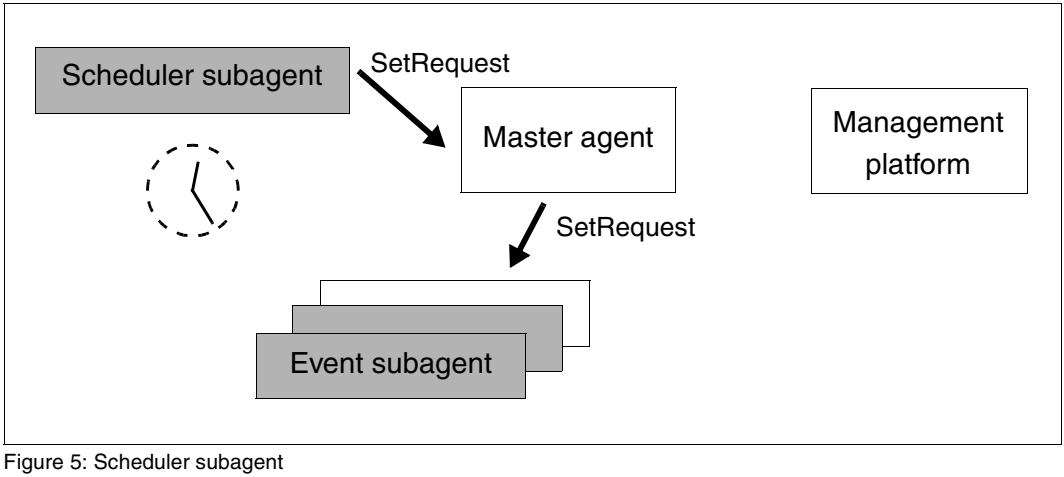

Figure 5: Scheduler subagent

4.6.1 Functionality of the Scheduler subagent

The Scheduler subagent implements the Scheduler MIB (RFC2591), which supports the following types of scheduling:

– Periodic scheduling
– Scheduling on the basis of calendar dates
– One-shot scheduling

Periodic scheduling

Periodic scheduling is based on specific time intervals between two consecutive SNMP set operations initiated by the Scheduler subagent. A time interval is defined by the number of seconds elapsing between two continuous SMNP statements.

Scheduling on the basis of calendar dates (calendar scheduling)

Scheduling on the basis of calendar dates initiates actions on specific week days or days in a month. A calendar time is specified by giving the month, day, weekday, hour and minute.

You can specify a number of values for each date and thus define complex scheduling operations. The scheduling can for instance initiate a specific action every 15 minutes on a given weekday.

Date specifications based on months, days and weekdays can be defined using the following BITS type scheduler MIB objects:

– *schedMonth*
– *schedDay*
– *schedWeekDay*

Setting several bits in one of these MIB objects has the effect of a logical OR. If, for instance, you set the bits monday (1) and friday (5) in *schedWeekDay*, the scheduling initiates the actions specifically on Mondays and Fridays.

The cross-object combination of bit fields from *schedMonth*, *schedDay* and *schedWeekDay* has the effect of a logical AND. If, for instance, you set the bits june (5) and july (6) in *schedMonth* and the bit fields monday (1) and friday (5) in *schedWeekDay*, the scheduling is limited to initiating actions exclusively on Mondays and Fridays in the months of June and July.

When specifying dates, you can implement wildcard functionality if you set all bits to "1".

One-shot scheduling

One-shot scheduling is similar to scheduling on the basis of calendar dates. The only difference is that one-shot scheduling is automatically deactivated once an action has been initiated.

Actions

Actions initiated by scheduling model SNMP set operations on MIB objects whose OID is configured in the object *schedVariable*. The value to be set is specified in the object *schedValue*. Actions defined in this MIB are limited to INTEGER type objects. This restriction does not however reduce the usability of the scheduler MIB. Simple scheduling is thus possible, e.g. scheduling the activation/deactivation of resources with a corresponding status MIB object (e.g. *ifAdminStatus*).

4.6.2 Configuring the Scheduler subagent

You will find a description of how to configure the Scheduler subagent in the Scheduler subagent configuration file sample which is supplied.

```
# Entry type: schedEntry
# Format: schedOwner            String   (Index)
#         schedName             String   (Index)
#         schedDescr            String
#         schedInterval         Integer  (for schedType == periodic(1))
#         schedWeekDay          BITS     (for schedType == calendar(2), oneshot(3))
#         schedMonth            BITS     (for schedType == calendar(2), oneshot(3))
#         schedDay              BITS     (for schedType == calendar(2), oneshot(3))
#         schedHour             BITS     (for schedType == calendar(2), oneshot(3))
#         schedMinute           BITS     (for schedType == calendar(2), oneshot(3))
#         schedContextName      String   (not yet)
#         schedVariable         OID      (object to be set)
#         schedValue            SR_INT32 (value to be set)
#         schedType             SR_INT32 (periodic(1), calendar(2), oneshot(3))
#         schedAdminStatus      SR_INT32 (enabled(1), disabled(2))
#         schedStorageType      SR_INT32 (other(1), volatile(2), nonVolatile(3),
#                                         permanent(4), readOnly(5))

# Example for BITS coding
# Weekday: Monday and Thursday are coded as 48
#   Sunday
#   |   Monday
#   |   |   Tuesday
#   |   |   |   Wednesday
#   |   |   |   |   Thursday
#   |   |   |   |   |   Friday
#   |   |   |   |   |   |   Saturday
#   0   1   0   0   1   0   0
#   |           |   |
#   -------------   -----------
#         |               |
#         4               8                => 48
```

Examples

Scheduler subagent with MIB-II : shut down interface on weekend.

```
ifOperStatus = down every Friday 18:30
ifOperStatus = up every Monday 07:30
```

Scheduler subagent with Event MIB: switch off monitoring on weekend.

```
mteTriggerEnabled.xxx = false every Friday 18:30
mteTriggerEnabled.xxx = true every Monday 07:30
```

Scheduler subagent with Crit-Appl-MIB: switch on application monitoring at specific times:

```
status = (appMonLogFState == start-begin)
```

The first example (ifOperStatus = down every Friday 18:30), requires the following configuration:

```
Entry type: schedEntry
Format: schedOwner            Owner    (Index)
        schedName             IfDown   (Index)
        schedDescr            "     "

        schedInterval         0
        schedWeekDay          04                                  - Friday
        schedMonth            FF:F0                               - all
        schedDay              FF:FF:FF:FE:00:00:00:00             - all
        schedHour             00:00:20                            - 18
        schedMinute           00:00:00:02:00:00:00                - 30

        schedContextName      -
        schedVariable         iso.3.6.1.2.1.2.2.1.7.1             - ifOperStatus
        schedValue            2                                   - down
        schedType             calendar                            - calendar

        schedAdminStatus      enabled
        schedStorageType      readonly
```

4.6.3 Starting / stopping the Scheduler subagent

The Scheduler subagent is started in the POSIX shell or in BS2000/OSD.

To start the Scheduler subagent, the privilege NET- ADMINISTRATION is required.

1. Starting in BS2000/OSD:

/START-SNMP-SCHEDULER
VERSION=***STD** / <product-version> , **MONJV**=***NONE** / <filename 1 .. 54 without-gen-vers> , **CPU-LIMIT**=***STD** / <integer 1 .. 32767> / ***NO** , **JOB-CLASS**=***STD** / <name 1 .. 8> , **FILE-NAME**=***STD** / <posix-pathname_1 .. 1023>

2. Starting in the POSIX shell:

```
schedagt  [-f <config-file>]
```

The Scheduler subagent is stopped in BS2000/OSD (irrespective of the environment in which it was started) with:

/STOP-SNMP-SCHEDULER
VERSION=***STD** / <product-version>

or in the POSIX shell with:

```
schedcmd T
```

Description of operands:

VERSION=*STD / <product-version>
Defines the version of the agent to be started or stopped. This parameter is currently not used.

MONJV=*NONE / <filename 1 .. 54 without-gen-vers>
Name of the job variable that is to monitor the agent. If *NONE is specified (default value), there is no monitoring by job variable.

CPU-LIMIT=*STD / <integer 1 .. 32767> / *NO
Maximum CPU runtime in seconds. If *STD is specified, the generated default value is used.

JOB-CLASS=*STD / <name 1 .. 8>
Job class with which the agent is started. If *STD is specified, the generated default job class is used.

FILE-NAME=*STD / <posix-pathname_1 .. 1023>
Name of the configuration file. If *STD is specified, the file */etc/snmp/agt/scheduler.cnf* is used.

5 Product-specific agents - Functional enhancements in SNMP V6.0

This chapter describes:

– The HIPLEX subagent
– The updated functional scope of the *open*UTM subagent in SNMP V6.0.

The functional scope of the other product-specific agents - agents in the SNMP standard collection and the subagent for performance monitoring with SM2 - has not been changed since SNMP V5.0. Information on these subagents in the manual "SNMP Management V5.0" thus still applies without restriction.

5.1 HIPLEX subagent

The HIPLEX subagent monitors high availability clusters from BS2000/OSD systems and supplies information on the current cluster configuration from the view of a subsystem. The HIPLEX subagent reports any important events and changes in the cluster to the management platform by means of traps, thus enabling the cluster to be monitored efficiently. The HIPLEX subagent uses HIPLEX MSCF and HIPLEX AF interfaces. Both these products are required in versions 1.0 (HIPLEX MSCF) or 3.0 (HIPLEX AF) to make effective use of the HIPLEX subagent.

