1 Preface

SNMP Basic Agent BS2000 V5.0 (SBA-BS2) and SNMP Standard Collection BS2000 V5.0 (SSC-BS2) provide the basic functionality for linking BS2000/OSD systems into SNMP-based management environments, e.g. the Unicenter TNG of COMPUTER ASSOCIATES. SBA-BS2 and SSC-BS2 provide network, system and application management capability from a central network management station, via SNMP. An integration package (SMBS2 or SMAWsmbs2) is supplied for integration in management platforms. The Console Monitor in SBA-BS2 comprises a management application (CMBS2 or SMAWcmbs2) on the management side. The SNMP subagent for SM2 (SSA-SM2-BS2), the SNMP subagent for *open*UTM (SSA-OUTM-BS2), and the subagents for the products *open*Net Server (BCAM subagent) and *inter*Net Services supplement the continuously increasing number and functionality of BS2000/OSD subagents. SNMP management for BS2000/OSD also permits web access to management information.

1.1 Contents of the manual

This manual describes the embedding of SBA-BS2, SSC-BS2 and the additive subagents SSA-SM2-BS2 and SSA-OUTM-BS2 into BS2000/OSD, the installation and configuration steps required for operation, and operation itself. The agents required for monitoring, together with their MIBs, are described in detail. The installation and configuration of SMBS2 / SMAWsmbs2 and PMBS2 / SMAWpmbs2 on the management station are also described. A separate chapter is devoted to web access.

1.2 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.

1.3 Summary of contents

This manual is structured as follows:

- Chapter 2: Overview

  This chapter leads into the SNMP architecture, introduces fundamentals and describes the embedding into BS2000/OSD/OSD and the functionality of SBA-BS2, SSC-BS2, SSA-SM2 and SSA-OUTM. The function of the master and subagents is also described.

- Chapter 3: Installation and configuration

  The installation requirements and installation itself are described in this chapter for SBA-BS2, SSC-BS2 and the additive subagents into BS2000/OSD, and for SMBS2 / SMAWsmbs2, CMBS2 / SMAWcmbs2 and PMBS2 / SMAWpmbs2 on the management station. The configuration steps are also shown from the viewpoint of BS2000/OSD and of the management station.

- Chapter 4: Operation

  Chapter 4 describes the BS2000/OSD commands for startup and shutdown of the master and subagents.

- Chapter 5: Functions of the BASIC AGENT

  This chapter describes the system and SNMP group for management via the master agent and supervisor subagent, as well as application monitoring via the Application Monitor subagent and console monitoring via the Console Monitor subagent. It also describes the functions of the HTML subagent for generating customized web sites.

- Chapter 6: Functions of the STANDARD COLLECTION

  The subagents in the STANDARD COLLECTION are described in chapter 6. It also includes a description of functions and complete listings of the MIBs concerned.

- Chapter 7: SNMP management for extended performance monitoring with SM2

  The topic of chapter 7 is the additive subagent for SM2-based performance monitoring. The functionality is described and the MIB is listed.

- Chapter 8: SNMP management for monitoring *open*UTM and *open*UTM applications

  The functionality and MIB of the additive subagent for monitoring *open*UTM applications are described in chapter 8.

- Chapter 9: Operating the management station

  Chapter 9 describes in detail the management applications available for use on management stations.

- Chapter 10: Web access to management information

  Chapter 10 describes the access to management information via the world wide web (WWW). The description of the web interface and the handling of the web interface is followed by an explanation of the how create and configure custom pages.

- Chapter 11: Trap server

  Chapter 11 describes the trap server offered for Solaris and Reliant UNIX. In addition to the trap server process, the command program for server configuration, the trap send program, and the trap receive program are also explained.

- Chapter 12: Configuration examples

  This chapter provides configuration examples for the topics basic monitoring, message monitoring, monitoring of applications and performance monitoring.

- Annex

  The Annex contains a list of the DCAM return codes (Application Monitor subagent).

This manual contains a series of Figures, which show the information displayed by the relevant subagents on the various management platforms and on the web interface.

1.4 Notational conventions

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

for general information

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.

1.5 Changes compared to the previous version

Version 5.0 of the SNMP management for BS2000/OSD supports the following new functions:

- Availability of the integration package SMAWsmbs2 under Solaris (in connection with the management platform Unicenter TNG).
- Support of a trap server for Solaris and Reliant UNIX
- Remote capability of the master agent for the SINIX2000 subagents.
- Trap Acknowledge group for the supervisor subagents.
- Addition of HSMS subagents to the STANDARD-COLLECTION (SSC-BS2).
- Support for a generic trap format in the Application Monitor subagents and in the Console Monitor subagents. Filters in file monitoring in the Application Monitor subagents.
- Addition of new traps to the *open*FT subagents and of new MIB objects which are sent in conjunction with the traps.
- Trap display in the web browser.

1.6 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-BS2.050.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 Overview

SNMP stands for **S**imple **N**etwork **M**anagement **P**rotocol and was developed as a protocol for network management services in TCP/IP networks. The original task of SNMP was only the monitoring and administration of LAN components such as bridges, routers, hubs, etc. in heterogeneous networks with TCP/IP protocols. In the meantime, the application range of SNMP has been extended to include system and application management. SNMP does not stand for the protocol alone, but rather for the complete corresponding management system, in the same way that the term TCP/IP designates the complete network rather than just the protocol as such.

The documents relevant to SNMP are stored in RFCs (Request for Comment) by the IAB (Internet Architecture Board) as is normally the case with TCP/IP. The basic RFCs for SNMPv1 (version 1) are:

– RFC 1155: “Structure and Identification of Management Information for TCP/IP-based Internets (SMI)”, May 1990

– RFC 1157: “A Simple Network Management Protocol (SNMP)”, May 1990

– RFC 1212: “Concise MIB Definitions”, March 1991

– RFC 1213: “Management Information Base for Network Management of TCP/IP-based Internets: MIB-II”, March 1991

SNMP Level 5 of the SNMP Management of BS2000/OSD also supports SNMPv3 (version 3). The following are the associated RFCs:

– RFC 2271: “An Architecture for Describing SNMP Management Frameworks”, January 1998

– RFC 2272: “Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)”, January 1998

– RFC 2273: “SNMPv3 Applications”, January 1998

– RFC 2274: “User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)”, January 1998

– RFC 2275: “View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)”, January 1998

2.1 Basic features of the SNMP management architecture

The management platform is the central component of an SNMP installation. The management platform is a master console with graphical terminal, which enables a well-structured display of the components it manages and offers easy operation. The network and all of its systems and applications can be monitored and controlled from the management platform. SNMP is not fixed to a particular platform.

The SNMP manager, also referred to as management station, resides in the management platform. The SNMP manager is an application that communicates with partner applications, the SNMP agents via SNMP over a TCP/IP network. Each managed component has an agent that provides the SNMP manager with current information about the component. 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.

The basis for managing the components concerned is an exact description of the parts of these components that are to be administered (objects) in the MIB (Management Information Base). The MIB is the informational backbone of each Management Agent. It contains information on the characteristics, such as name, syntax, access rights and state, of each separate component. Specific MIBs are supplied by many hardware and software component manufacturers. The MID coding is carried out in ASN.1 (Abstract Syntax Notation One). ASN.1 has been ratified by the ISO as a standard for the presentation layer (see ISO/IEC 8824 and 8825).

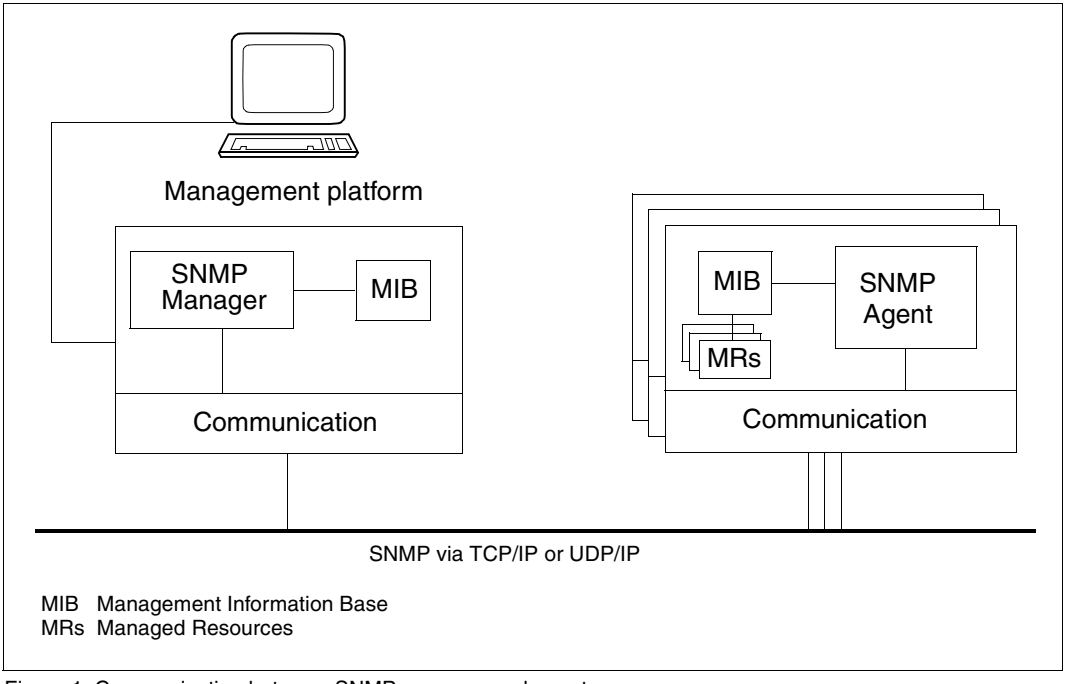


Figure 1: Communication between SNMP manager and agents

SNMP protocol elements

The information is transported over the network by means of function-dependent SNMP protocol elements. SNMPv1 only requires four different protocol elements for requesting, setting and displaying values that contain the relevant management information (object values). A fifth protocol element, trap, is used by the agents for asynchronous reporting of important events.