5.1.1 Functionality of the HIPLEX subagent

The HIPLEX subagent supports the HaCl-MIB (High Availability Cluster MIB). This is a generally defined MIB to monitor and display the configuration in high availability clusters. The following description provides an overview of how the HIPLEX subagent supports the HaCl-MIB, and points out special features resulting from the HIPLEX MSCF and HIPLEX AF concepts.

Objects in the HaCl-MIB

The HaCl-MIB object range is divided into the following areas:

– General data on the subagent and the cluster
– System table
– Application table
– Resource table
– Table displaying application status on connected systems
– Table displaying resource status on connected systems

General data on the subagent and the cluster

The following parameters are displayed for the subagent:

– Subagent version
– Description text
– Subagent type
– Information range
– Notification range
– Name of local system

The HIPLEX subagent supplies information on the cluster configuration from the point of view of the local system.

The information range and notification range can be defined by setting two MIB objects.

The following values can be set:

no: No system-specific information is supplied or no trap with system-specific information is sent.

local: Only information or traps affecting the local system are displayed or sent.

down: The switchover unit is deactivated on all systems on which it is defined.

subset/all: All available information is supplied or traps are sent for all events.

The notification range can never be larger than the information range.

The following general parameters are displayed for the cluster:

– Cluster name
– Description text
– Cluster type
– Cluster version
– Cluster supplier
– Cluster monitoring status

The name of the XCS cluster is displayed as the cluster name if the system belongs to such a cluster.

The cluster monitoring status indicates whether the main procedure of HIPLEX AF has been started.

The main procedure is a prerequisite for participating in a HIPLEX AF cluster. Information on switchover units and objects can only be supplied on this condition.

System table

The system table provides an overview of all systems for which a CCS or an XCS connection exists.

The following are displayed:

– Name of system
– Description text "CCS partner" or "XCS partner"
– Operating status
– Operating system installed
– Operating system version

The basic information in this table corresponds to the MSCF configuration. As BS2000/OSD systems can be linked to UNIX systems as co-systems in a HIPLEX AF cluster, UNIX systems are also displayed. The information on these systems is displayed provided it is available.

Application table

The application table contains information on the switchover units.

The following are displayed:

– Name of switchover unit
– Description text
– Current status
– Current work system

Each switchover unit is defined on a pubset in HIPLEX AF. The indicated name is thus formed from the catalog ID and the actual name of the switchover unit.

The description text comes from the BEGIN-SWITCH-UNIT-DEFINITION statement in the definition of the switchover unit.

The status displayed has the following significance:

online:	The switchover unit has "work" status on a system.
offline:	The switchover unit does not have the "work" status on any system, but does have "stand-by" status on a system.
down:	The switchover unit is deactivated on all systems on which it is defined.
faulted:	The switchover unit was terminated with errors on a system and is deactivated on all other systems.

The work system can only be specified with "online" status.

Resource table

The resource table contains a list of all objects on all switchover units whose switchover procedure is started on the local system.

The following information is supplied for each object:

– Object name
– Description text
– Object status
– Type description

The description text comes from the //ADD... statements (see manual "HIPLEX AF High-availability of Applications in BS2000/OSD") with which the objects are specified in the definition of the switchover unit.

The type description contains the following values:

– BS2000 Application
– UNIX Application
– BS2000 Action
– UNIX Action
– Device
– Virtual Host
– Action on System Crash
– System Action

With HIPLEX AF, only the status of the BS2000/OSD and UNIX applications is monitored. All other objects thus have "not-monitored" status.

Table showing the application status on the connected systems

This table contains entries showing the status for every switchover unit on every system on which the switchover units are defined.

Table showing the resource status on the connected systems

This table is not supported by the HIPLEX subagent. In a HIPLEX AF cluster, resource status and object status are only defined on the work system of a single application.

Notifications

The HIPLEX subagent can report the following changes in the cluster to the management platform by means of a trap:

– Cluster information is available: the MSCF subsystem is started.

– Cluster information is not available: the MSCF subsystem has been stopped.

– The cluster monitoring status has changed: the main procedure of HIPLEX AF was started or stopped or has stopped due to an error.

– The status of a system has changed: the connection to a system was (re-)established or has been lost.

– The status of an application on a system has changed: the status of a switchover unit on a system has changed.

– The status of a resource has changed: a BS2000/OSD or UNIX application was started or stopped or has stopped due to an error.

5.1.2 Configuring the HIPLEX subagent

The HIPLEX subagent does not need to be configured. All important runtime parameters can be set with the start command (see next section).

5.1.3 Starting / stopping the HIPLEX subagent

Starting the HIPLEX subagent in BS2000/OSD:

/START-SNMP-HIPLEX
VERSION=***STD** / <product-version> , **MONJV**=***NONE** / <filename 1 .. 54> , **CPU-LIMIT**=***STD** / <integer 1 .. 32767> / ***NO** , **JOB-CLASS**=***STD** / <name 1 .. 8> , **TIMER-INTERVAL**= 5 / <integer 1 .. 32767> , **HIPLEX-AF-LIBRARY**=***STD** / <full-filename 1 .. 54 without-generation-version> , **INFORMATION-SCOPE**=***ALL** / ***LOCAL** / ***NO** , **NOTIFICATION-SCOPE**=***ALL** / ***LOCAL** / ***NO**

or in the POSIX shell:

```
hiplexagt [-t int] [-l <HIPLEX-AF library>] [-i info-scope] [-n notification-scope]
```

`info-scope` and `notification-scope` can each assume the values "no", "all" or "local".

Stopping the HIPLEX subagent in BS2000/OSD:

/STOP-SNMP-HIPLEX
VERSION=***STD** / <product-version>

or in the POSIX shell:

```
hiplexcmd T
```

Description of operands:

VERSION=*STD / <product-version>
Defines the version of the agent to be started or stopped. This parameter is currently not used.

MONJV=*NONE / <filename 1 .. 54>
Name of the job variable that is to monitor the agent. If *NONE is specified (default value), there is no monitoring by job variable.

CPU-LIMIT=*STD / <integer 1 .. 32767> / *NO
Maximum CPU runtime in seconds. If *STD is specified, the generated default value is used.

JOB-CLASS=*STD / <name 1 .. 8>
Job class with which the agent is started. If *STD is specified, the generated default job is used.

TIMER-INTERVAL=5 / <integer 1 .. 32767>
Interval at which the agent checks for requests from the command program. The default interval is set to five seconds.

HIPLEX-AF-LIBRARY=*STD / <full-filename 1 .. 54 without-gen-vers>
Path name of the SYSLIB of HIPLEX AF. If *STD is specified, the name is determined by IMON.

INFORMATION-SCOPE=*ALL / *LOCAL / *NO
Scope of information supplied by the subagent.
The standard specification *ALL supplies all available information. With *LOCAL, only information affecting the local system is displayed. *NO does not supply any system-specific information.

NOTIFICATION-SCOPE=*ALL / *LOCAL / *NO
Scope of the traps sent by the subagent.
With the standard specification *ALL, traps are sent for all events. With *LOCAL, only traps relating to the local system are sent. With *NO, no traps with system-specific information are sent.

5.2 Subagent for *open*UTM - enhanced functionality in SNMP V5.0B

The functionality of the subagent for *open*UTM has been enhanced as follows:

- Several *open*UTM applications can be monitored.
- Monitored applications can be displayed.
- Traps can be sent for status changes.

Due to these additions to the functional scope, the following areas have been subject to change made since SNMP V5.0:

– Configuring the *open*UTM subagent

– Starting the *open*UTM subagent

– *open*UTM MIB

5.2.1 Configuring the *open*UTM subagent

To monitor several *open*UTM applications, a configuration file is required in which every monitored *open*UTM application must be specified.

Entries in the configuration file are generated with the SDF statement //ADD-APPLICATION-RECORD. With the statement //REMARK, comments can be stored in the configuration file. The file must be terminated with the statement //END.

ADD-APPLICATION-RECORD

The statement //ADD-APPLICATION-RECORD identifies the *open*UTM applications to be monitored.

//ADD-APPLICATION-RECORD
APPLICATION-NAME = <structured-name 1 .. 8>
, **FILEBASE**= ***SAME** / <full-filename_1 .. 54>
, **USER-ID**= **TSOS** / <name_1 .. 8>
, **TRAP-CONDITION** = list-poss (3): ***ABNORMAL-TERMINATED / *NORMAL-TERMINATED / *RUNNING**

APPLICATION-NAME=<structured-name 1 .. 8>
Defines the *open*UTM application which the subagent is to monitor.

FILEBASE= *SAME / <full-filename_1 .. 54>
Basic name of the A-parts of the KDCFILE.
*SAME is the default value.

USER-ID=TSOS / <name 1 .. 8>
ID under which the *open*UTM application is started.
TSOS is the default value.

TRAP-CONDITION= list-poss (3): *ABNORMAL-TERMINATED / *NORMAL-TERMINATED / *RUNNING
Conditions under which a trap is to be generated.
*ABNORMAL-TERMINATED is the default value.

5.2.2 Starting / stopping the *open*UTM subagent

Starting the *open*UTM subagent in BS2000/OSD:

/START-SNMP-UTM
VERSION=***STD** / <product-version> , **MONJV**=***NONE** / <filename 1 .. 54 without-gen-vers> , **CPU-LIMIT**=***STD** / <integer 1 .. 32767> / ***NO** , **JOB-CLASS**=***STD** / <name 1 .. 8> , **FILE-NAME**=***NONE** / <filename 1 .. 54 without-gen-vers> ,**TIMER-INTERVAL** = 5 / <integer 1 .. 32767>

or in the POSIX shell:

```
utmagt [-t <int>] [-f filename]
```

Stopping the *open*UTM subagent in BS2000/OSD:

/STOP-SNMP-UTM
VERSION=***STD** / <product-version>

or in the POSIX shell:

```
utmcmd T
```

Description of operands:

VERSION=*STD / <product-version>
Defines the version of the agent to be started or stopped. This parameter is currently not used.

MONJV=*NONE / <filename 1 .. 54 without-gen-vers>
Name of the job variable that is to monitor the agent. If *NONE is specified (default value), there is no monitoring by job variable.

CPU-LIMIT=*STD / <integer 1 .. 32767> / *NO
Maximum CPU runtime in seconds. If *STD is specified, the generated default value is used.

JOB-CLASS=*STD / <name 1 .. 8>
Job class with which the agent is started. If *STD is specified, the generated default job class is used.

FILE-NAME=*NONE / <filename 1 .. 54 without-gen-vers>
When the agent is started, a configuration file can be specified (see page 86). The configuration file must be saved in the BS2000/OSD file system.
Default: no configuration file is used.

TIMER-INTERVAL=5 / <integer 1 .. 32767>
Interval at which the agent checks for requests from the command program. The default interval is set to five seconds.

5.2.3 Enhancements to the *open*UTM MIB

The *open*UTM MIB has been enhanced as follows:

– Displaying monitored applications
– Global data
– Trap

Displaying monitored applications

In SNMP V5.0, the *open*UTM MIB includes a table of *open*UTM applications. These objects are also supported in BS2000/OSD with SNMP V5.0B.

All applications entered in the configuration file are displayed with their name and status. The objects *utmApplSharedMemKey*, *utmApplSharedMemSegSize* and *utmApplSemId* are no longer supported. *utmApplHomeDir* is still only supported still in Reliant UNIX.

Global data

There is a new object group *utmGlobalData* which contains the following:

- utmGlobalSubagentVersion:
  Version number of the SNMP subagent and type of operating system
- utmGlobalConfFile
  Name of the configuration file. This object is defined as "read-write", so that the configuration file can be changed during live operation.

Trap

The new trap has the following structure:

1. Enterprise:	1.3.6.1.4.1.231.2.19.20.2
2. Trap number:	dependent on the state transition of the *open*UTM application: 1: stopped abnormally 2: stopped normally 3: started
3. Variable binding:	utmApplication 1.3.6.1.4.1.231.2.19.20.1.1 (OCTET STRING) name of associated *open*UTM application

6 SNMP management

There are three options available for SNMP management for BS2000/OSD:

- **Web access** to the BS2000/OSD management system enables you to be kept comprehensively and constantly up to date on the status of your BS2000/OSD systems wherever you are working, whether centrally at a computer center or locally, at home or when travelling. You can also take steps to control it. All access, both over SNMP and the web, is protected by stringent security mechanisms.
- For a number of subagents, there are **management applications** tailored specifically to the functionality of the corresponding subagent. You can thus display the values returned by the subagent, for instance, in table format or even in an appropriate graphical format. The management applications can either run as standalone applications or be integrated into a management platform.
- **Integration in the Unicenter management platform** provides you with the full range of the Unicenter network and system management functions on BS2000/OSD systems. The BS2000/OSD SNMP management system can be integrated in a few minutes in Unicenter 3.0 or one of the preceding versions. In a prompted dialog, you can optionally change the preset parameters of the SMBS2 installation package to your own personal preferences or accept the standard settings of the package. No time-consuming "manual" installation and configuration requiring detailed knowledge is needed.

The individual options for access to SNMP management for BS2000/OSD are described in detail below.

6.1 Web access to the BS2000/OSD management

Besides processing SNMP requests, the SNMP master agent also enables management information to be accessed via the World Wide Web (WWW). The information provided by the subagent can thus be queried and modified both over an SNMP management platform and a web browser.

6.1.1 Two different types of request

The master agent polls the network for two different types of request:

– At the SNMP port (normally UDP 161), the master agent expects SNMP SetRequests and SNMP GetRequests.

  The master agent sends SNMP GetResponse messages in response to the SNMP Requests.

– At the web-based management port (normally TCP 280), the master agent expects HTTP connection requests.

  The master agent returns an HTML page to the browser in response to an HTTP message. This HTML page can be a predefined, user-specific web page (custom page) or an automatically generated web page (subtree page).
  The section of the master agent which is responsible for processing HTTP messages is called the HTTP engine.

HTTP requests are processed in the same way as SNMP requests. When an SNMP or HTTP request has been evaluated, the master agent stores the relevant components of the request in an internal queue and obtains the information from the subagent in the usual way. As soon as the master agent has received the information from the subagent, it generates, depending on the type of request, an SNMP GetResponse message or an HTML page and returns this with the requested information to the sender of the original message. For the subagent, there is no difference between SNMP requests or requests from the web.

The relationship between the SNMP and web interfaces of the BS2000/OSD agent is illustrated in figure 6.

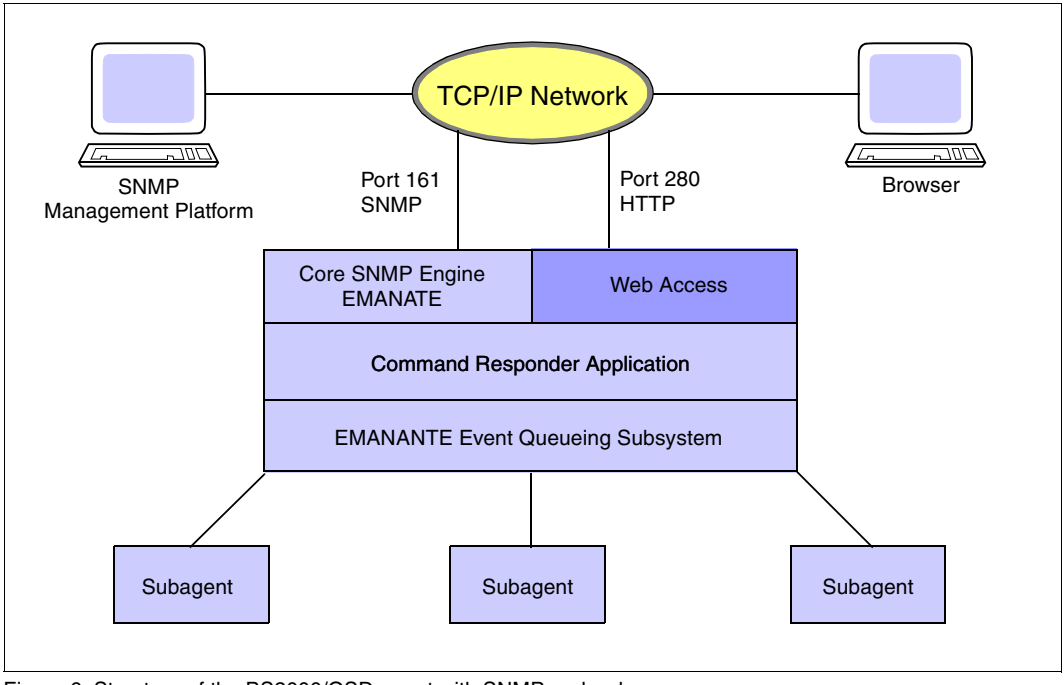

Figure 6: Structure of the BS2000/OSD agent with SNMP and web access

6.1.2 Establishing a connection to the BS2000/OSD web agent

To connect to the BS2000/OSD web agent (DR-Web Entity) enter the network address and port number at the browser prompt as follows:

`http://<networkaddress>:<portnumber>`

For example: http://D016ZE07:280 is the address of the web agent running on system D016ZE07.

Entering the Username and Password

When a web browser has successfully connected, the browser will prompt the user for a username and a password:

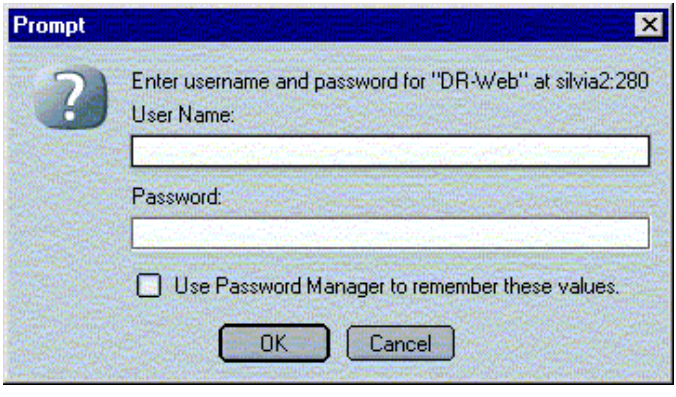

Figure 7: Entering the username and password

The username and password must be configured at the agent. The configuration is set on delivery to accept *gast* as the username and an empty string as the password (i.e. you do not enter anything as password).

- Enter "gast" in the user ID field and leave the password field empty.
- Click *OK*.

When the logon has been accepted, the web agent displays a welcome screen on the browser.

BS2000/OSD web agent welcome screen

Figure 8 shows the standard welcome screen used, with hyperlinks to the subtree and custom page branches and the trap reception.