Protocol element	Type	Function
GetRequest-PDU	0	Read request from the manager for a specific defined object
GetNextRequest-PDU	1	Read request from the manager for the next (unknown) object
GetResponse-PDU	2	Reply from the agent with the requested values
SetRequest-PDU	3	Write request from the manager to a specific, defined object
Trap-PDU	4	Asynchronous report from the agent when special events occur

SNMPv1 protocol elements

The structure of the actual SNMP message is quite simple. It consists of the SNMP header and PDU (Protocol Data Unit). The SNMP header contains a version ID and the Community Name.

The PDU consists of the field for the PDU type and a list of

– variables to be read (for GetRequest and GetNextRequest) or
– the variable to be set (for SetRequest).

Each variable consists of the name of the object monitored and the associated value. The list of variables that belong to an SNMP message is referred to as variable bindings.

2.2 SNMP management in BS2000/OSD - embedding and functionality

TransView offers three solutions with different targets for connecting BS2000/OSD to an SNMP management system.

- The products SNMP Basic Agent BS2000 V5.0 (SSA-BS2) and SNMP Standard Collection BS2000 V5.0 (SSC-BS2) offer the capability of integrating BS2000/OSD systems directly into SNMP-based management platforms such as Unicenter TNG, Transview or OpenView. SNMP Basic Agent and SNMP Standard Collection allow network, system and application management via an implementation of the SNMP protocol in BS2000/OSD. The SNMP Basic Agent also permits access to management information via HTTP/HTML (see chapter “Web access to management information” on page 399).
  The SNMP subagent for SM2 (SSA-SM2-BS2), the SNMP subagent for *open*UTM (SSA-OUTM-BS2) and the subagents for the products *open*Net Server and *inter*Net Services complement the continuously increasing number and functionality of BS2000/OSD subagents.
- Within the framework of the HIPLEX concept, it is additionally possible to monitor BS2000/OSD systems in the start-up of shutdown phases, switch computers on and off centrally from the management station with POWER ON/POWER OFF, as well as executing all further SKP activities. A prerequisite for this is the use of a management platform (CA Unicenter TNG or TransView Control Center) together with LAN networking (TCP/IP) of the SKP consoles concerned and the HIPLEX OP product (please see the HIPLEX OP manual for further information).
- The SNMP Proxy BS2000/PDN (UNIX) product uses the NMCP protocol in the SNMP protocol to link up the TRANSDATA world. A UNIX computer is used as the gateway. In this way, it is possible to operate a TRANSDATA network for BS2000/OSD, PDN, SINIX and INCA systems with proprietary TRANSDATA protocols (NMCP).
- The HNC (**H**igh-Speed **N**et **C**onnect) is also incorporated in the central SNMP management of a heterogeneous system network.

Figure 2 on the next page gives an overview of the SNMP integration of BS2000/OSD.

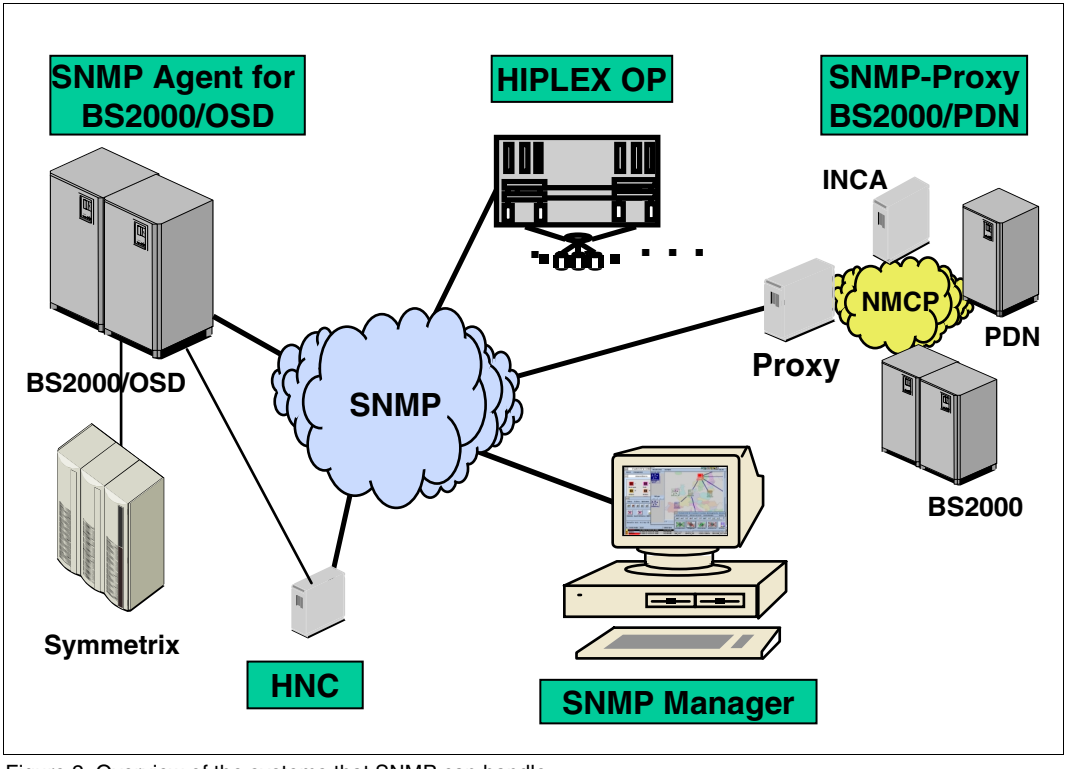


Figure 2: Overview of the systems that SNMP can handle

2.2.1 Product structure

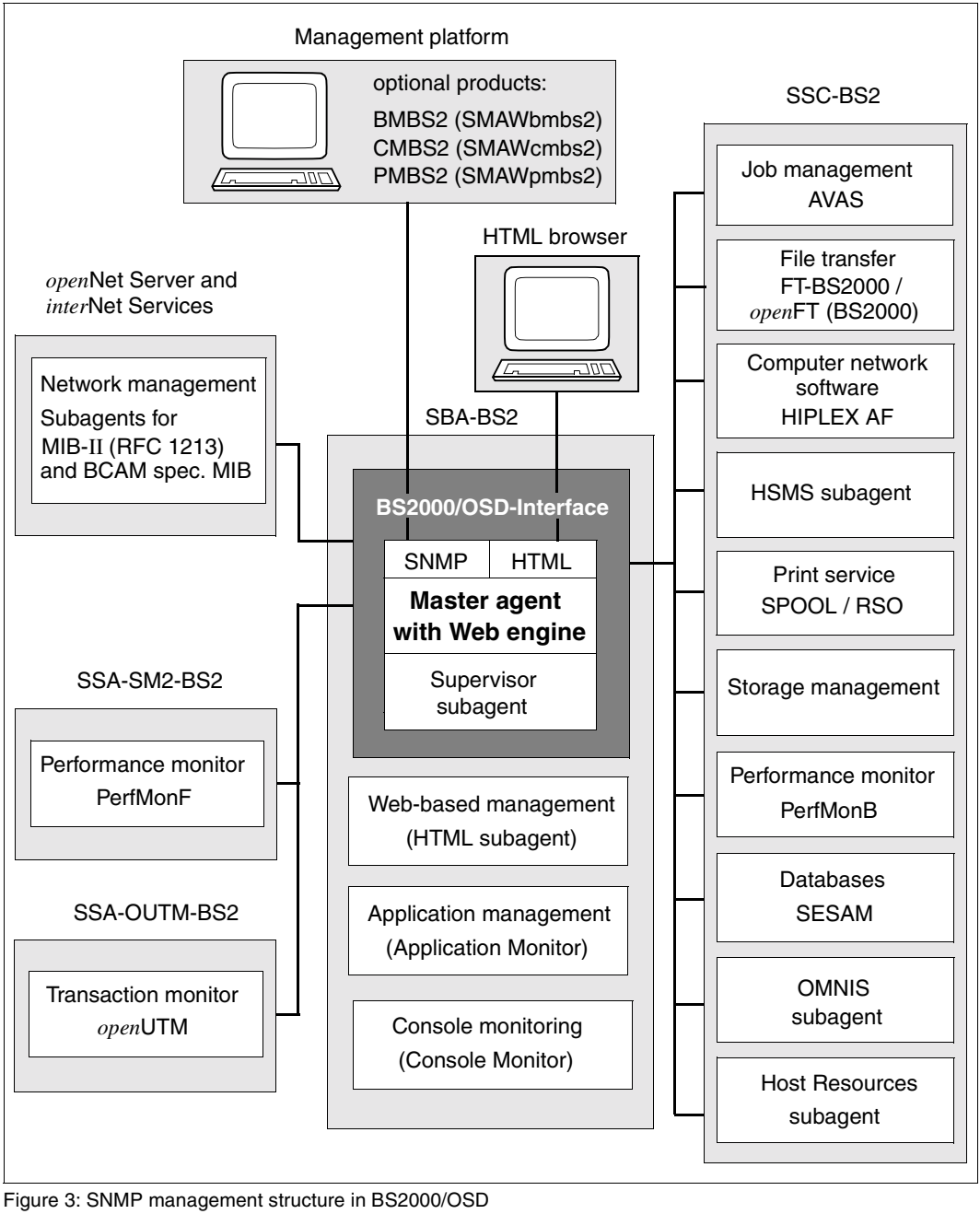


Figure 3: SNMP management structure in BS2000/OSD

SBA-BS2 (SNMP-Basic-Agent BS2000)

The master agent is supplied with the SBA-BS2 V5.0 delivery unit. The delivery unit also contains the “Supervisor Subagent”, “Application Monitor Subagent”, “Console Monitor Subagent” and the “HTML subagent”.