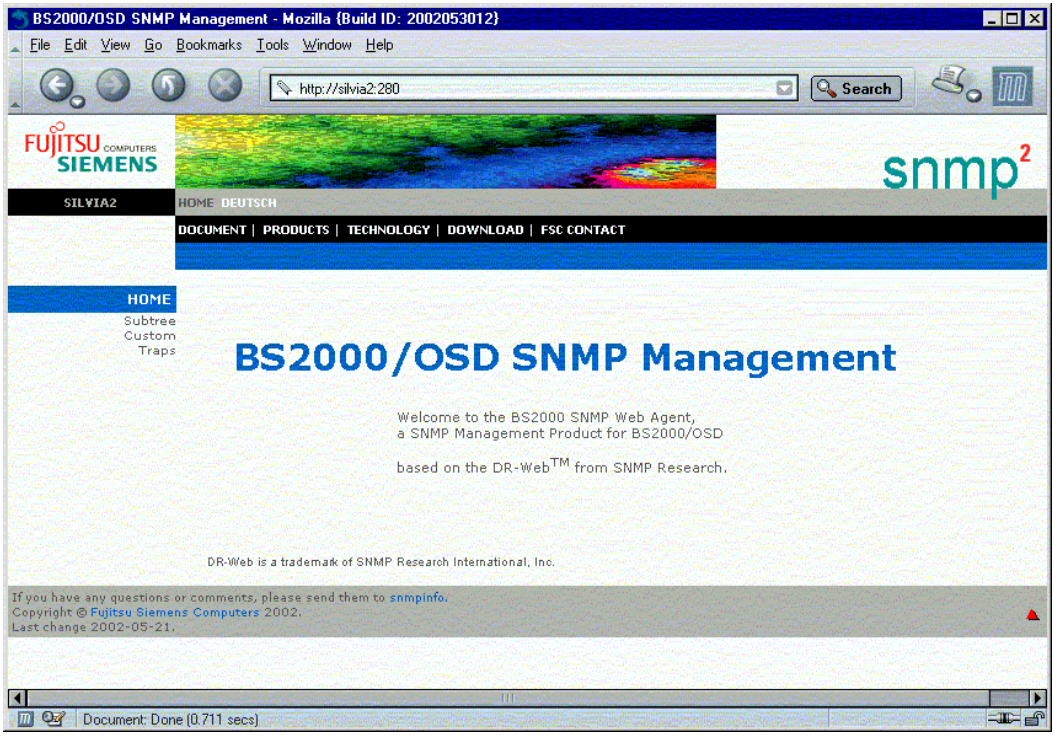

Figure 8: Web agent welcome screen

6.1.3 Subtree functionality, custom pages and trap display

You can select the following from three links in the BS2000/OSD web agent welcome screen:

- *Subtree* takes you to a display of the standard pages which the master agent generates from the MIB definitions (subtree functionality).
- *Custom* takes you to the user-specific pages defined by the HTML subagent and the HTML MIB (custom page functionality).
- *Traps* opens a page to display the traps. A plug-in is downloaded to do this, with a java applet which enables the trap to be received.

Subtree functionality

The subtree page contains hyperlinks to subtree URLs with which you can display management information accessible via the web agent. This page is pre-configured to ensure quick access to all MIBs supported by the SNMP management system in BS2000/OSD. For further information, see the manual "SNMP Management V5.0".

Custom page functionality

When you use the HTML subagent on your system, a single click on the *Custom* hyperlink of the DR web welcome screen will make the custom page functionality available to you. With this, you can use pre-configured web pages or create your own web pages (custom pages) which contain, besides all components such as text, graphics etc., additional macros for accessing individual MIB objects. Furthermore, you can group information according to individual criteria. For further information on how to configure custom pages as well as on the HTML subagent and the HTML MIB, see the manual "SNMP Management V5.0".

Trap display in the web browser

With the DR web interface, you can call up a web page which displays incoming traps in a table. This table is implemented as a Java applet. Due to the necessary access restrictions for the security of Java applets, it is only possible to receive traps that originate from the system from which the web page was loaded. For further information on this, see the manual "SNMP management V5.0".

6.2 Management applications

This section describes the management applications:

– BCAM Manager (BMBS2)
– Console and Application Monitor (CMBS2)
– Performance Monitor (PMBS2)
– Cluster Monitor (CluMon)

It also describes how to install the Tcl-Set V6.0 interpreter, which is a prerequisite for using the management applications above.

6.2.1 Installing the Tcl-Set interpreter

The Tcl-Set interpreter (tclset or SMAWtcl) is installed as follows:

► In Windows, use the setup call:

```
tclset6A00.exe
```

► In Solaris, use the call:

```
pkgadd -d <pathname>/SMAWtcl-6.0.stream
```

► In Linux, use the call:

```
rpm -i snmptcl-6.0.rpm
```

6.2.2 BCAM Manager

The BCAM Manager (BMBS2) is an SNMP management application optimized to work with the following MIBs:

- MIB-I I
- private BCAM MIB

Installing the BCAM Manager

The BCAM Manager (BMBS2 or SMAWbmbs2) is installed as follows:

► In Windows, use the setup call:

```
bmbs250A00.exe
```

► In Solaris, use the call:

```
pkgadd -d <pathname>/SMAWbmbs2-5.0.stream
```

► In Linux, use the call:

```
rpm -i snmp-5.0.rpm
```

A Tcl-Set ≥ V 5.0 interpreter is a prerequisite for installing the BCAM Manager.

Functions of the BCAM Manager

The BCAM Manager offers the following functions:

– Monitoring several systems
– Supporting all MIBs
– Defining short names for every MIB variable
– Defining default variables for every MIB group or MIB table
– Searching for table instances with any criteria
– Searching for table instances using criteria from other tables for the BCAM MIB
– Defining and storing search queries
– Defining graphics functions with bar charts and line charts
– Automatically searching for new table instances

For more detailed information on the BCAM Manager, see the manual "SNMP management for *open*Net Server and *inter*Net Services".

6.2.3 Console and Application Monitor

The Console and Application Monitor BS2000 (CMBS2) is an SNMP management application used to monitor the BS2000/OSD console and applications.

The Console and Application Monitor interoperates with two special subagents:

- Console Monitor Subagent in BS2000/OSD
- Application Monitor Subagent in BS2000/OSD or AppMon-SX subagent on Reliant UNIX

The SNMP protocol is used for communication between the CMBS2 application and the relevant subagent.

For more detailed information on how to operate the Console and Application Monitor, see the Online help for this SNMP management application.

Figure 9 shows the Console and Application Monitor trap window. The trap window is also the main window of CMBS2. Windows for further actions can be opened from it.

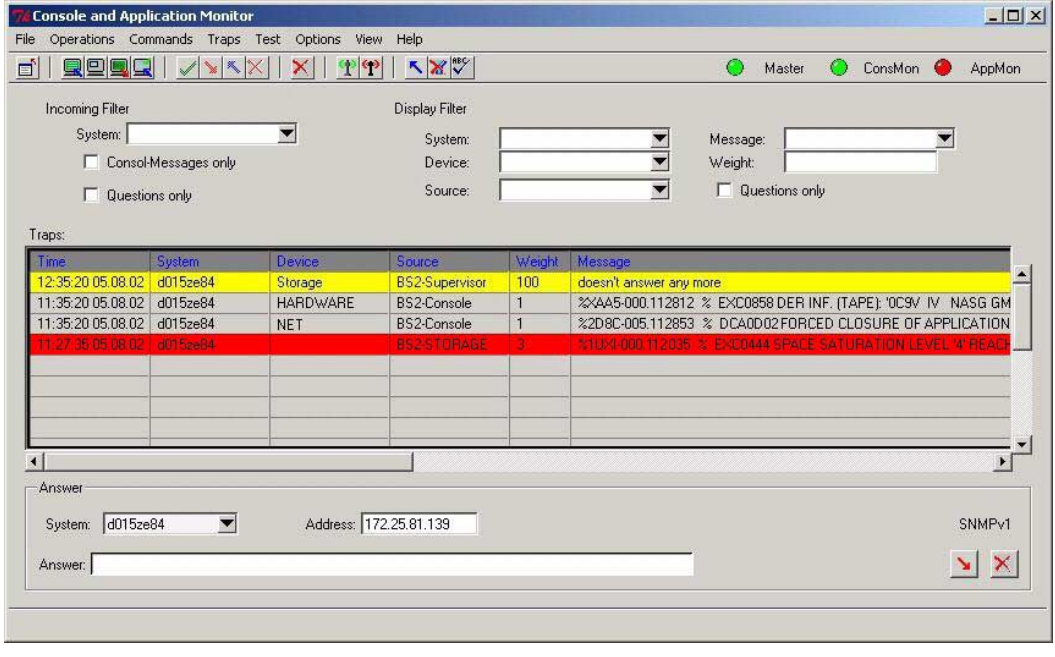

Figure 9: Main window of the Console and Application Monitor (CMBS2)

Requirements

- The following are prerequisites for operating the Console and Application Monitor:
  - On the BS2000/OSD side: a started SNMP master agent and the Console Monitor subagent or the Application Monitor subagent
  - On the Reliant UNIX side: AppMon-SX subagent
- The trap acknowledgement mechanism (see page 101) additionally requires the started Supervisor subagent in BS2000/OSD.

Installing the Console and Application Monitor

The Console and Application Monitor (CMBS2 or SMAWcmbs2) is installed as follows:

► In Windows, use the setup call:

```
cmbs26A00.exe
```

► In Solaris, use the call:

```
pkgadd -d <pathname>/SMAWcmbs2-6.0.stream
```

► In Linux, use the call:

```
rpm -i snmpcmbs2-6.0.rpm
```

Monitoring the console

The Console Monitor subagent records the console messages and sends them individually as traps to the Console and Application Monitor. The basic functionality of the Console and Application Monitor is to display these messages. In addition to this display function, the Console and Application Monitor offers the following options:

– Filtering messages
– Highlighting and logging specific messages
– Conveniently replying to console queries

Console commands

You can use the Console and Application Monitor and the Console Monitor subagent in BS2000/OSD to execute console commands. If you are using the AppMon-SX subagent on an agent system, you can issue UNIX shell commands there. It is also possible to predefine commands.

Monitoring applications

The Application Monitor subagent in BS2000/OSD monitors user and BCAM applications (by means of Monitor Job variables), DCAM applications, subsystems, job variables and logging files.

You can use the AppMon-SX subagent to monitor log files in Reliant UNIX. The Application Monitor subagent can report any changes to the Console and Application Monitor with a trap.

The following options are also offered here:

– Filtering messages
– Highlighting and logging specific messages
– Conveniently replying to console queries

Independently of this, you can poll the status of the monitored subsystems and user applications at the Application Monitor subagent from the management platform.

Automatic reactions

You can define automatic reactions to respond to the problem messages which the Application Monitor subagent reports in a trap. Specify the criteria with which a reaction is to be triggered by a trap and define the reaction. The following are some options you may choose from:

– Send a BS2000 command
– Execute a local command
– Send mail
– Write an entry in a log file
– Execute any Tcl-Script
– Send a trap

Trap acknowledgement

Using an agent to asynchronously report problems via traps is extremely efficient, as it keeps the network load to a minimum. One problem, however, is that the information is lost if no management platform is active when the trap is sent or if communication to the management platform is faulty.

To prevent trap information from being lost in these circumstances, you can declare a trap "to be acknowledged". In this case, the subagent includes internal information with the trap. The Console and Application Monitor can recognize this information and subsequently automatically sends a Set-Request to the agent. If the subagent receives the request, then it considers the trap to be acknowledged. The subagent stores traps which have not been acknowledged until an acknowledgement is received.

6.2.4 Performance Monitor

The Performance Monitor BS2000 (PMBS2) is an SNMP management application used for monitoring performance in BS2000/OSD. The PMBS2 application is tailored specifically to the functionality of the Performance Monitor subagent in BS2000/OSD. The SNMP protocol is used for communication between the PMBS2 application and the Performance Monitor subagent in BS2000/OSD. For a short overview of the SNMP protocol and the SNMP management concept, see the section “SNMP management architecture” on page 8.

For more detailed information on how to operate the Performance Monitor, see the online help on this SNMP management application.

Requirements

Prerequisites on the BS2000/OSD side for operating the Performance Monitor application are: a started SNMP master agent, the Performance Monitor subagent and a started SM2 subsystem.

Installing the Performance Monitor

The Performance Monitor (PMBS2 or SMAWpmbs2) is installed as follows:

- In Windows, use the setup call:

  ```
  pmbs26A00.exe
  ```

- In Solaris, use the call:

  ```
  pkgadd -d <pathname>/SMAWpmbs2-6.0.stream
  ```

- In Linux, use the call:

  ```
  rpm -i snmppmbs2-6.0.rpm
  ```

Tabular and graphical display

The basic functionality of the Performance Monitor application consists of the tabular and especially the graphical display of the SM2 measured values provided by the Performance Monitor subagent on the BS2000/OSD side. The current measured values of one or more BS2000/OSD systems are displayed in forms and tables or graphically as bar charts and line graphs. You can determine any interval for the automatic updating of the displays.

The objects and values to be displayed are defined precisely in the SNMP performance MIB. Both the PMBS2 application on the management platform and the Performance Monitor subagent in BS2000/OSD have access to this MIB. The Performance Monitor only accesses the MIB objects presently to read them. For a detailed list of the MIB objects of the Performance MIB, see the manual "SNMP Management V5.0".

Figure 10 shows a graphical display of the CPU utilization by the Performance Monitor.

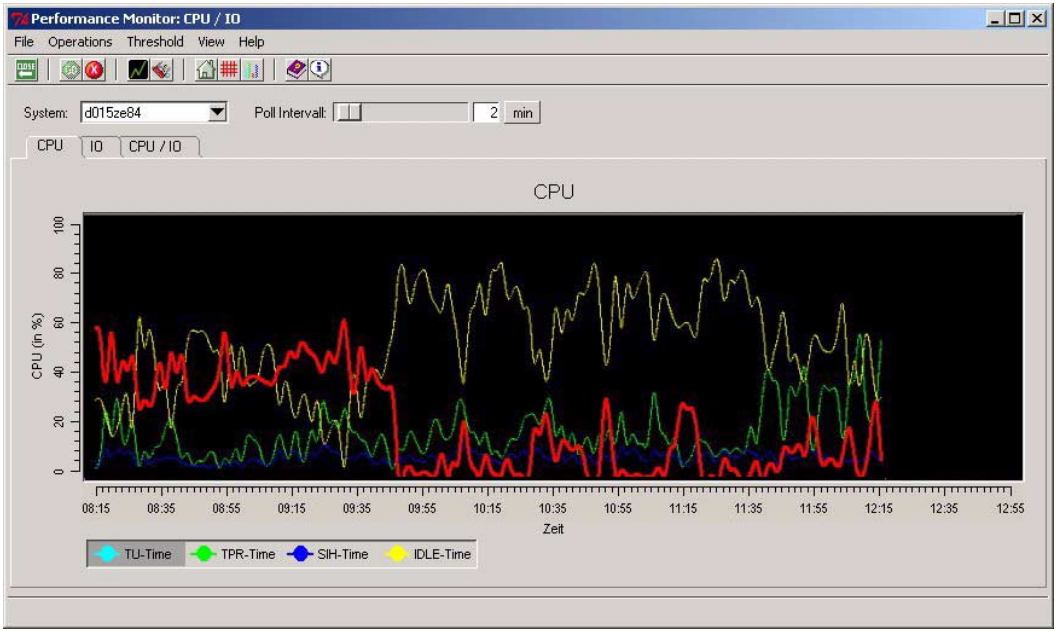

Figure 10: Display of the CPU utilization by the Performance Monitor (PMBS2)

Threshold values and reactions

In addition to the basic functionalities for displaying current SM2 measured values, the Performance Monitor enables critical threshold values to be defined and determines appropriate reactions if values exceed or fall below these threshold values. You can choose freely from several types of reaction.

Possible reactions include:

– Entries on a logfile
– Audible signals
– Execution of specific commands
– Generation of SNMP traps

Traps can be sent to further SNMP management applications as asynchronous event messages, so that the Performance Monitor application is incorporated seamlessly to the SNMP management system.

6.2.5 Cluster Monitor

The Cluster Monitor (CluMon) is an SNMP management application used to monitor high availability clusters in BS2000/OSD and on Reliant UNIX. It interoperates with the HIPLEX subagent in BS2000/OSD. The SNMP protocol is used for communication between the CluMon application and the HIPLEX subagent in BS2000/OSD.

For more detailed information on how to operate the Cluster Monitor, see the online help section on this SNMP management application.

Requirements

Prerequisites for operating the Cluster Monitor are a started SNMP master agent, the HIPLEX subagent in BS2000/OSD or the corresponding subagent on Reliant UNIX.

Installing the Cluster Monitor

The Cluster Monitor (CluMon or SMAWclumn) is installed as follows:

- In Windows, use the setup call:

  ```
  clumon1A00.exe
  ```

- In Solaris, use the call:

  ```
  pkgadd -d <pathname>/SMAWclumn-1.0.stream
  ```

- In Linux, use the call:

  ```
  rpm -i snmpclumn-1.0.rpm
  ```

Monitoring the cluster

HIPLEX subagents are located on the cluster systems. They supply information on the current cluster configuration from the viewpoint of one of these subsystems. The Cluster Monitor application evaluates the information from all HIPLEX subagents operating in the cluster and presents an overall view of the status of the cluster (see figure 11). It can still do this even if a system in the cluster fails. The graphical display enables the configuration and any problems occurring in the cluster network to be recorded quickly.

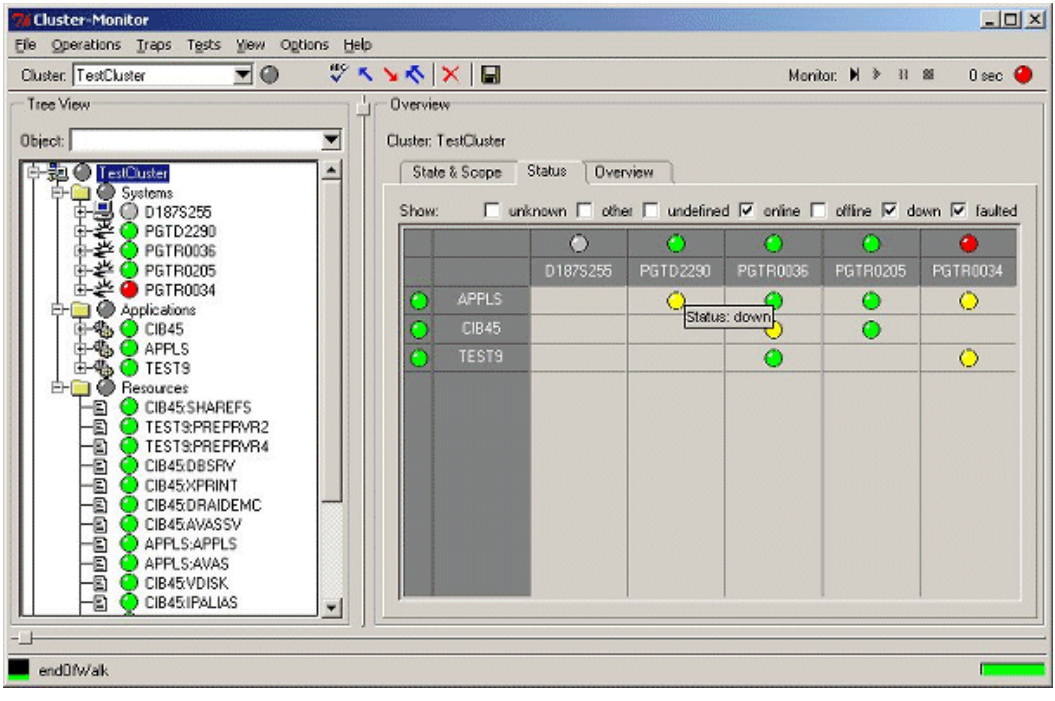

Figure 11: Overall view of the status of the cluster

The Cluster Monitor makes various views possible:

– Cluster view
– Systems view
– Applications view
– Resources view

Cluster view

The cluster view provides:

– a tabular listing of the information made available by the agents of the individual systems
– a matrix display of the status of the applications on the individual systems
– a compressed overview of the status of the systems and applications participating on the cluster

Systems view

The systems view provides:

– a display of the cluster configuration as a network of systems.
– a tabular overview of the data of systems participating on the cluster, e.g. name, status, operating system etc.