- The master agent is the BS2000/OSD communication partner for the management station, 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, tasks, job variables, BS2000/OSD subsystems and log files are monitored by the Application Monitor. It also monitors the logging 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 is used for console monitoring. On the one hand, it allows you 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 station and query the results.

  The associated management application contained on the CD-ROM supplied enables the display of console messages and easy console operation for all BS2000/OSD systems.
- The HTML subagent enables the definition of custom pages for web access to management information of BS2000/OSD.

SBA-BS2 also contains two SDF commands for sending traps.

SSC-BS2 (SNMP-Standard-Collection BS2000)

SSC-BS2 V5.0 includes a set of subagents for BS2000/OSD-specific management tasks.

- 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 HIPLEX-AF subagent informs about the current configuration in the HIPLEX network and notifies all relevant 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 databases supplies information on SESAM databases and SESAM 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 station 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.

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 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 subagent SSA-OUTM-SX is provided.

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

An MIB-II subagent compliant with RFC 1213 is available for network management. A subagent that supplies information on BCAM-specific settings and values is also offered.

The following additional products are offered:

Proxy agent

If specific subnetworks within a heterogeneous network have to be managed with proprietary network management products, SNMP offers the option of management via a proxy agent. The proxy agent provides gateway functionality in the network management, which allows the complete subnetwork to be connected to the SNMP management while still ensuring the existence of the proprietary network management within this subnetwork. The SNMP protocol elements are converted by the proxy agent into the corresponding elements of the proprietary network management system and vice versa. The SNMP proxy agent BS2000/PDN is provided for connecting the TRANSDATA world with the systems PDN and INCA to the SNMP management.

HIPLEX OP

HIPLEX operation uses an interface to the SINIX2000-based service console processors of the BS2000/OSD server and offers a broad palette of administration, control and monitoring functions. The functional spectrum of HIPLEX OP is between POWER ON and POWER OFF. HIPLEX OP supplies the management station with service processor messages as well as BS2000/OSD system console and VM2000 messages. The events from the SINIX2000 event screen can also be sent to the management station. A large number of standard filters are available for filtering the messages.

HNC-SNMP

SNMP management for the HNC 91849, HNC-II 91850 and HNC-III 91851 channel adapters allows comprehensive monitoring of the components in the HNC. The various HNC components are each assigned separate MIBs. The HNC component management is supported by extensive alarm management in the form of trap MIBs.

2.2.2 Structure of the SNMP agent in BS2000/OSD

BS2000 is connected to SNMP via a LAN connection that uses TCP/IP protocols. A network management agent that can handle the SNMP protocol elements is installed in BS2000. The functionality of the SNMP agents is split into one master and a number of subagents. The advantages of this solution include reliability and user-friendliness with regard to maintenance and modification overhead.

The basis of this solution is the product EMANATE from the company SNMP Research. EMANATE was ported into BS2000 and is also available on Reliant UNIX and UNIX derivatives from various well-known manufacturers as well as for DOS and DOS/Windows.

2.2.2.1 Master agent

The current requirements for the SNMP subagents in an end system go beyond normal network management. They extend over system and application management up to the management of middleware (transaction systems and databases). Because of the manifold requirements, with large end systems in particular, the desire arises to be able to employ a number of task-specific agents: this is supported by the structuring into master and subagents.

The subagents are subordinate to the master agent. This contains the basic functions, such as SNMP protocol processing, safety functions and management of the subagents, and can also run without subagents. The master agent is therefore also responsible for outputting and setting the values of the system and SNMP group of MIB-II and other standardized MIBs (RFC 2272 - 2275).

Besides processing SNMP requests, the master agent also provides access to management information via HTTP using the World Wide Web (web-based management, see page 399 ff). This means that it is also possible to call up a web page that displays incoming traps in tabular form (see page 416).

The option of being able to start and stop subagents separately simplifies the updating and use of individual subagents without having to run the complete management system down. It also allows interrupt-free management of the remaining system if a component fails, as well as the parallel processing of different subagent jobs.

The management station only communicates with the master agent. The master and subagents communicate with each other via an asynchronous message interface. The asynchronous message interface guarantees high master agent performance with job processing because it is not blocked while processing lengthy jobs, but instead can process further SNMP requests in parallel using multithreading.

The master agent has remote capabilities when using SINIX2000 subagents, i.e. the master agent and subagents need not be running in the same operating system.

2.2.2.2 Subagents

The subagents are only functional if the master agent is working. In the initialization phase, the subagent signs on to the master agent and passes its MIB to the master agent.

Subagents are event-oriented. The subagent goes into a wait loop after initialization. It leaves the loop on arrival of an event that it must process. Requests from the master agent, timer expiry or the arrival of an agreed signal, are examples of events. The subagent goes back into its wait loop after it has processed all existing events.

The supervisor subagent plays a special role among the subagents. It does act as an autonomous subagent, but can only be started in conjunction with the master agents and runs in the same task.

2.2.3 User interface for the SNMP management of BS2000/OSD

The standard protocol SNMP enables the connection of BS2000/OSD systems to any management platform that supports SNMP. This is the case for all market-relevant management platforms. The management platforms of the various manufacturers offer a variety of features. The strategic management platforms recommended by Fujitsu Siemens Computers, namely CA Unicenter TNG, TransView and HP OpenView are universal and have a sophisticated alarm management system with numerous options for linking reactions to events.

Integration packages and management applications

Fujitsu Siemens Computers offers integration packages (SMBS2 and SMAWsmbs2) for the management platforms stated, which enable the automatic integration of BS2000/OSD into these management platforms. These integration packages include additional features for the user interface.

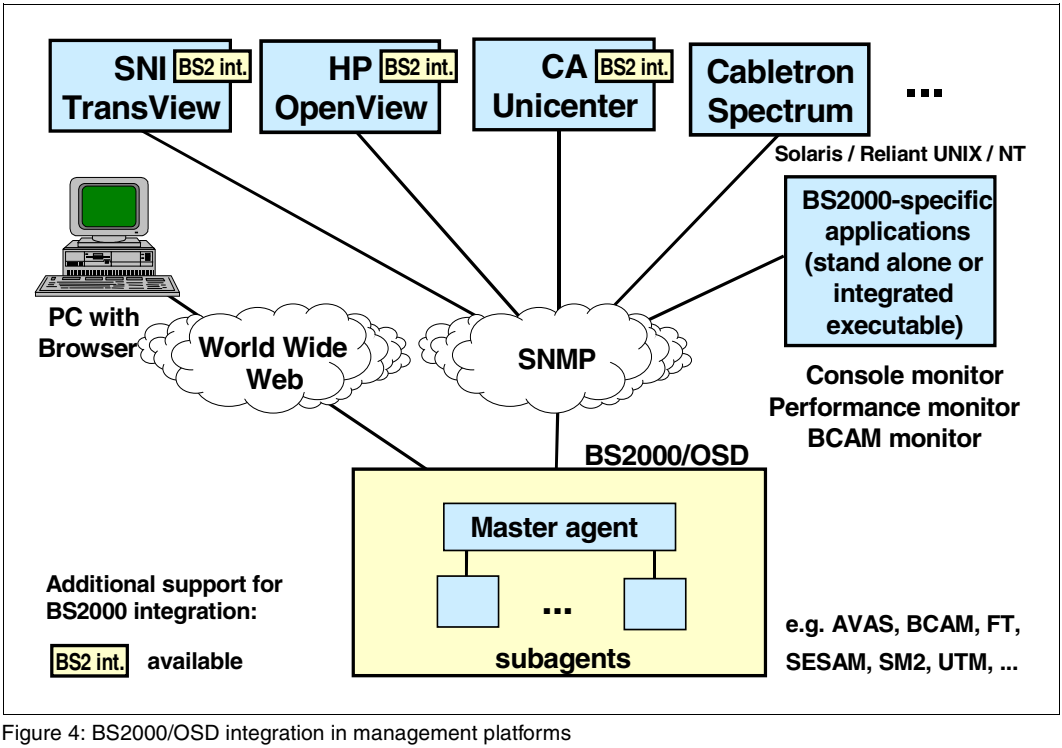


Figure 4: BS2000/OSD integration in management platforms

An integration package is available in the following products:

BS2-SNMP-SO (for Solaris): SMAWsmbs2

BS2-SNMP-SX (for Reliant Unix): SMBS2

BS2-SNMP-NT (for Windows NT): SMBS2

Besides the integration packages, the products BS2-SNMP-SO, BS2-SNMP-SX and BS2-SNMP-NT contain management applications for special subagents tailored to the specific attributes and tasks of the particular subagent. These management applications can be installed on the management platform. They augment and enhance the representation and handling of the existing management platform. The management applications can also be used separately on a Solaris, Reliant UNIX, or Windows NT system.

Overview of all integration packages

The following integration packages are offered for strategic management platforms:

Integration packages	**Associated agents**	**Management station**	**Operating system**
SMAWsmbs2 from BS2-SNMP-SO	all subagents	CA Unicenter TNG-Base	Solaris
SMBS2 from BS2-SNMP-SX	all subagents	TransView SNMP / TransView Control Center/ OpenView NetWork Node Manager	Reliant UNIX
SMBS2 from BS2-SNMP-NT	all subagents	CA Unicenter TNG Framework / CA Unicenter TNG-Base	Windows NT

Overview of management applications

The following management applications are offered:

Management applications	Package name	Associated agents	Management station	Operating system
BCAM-Monitor BMBS2	BMBS2 for Solaris: SMAWbmbs2	BCAM subagent, MIB-II subagent	standalone / integrated	Solaris / Reliant UNIX / Windows NT
Console Monitor CMBS2	CMBS2 for Solaris: SMAWcmbs2	Console Monitor Subagent (SBA-BS2)	standalone / integrated	Solaris / Reliant UNIX / Windows NT
Performance Monitor PMBS2	PMBS2 for Solaris: SMAWpmbs2	Performance monitor subagent (SSA-SM2-BS2)	standalone / integrated	Solaris / Reliant UNIX / Windows NT

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). The web access is described in the chapter “Web access to management information” (see page 399).