Applications and resources view

Similarly to the systems view, this provides an overview of the applications and resources participating on the network with the relevant data and status.

Monitoring

To automatically update the display, you can activate polling. The poll frequency is freely selectable.

Traps

Significant events and status changes in the cluster are reported to the management platform by the HIPLEX subagents as traps. By asynchronously reporting problems via traps, the agent is extremely efficient, as this reduces network load to a minimum. The traps are displayed in the Cluster Monitor. Furthermore, the Cluster Monitor uses the trap information to update the status display of the cluster.

6.3 Integration in Unicenter

The integration packages SMBS2 (for Windows NT and Windows2000) and SMAWsmbs2 (for Solaris) contain supplementary sections for incorporating the system management system for BS2000/OSD into the management platform Unicenter from Computer Associates.

6.3.1 Requirements for integration

To integrate BS2000/OSD systems into Unicenter, the following are required:

- On the management side:
  - Unicenter 3.0 (WindowsNT V4.0, Windows2000) or

    Unicenter TNG 2.2, 2.4 or 2.4.1 (WindowsNT V4.0, Windows2000 or Solaris as of V2.6) or
  - Integration package SMBS2 V5.0B for the SNMP management system for BS2000/OSD.
- In BS2000 as of OSD V2.0:
  - SNMP basic agent BS2000 SBA-BS2 V6.0

  and optionally:
  - SNMP standard collection BS2000 SSC-BS2 V6.0
  - SNMP subagent SM2 SSA-SM2-BS2 V5.0B
  - SNMP subagent for *open*UTM (BS2000) SSA-OUTM-BS2 V5.0B

6.3.2 Installing the integration packages

SMBS2 and SMAWsmbs2 are installed as follows:

- SMBS2 is installed in Windows NT or Windows 2000 with the setup call:

  ```
  smbs2_setup.exe
  ```

- SMAWsmbs2 is installed in Solaris with the call:

  ```
  pkgadd -d <pathname>/SMBS2-SO.stream.5.0B00
  ```

The additionally required data is thus automatically transferred to the Unicenter installation directory and necessary adjustments are made to the configuration.

At the start of the installation, specify important installation parameters in a prompted dialog. Appropriate default settings can be adopted for all parameters.

6.3.3 Integration into Unicenter components

SMBS2 and SMAWsmbs2 contain several elements which complement and thereby extend the configuration of the "World View", "Enterprise Management", "Agent Technology" and Unicenter Explorer.

World View

A new host class with the name "SiemensBS2000" and a number of agent classes are introduced for the repository. These new classes have some enhancements in comparison to their superclasses "Host" and "Agent". They are linked with special icons for 2D and 3D network maps and the Unicenter Explorer.

The standard popup menu for host objects has several new functions for BS2000/OSD objects. BS2000/OSD objects can be added to the repository and the network map both manually and using the automatic discovery facility. BS2000/OSD MIBs are provided for the Object View.

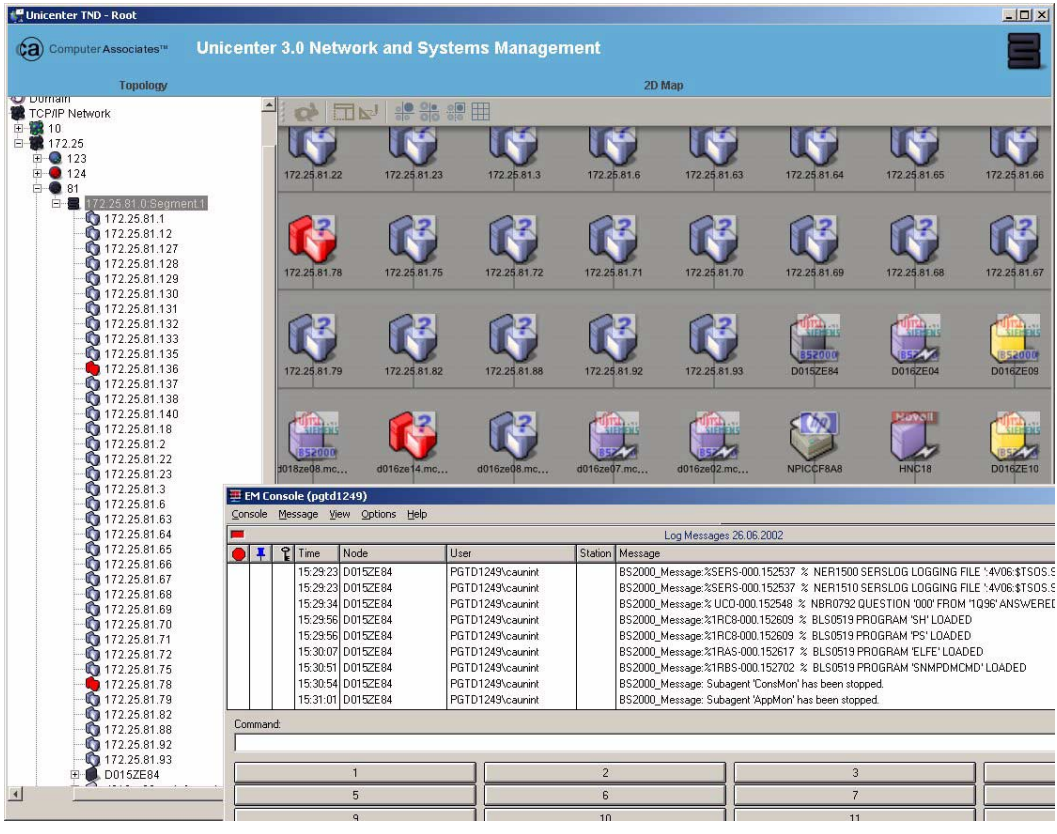

Figure 12: BS2000/OSD systems in the Unicenter Explorer 2D network map and Event console with BS2000 messages

Enterprise management

To monitor BS2000/OSD systems at the event console, SMBS2 contains message formats for all traps of the BS2000/OSD SNMP agent. In particular, all BS2000/OSD console messages which the console monitor subagent sends on as traps can be displayed on the Unicenter event console. The BS2000/OSD messages are provided with special attributes so that you can easily link the messages to reactions.

Agent technology (Agent Technology)

The host class "Siemens-BS2000" is defined in a similar way to "World View" and is assigned ten agent classes with DSM policies. Two of these are agent classes contained in the Unicenter scope of supply and eight are newly created classes. Each of the new agent classes has an agent view and a node view.

The node view in BS2000/OSD can include the following objects:

– Ping
– Mib2
– Application Monitor
– AVAS
– HSMS
– Omnis
– RDBMS
– Storage
– Supervisor
– UTM

Ping

This policy implements a general monitoring of the system by means of regular pings. If the response behavior of the system changes, the object status is changed.

Mib2

There are subobjects for all interfaces in the MIB-II interface table which display the status of the interfaces.

Application Monitor

Four subobjects are generated for subsystems and for BCAM, user and DCAM applications. These objects have subobjects for the individually monitored subsystems, BCAM, user and DCAM applications. These subobjects display the status of the subsystems being monitored and of the BCAM, user and DCAM applications.

AVAS

There is an object which displays the overall status of AVAS.

HSMS

There is an object which displays the availability of the HSMS subagent.

OMNIS

Subobjects are generated for all OMNIS systems being monitored, and there is a subobject for each trap class for every OMNIS system. The general status of all OMNIS systems being monitored is displayed. The subobjects to the trap classes change their status as soon as a trap of this class is received.

RDBMS

There are subobjects for all database servers. Database subobjects are generated for each database server. The status of these database subobjects indicates the availability of the database servers for the databases.

Storage

Two subobjects are generated for pubsets and private disks. Each of these subobjects contains subobjects either for the pubsets or the private disks to be monitored:

– The status of the pubset objects displays the saturation level of the pubsets which has been reached.

– The status of the private disk objects displays the availability of the private disks.

Supervisor

There are subobjects for all subagents in the subagent table, displaying the status of the subagents.

openUTM

There are subobjects for all *open*UTM applications being monitored, displaying the status of the *open*UTM application.

On the next page you will find an example of a NodeView display in Unicenter.

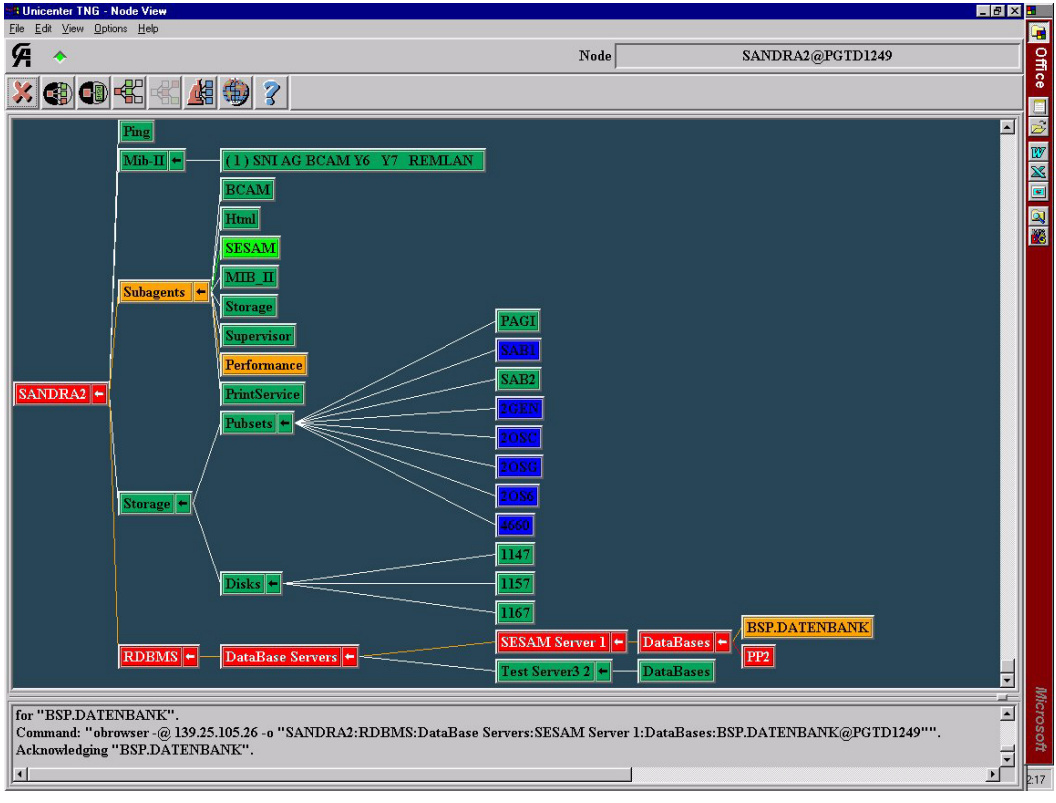

Figure 13: NodeView display in Unicenter

7 Security considerations when using SNMP

This chapter provides notes and recommendations on how to implement secure use of the SNMP-based BS2000/OSD management system with security in mind. It is not however intended as a set of instructions on security analysis or on how to draw up security guidelines. Both of these important subjects go beyond the context of the present manual.

For functional details of how to set the configuration parameters of the SNMP agent in BS2000/OSD with security in mind, see the manual "SNMP Management V5.0", section "3.3.1 Security configuration".

Details on the corresponding settings for the management platform are described in the section “Integration in Unicenter” (see page 108) of the present manual.

7.1 Security as a process

Security is not just a product or a solution but also a process, comparable with processes involved in quality management. Only permanent security management can ensure durable security.

The process "Protection for objects worth protecting" pursues the following objectives against inadvertent or arbitrary external and internal threats:

– confidentiality
– integrity
– availability
– responsible use

The process relies heavily on the guidelines in the security manual.

A comprehensive security manual contains the following process steps:

- preventive measures
- recognition
- reaction

Preventive measures

Preventive measures include configuring security parameters correctly in accordance with the security rules. Preventive measures alone, however, do not guarantee security as security safeguards can never be perfect.

Recognition

Recognition involves logging the line operation and regularly checking logs of security-related incidents, such as repeated violation of access protection (infiltration attempts).

Reaction

Reaction involves checking:

– the adequacy and efficiency of the security rules
– whether the preventive measures need to be revaluated
– whether the mechanisms used in the recognition step need to be intensified

7.2 Recommendations for general network and system security

With the SNMP protocol you communicate across the internet. However, this exposes the participating systems to the potential attacks and risks associated with the public internet.

Recommendation

Put the BS2000/OSD systems and management platform which are to be managed in a subnetwork over which you have exclusive control, for instance, behind a firewall. In this way, you can control access to your systems more easily and more centrally and counter any possible attacks.

The security functions implemented in the SNMP agent are based on the fact that its configuration and program files are only accessible to privileged users, usually an administrator. The installation creates these files with the correct privileges.

Recommendation

Check the configuration and program files of the SNMP agent in BS2000/OSD at regular intervals to ensure they are only accessible to privileged users.

7.3 Recommendations for using the SNMP service safely

SNMP is insecure when used naively. The standard configuration effective with a minimum configuration after installation is a compromise between safety and comprehensive interoperability in the SNMP network, which tends toward interoperability requirements.

Avoid live operation with a minimum configuration of the BS2000/OSD system. If you change the configuration on the BS2000/OSD system being managed, please adapt the corresponding settings on the management platform.

Recommendation

Change the minimum configuration of the SNMP agent and the corresponding settings on the management platform in accordance with the guidelines of your security manual.

You should be particularly careful from a security point of view with the following configuration parameters:

- Community strings for receiving SNMP requests
- Community strings and access control of MIB objects
- Community strings and sender addresses
- Recipient's addresses for SNMP traps
- Activating the sending of Authentication Failure Traps

7.3.1 Community strings for receiving SNMP requests

In SNMP terminology, a "community" denotes a group comprising one or more management platforms and several SNMP agents handled by these platforms.

Every community is identified by a community string. The community string is a non-encrypted component of every SNMP request and identifies the sender of the request as a member of the community concerned. Authorization for a read or write request which a management platform sends to an SNMP agent is controlled with this community string.

The community string makes a simple authentication mechanism available in SNMP. Management platforms and SNMP agents may only communicate with one another if they belong to the same community: the SNMP agent will only accept SNMP requests from management platforms whose community strings are known to it, i.e. preconfigured.

Since the community string is sent in non-encrypted form with the SNMP message, it is always at a risk of being used without authorization. This can be problematic for using SNMP with security in mind. On the other hand, most communities use the preset community string "public" in any case.

Recommendation

Select suitable communities corresponding to the organization of your systems and operations and assign suitable community strings to them. Change the community string in accordance with the guidelines of your manual: in a similar way, for instance, as for passwords. Note that you must modify the community string in all participating systems in the community.

Recommendation

If the environment of your SNMP agents and management platform(s) allows you to do so, use the user-specific authentication available with the SNMPv3 protocol.

7.3.2 Community strings and controlling access to MIB objects

Community strings can be given the access rights "read-only", "read-write" etc. In accordance with these access rights, SNMP requests containing this community string may read objects defined as "read-only" and read and change objects defined as "read-write". If there an no further precautions, *all* accessible objects can be read or *all* changeable objects can be read and/or changed.

Recommendation

Use the option of assigning selective read or write privileges to specific community strings, for:

- MIB branches
- objects
- object instances (in tables)

Recommendation

If the environment of your SNMP agents and management platform(s) allows you to do so, use the user-specific authorization available in the SNMPv3 protocol.

7.3.3 Community strings and sender addresses

You can explicitly preconfigure the IP addresses of the authorized management platforms. In this way, you force the SNMP agent to only accept SNMP requests from these systems.

The management platforms must have fixed IP addresses to do this. It is not possible to dynamically assign them with the Dynamic Host Configuration Protocol (DHCP).

Recommendation

Check the sender addresses of the management platforms by configuring their IP addresses in the SNMP agent.

7.3.4 Recipient's addresses for SNMP traps

You can explicitly predefine the IP addresses of the authorized management platforms to which an SNMP agent is to send SNMP traps. The management platforms must have constant IP addresses to do this. It is not possible to dynamically assign them with the Dynamic Host Configuration Protocol (DHCP).

Recommendation

Predefine the receiver's addresses of the management platforms to receive SNMP traps. To do this, configure the IP addresses of these management platforms in the SNMP agent.

7.3.5 Community string for SNMP traps

You can configure the community string which an SNMP agent sends to the management platform as part of an SNMP trap. The management platform will then only accept SNMP traps with this community string.

Recommendation

Select suitable communities corresponding to the organization of your systems and operations. Change the community string in accordance with the guidelines of your manual: in a similar way, for instance, as for passwords. Note that you must change the community string in all participating systems in the community.

7.3.6 Activating Authentication Failure Traps

The SNMP agent checks every SNMP request in accordance with the configured security parameters and options (see preceding sections). If the SNMP request passes the checks, the agent processes it. Otherwise, the agent discards the SNMP request and sends an Authentication Failure Trap.

By default, the sending of Authentication Failure Traps is deactivated. When it is activated, the SNMP agent sends this trap to all configured receivers.

Recommendation

Configure the parameter `snmpEnableAuthenTraps`, so that the sending of Authentication Failure Traps is activated.

Configure a suitable receiver for traps like this.

Check at regular intervals whether these traps have occurred and analyze all events of this type.

Related publications

Ordering manuals

SNMP Management V5.0
SNMP Management for BS2000/OSD
User Guide

Target group
The manual addresses network administrators/operators and system administrators who wish to integrate a BS2000 system in SNMP-based management or operate such a system.
Contents
This manual describes how SBA-BS2, SSC-BS2, SSA-SM2-BS2 and SSA-OUTM-BS2 are embedded in BS2000/OSD, the installation and configuration procedures required to enable operation, and actual system operation. The Agents and their MIBs which are required for monitoring are dealt with in detail. Installation and configuration of the relevant management applications on the Unicenter TNG, TransView SNMP and HP OpenView management platforms are also described.
Further central topics of the manual are access to management information via the World Wide Web, and the Trap Server for Solaris and Reliant UNIX.