3 Installation and configuration

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

– SBA-BS2 V5.0

– SSC-BS2 V5.0

– SSA-SM2-BS2 V5.0

– SSA-OUTM-BS2 V5.0

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 CD-ROM contains the following for both Reliant UNIX and Windows NT:

– an SMBS2 V5.0 integration package with extensions for the management platforms CA Unicenter TNG (on Windows NT) or TransView SNMP, TransView Control Center, or HP OpenView NNM (on Reliant UNIX),

– the management applications BMBS2 V5.0 (BCAM Monitor), CMBS2 V5.0 (Console Monitor application) and PMBS2 V5.0 (Performance Monitor application for use with SSA-SM2-BS2),

– the associated Interpreter tclset V5.0,

– the trap server trpsrv (for Reliant UNIX only),

– all BS2000/OSD-specific MIBs in ASN.1 format.

For Solaris, the CD-ROM contains:

– an SMAWsmbs2 V5.0 integration package with extensions for the management platform CA Unicenter TNG on Solaris,

– the management applications SMAWbmbs2 V5.0 (BCAM Monitor), SMAWcmbs2 V5.0 (Console Monitor application) and SMAWpmbs2 V5.0 (Performance Monitor application for use with SSA-SM2-BS2),

– the associated SMAWtcl V5.0 interpreter,

– the trap server SMAWtrpsv,

– all BS2000/OSD-specific MIBs in ASN.1 format.

The SNMP agents are hardware-independent. They run on all central processing units (including RISC-based models) supported by BS2000/OSD as of V2.0 or OSD-SVP 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 V5.0 requires the following software:

– BS2000/OSD-BC ≥ V2.0 or OSD-SVP V2.0
– SBA-BS2 V5.0
– AVAS ≥ V3.0
– FT-BS2000 V5.2 or *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 ≥ V2.1B50***
– SM2 ≥ V11.2
– JV ≥ V11.2

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 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 are also required.

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

SBA-BS2 V5.0 and DCAM as of V13.0 or *open*Net Server V1.0 are required to implement the MIB-II subagent. The BCAM subagent (private MIB) can run with DCAM V14.0 or later.

Software requirements for the integration packages SMBS2 and SMAWsmbs2

In order to use SMBS2 on Windows NT or SMAWsmbs2 on Solaris, Unicenter TNG Version 2.2 must be installed and available. If only CA Unicenter Framework is installed, SMBS2 can be used with restrictions on Windows NT.

If SMBS2 is used on a TransView management platform, TransView SNMP ≥ V3.1 and TransView Control Center ≥ V3.1 are needed.

If SMBS2 is used on an OpenView-based management station, OpenView Version 3.3 or 4.1 is required.

BMBS2, CMBS2 and PMBS2 require installation of the interpreter Tcl-Set ≥ V5.0 (see page 121).

3.2 Installation 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.

The integration package SMBS2, the management applications from the packages BMBS2, CMBS2 and PMBS2 as well as the interpreter tclset are installed on the management station in Windows NT or Reliant UNIX environment (see section “Integration in the management platforms” on page 91).

The integration package SMAWsmbs2, the management applications from the packages SMAWbmbs2, SMAWcmbs2 and SMAWpmbs2, as well as the interpreter SMAWtcl, are installed on the management station in the Solaris environment (see the section “Integration in the management platforms” on page 91).

SBA-BS2, SSC-BS2, SSA-SM2-BS2 and SSA-OUTM-BS2 are installed with the SOLIS2 software delivery and information system. Where necessary, the SOLIS2 installation includes BS2000/OSD-specific jobs 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.

3.2.1 Installing SBA-BS2 and SSC-BS2

The POSIX subsystem must be running. The executable agents of SBA-BS2are in SINLIB.SBA-BS2.050. 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: 050

SINLIB.SSC-BS2.050 contains the executable agents 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: 050

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

If SSA-SM2-BS2 is not installed with IMON, the SYSSII-file (IMON V2.0) must be added subsequently.

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 V5.0

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 5.0. 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 “Installation”
Product version: <prod-version>

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

3.3 Configuring the agent in BS2000/OSD

Configuration work is necessary both on the management platform and in the BS2000/OSD:

- The work to be carried out in BS2000/OSD is described in the sections listed below.
- The work necessary on the management platform is described on page 91.
  - Information concerning configuration work on a Unicenter TNG-based management platform is given on page 96 ff.
  - Information concerning configuration work on a TransView SNMP-based management platform is given on page 107 ff.
  - Information concerning configuration work on a TransView Control Center-based management platform is given on page 112 ff.
  - Information concerning configuration work on a OpenView-based management platform is given on page 117 ff.

3.3.1 Security configuration

The central task of any security system is to check whether a user is authorized to carry out the requested operations. In this way, the security system protects the SNMP agents against unauthorized access to MIB variables. Only a user who sends requests with a community string that is permitted and configured at the agent can perform the requested operation.