***open*Net Server V2.0** (BS2000/OSD)
BCAM V16.0A Volume 1
User Guide

Target group
The manual is intended for network planners, generators and administrators who define BS2000 systems.
Contents
BCAM Volume 1 describes BCAM itself, how it is embedded in TRANSDATA and TCP/IP and ISO networks, plus generation and administrative activities.
Generation examples illustrate the description.
Additionally BCAM tools for generation and diagnosis are described.

***open*Net Server V2.0** (BS2000/OSD)
BCAM V16.0A Volume 2
Reference Manual

Target group
The manual is intended for network operators, generators and administrators who define BS2000 systems.
Contents
BCAM Volume 2 is based on Volume 1 and describes in detail the BCAM commands required for generation and operation.
The KOGS macros required for static generation are introduced and the BCAM messages are listed.

***open*Net Server V2.0, *inter*Net Services V2.0** (BS2000/OSD)
SNMP Management for *open*Net Server and *inter*Net Services
User Guide

Target group
This manual is intended for network and system administrators wishing to use SNMP-based network and system management.
Contents
The manuals contains detailed descriptions of the MIBs delivered with *open*Net Server, the FTP-MIB delivered with *inter*Net Services, and the installation and operation of the subagents. Operation of the BCAM Manager is described in detail in a separate chapter.

***inter*Net Services V2.0** (BS2000/OSD)
Administrator Guide

Target group
This manual is intended for network planners, generators and administrators who wish to use Internet Services in BS2000/OSD.
Contents
The manual describes the functionality of the Internet Services
BOOTP/DHCP, TFTP, DNS, FTP, LDAP and NTP in BS2000/OSD. It also covers the installation, administration, operation, and logging and diagnostic options of the individual components, as well as the FTP exit and the TELNET exits.

***inter*Net Services V2.0** (BS2000/OSD)
User Guide

Target group
This manual is intended for users and network planners, generators and administrators who wish to use Internet Services in conjunction with BS2000/OSD.
Contents
The manual introduces the components of *inter*Net Services. It contains a detailed description of FTP, the FTAC interface for FTP and TELNET. Network administrators require this manual as a supplement to the Administrator Guide.

HIPLEX AF (BS2000/OSD)
High-Availability of Applications in BS2000/OSD
Product Manual

Target group
This manual addresses system administrators and operators of BS2000/OSD.
Contents
The manual provides information on the requirements for switching applications and on operating HIPLEX AF. As it provides concrete know-how on how applications can be made switchable (organisation, generation, adapting procedures), it can also be used in preparing applications for manual switching with maximum reliability.

DSSM/SSCM
Subsystem Management in BS2000/OSD
User Guide

Target group
This manual addresses systems support staff and software consultants of BS2000/OSD.
Contents
The following are described: BS2000/OSD subsystem concept, dynamic subsystem management (DSSM), subsystem catalog management (SSCM) and the associated commands and statements.

SPOOL V4.1A (BS2000/OSD)
User Guide

Target group
This manual is intended for nonprivileged users, Spool & Print administrators, RSO device administrators and systems support staff.
Contents
The manual describes the operation of SPOOL.

RSO V3.2A (BS2000/OSD)
Remote SPOOL Output
User Guide

Target group
This manual is directed at nonprivileged users, RSO device administrators, SPOOL administrators and systems support of BS2000/OSD.
Contents
The manual describes the functions and options of the user groups with respect to utilizing and controlling decentralized printers (RSO printers) and deals with the technical characteristics of all RSO printers.

AVAS V5.0A (BS2000/OSD)
AVAS Functions
User Guide

Target group
AVAS users
Contents
– Overview of AVAS functions
– Definition and scheduling of production
– Brief outline of administration

AVAS V5.0A / AVAS-SV V4.1C
(BS2000/OSD, UNIX, Windows NT))
AVAS for the Administrator
System Administrator Guide

Target group
AVAS administrators
Contents
– All AVAS administrator tasks, from system generation to system administration
– The AVAS-QUER utility routine
– The coupling of AVAS with MAREN
– AVAS reports
– BATCH functions
– External creation of AVAS elements
– Program interface
– AVAS-SV

AVAS (BS2000/OSD)
Job Management and Handling System
Introductory Guide

Target group
This manual is intended for all users wanting a basic introduction to the AVAS Job Management and Handling System.
Contents
The manual outlines the main features of AVAS. It explains the ways in which it benefits customers, describes its basic functions, recommends a procedure to be followed when using the system for the first time, and describes selected related products. For more detailed information you are referred to the AVAS user guides.

***open*FT V8.0 for BS2000/OSD**
Enterprise File Transfer in the Open World
User Guide

Target group
This manual addresses users who wish to transfer files or implement file management using *open*FT.
Contents
The manual describes the features of *open*FT. The description also covers the optional components *open*FT-AC for admission and access protection, and *open*FT-FTAM for supporting FTAM functionality. The command interface and messages are dealt with in detail.

***open*FT V8.0 for BS2000/OSD**
Enterprise File Transfer in the Open World
Installation and Administration
System Administrator Guide

Target group
This manual addresses administrators who want to use *open*FT, *open*FT-FTAM and *open*FT-AC on their BS2000 systems.
Contents
It describes how you install and start *open*FT and the optional components *open*FT-AC and *open*FT-FTAM. Operation and control of the *open*FT system are dealt with in detail. The command interface contains the description of all administrartor commands.

OMNIS (TRANSDATA, BS2000)
Administration and Programming
User Guide

Target group
– OMNIS administrators
– Programmers
Contents
Introduction to OMNIS administration, the OMNIS utility routines and the application interface for extending the OMNIS functionality
Applications
– Software development
– Application scheduling

SESAM/SQL-Server (BS2000/OSD)
Database Operation
User Guide

Target group
The manual is intended for SESAM/SQL system administrators.
Contents
The manual covers the options available to the system administrator for controlling and monitoring database operation.

SM2 (BS2000/OSD)
Software Monitor
Volume 1: Administration and Operation

Target group
This manual is addressed to users and systems support staff.
Contents
The monitoring system SM2 supplies users with statistical data on the performance of their DP systems and on resource utilization. Volume 1 of the manual describes operation of the SM2 monitor, the SM2 monitoring programs and the SM2 screen reports.
Analysis and display of the SM2 monitored data are dealt with in Volume 2.

SM2 (BS2000/OSD)
Software Monitor
Volume 2: Analysis and Display of SM2 Monitored Data

Target group
This manual is addressed to users and systems support staff.
Contents
The monitoring system SM2 supplies users with statistical data on the performance of their DP systems and on resource utilization. Volume 2 of the manual describes the SM2U1 utility routine for editing and administering the SM2 output files, and the analysis routines SM2R1, SM2R1-PC, SM2ONLINE-PC and SM2-PA.
Administration and operation of SM2 are described in Volume 1.

***open*UTM** (BS2000/OSD)
Generating and Handling Applications
User Guide

Target group
This manual is intended for application planners, technical programmers, administrators and users of UTM applications.
Contents
The manual describes the generation of UTM applications with distributed processing, the tools available with *open*UTM for this purpose, and the UTM objects created in the course of generation. It also contains all the information necessary for structuring, operating and monitoring a productive UTM application.

Unicenter TNG manuals

For literature about Unicenter TNG please refer to Computer Associates.

www.ca.com

Other relateted publications

Douglas Steedman
Abstract Syntax Notation One (ASN.1): The Tutorial and Reference
Isleworth, 1990
(ISBN 1-871802-06-7)

Marshall T. Rose
The Simple Book: An Introduction to Management of TCP/IP-based Internets
Prentice-Hall
(ISBN 0-13-812611-9)

RFCs

For comprehensive information concerning Requests for Comments (RFCs) please refer to the home page of the Internet Engineering Task Force (IETF):

www.ietf.org

Index

Contents

SNMP Management V6.0 (BS2000/OSD)

User Guide

Target group
The manual addresses network administrators/operators and system administrators who wish to integrate a BS2000 system in SNMP-based management or operate such a system
Contents
The manual describes the updates in SNMP Management V6.0 for BS2000/OSD as compared with version 5.0:

– new and functionally enhanced subagents and management applications
– extended integration in the CA Unicenter management platform
– added security

In any case where the functionality of version 6.0 of SNMP management for BS2000/OSD has not changed as compared with version 5.0, the description in the manual "SNMP Management V5.0 SNMP Management for BS2000/OSD" remains valid.

Edition: July 2002

File: snmp.pdf

This manual was produced by
cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Fujitsu Siemens computers GmbH
User Documentation
81730 Munich
Germany

Fax: (++49) 700 / 372 00000

e-mail: manuals@fujitsu-siemens.com
http://manuals.fujitsu-siemens.com

Comments
Suggestions
Corrections

Submitted by

Comments on SNMP Management V6.0
SNMP Management for BS2000/OSD

Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format *...@ts.fujitsu.com*.

The Internet pages of Fujitsu Technology Solutions are available at *http://ts.fujitsu.com/*...
and the user documentation at *http://manuals.ts.fujitsu.com*.

Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form *...@ts.fujitsu.com*.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter *http://de.ts.fujitsu.com/*..., und unter *http://manuals.ts.fujitsu.com* finden Sie die Benutzerdokumentation.