3.3.1.1 Security mechanisms

Each message sent to the agent from the SNMP manager is checked according to the following criteria:

– Who is sending the message? (*who?*, authentication)

– What operation is requested? (*what?*, authorization)

– Which objects in the MIB are affected by the operation? (*where?,* access check)

– How was the request sent? (*how?*, security level)

The security system compares the message with the security configuration at the agent. Depending on the result of this comparison, the security system then permits execution of the request operation or rejects it.

Figure 5 outlines the security mechanism.

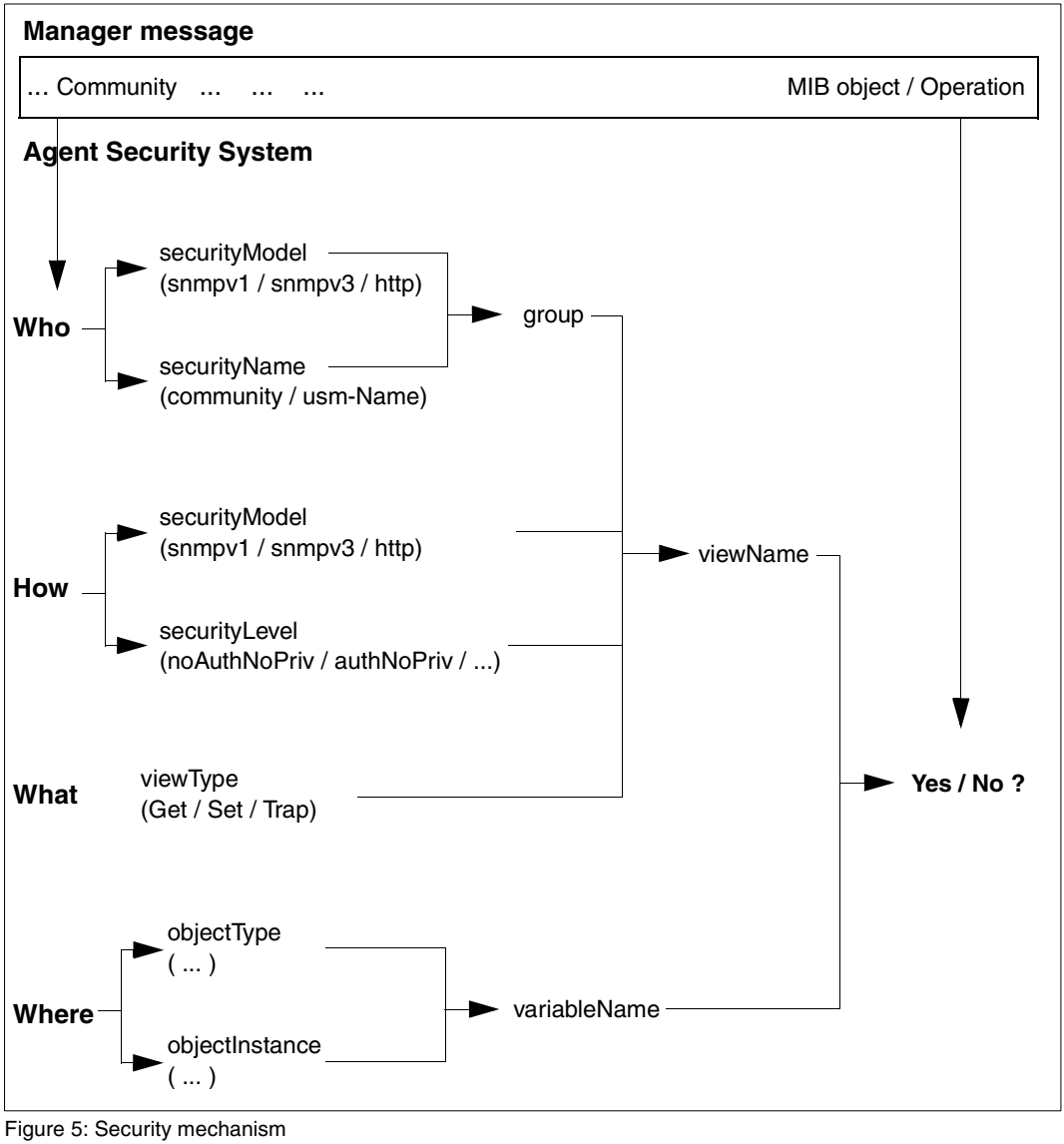


Figure 5: Security mechanism

3.3.1.2 Extended security mechanisms in SNMPv3

The current version of the SNMP management system for BS2000/OSD uses elements of the comprehensive security concept of SNMPv3. It does not matter whether the SNMP manager sends the message as sNMPv1 or SNMPv3 requests in this context. Compared to the original SNMPv1 security mechanisms, the security concept now adopted offers the following extensions:

– Selective assignment of access rights for MIB variables

– Definition of access rights for a group of management stations

– Detailed trap reception

– Authentication of SNMP requests

Selective assignment of access rights for MIB variables

In earlier version of the SNMP management system for BS2000/OSD, which were based only on SNMPv1 security mechanisms, only the same access rights could be assigned for all MIB variables. Thus, a community string with defined write access had write access to *all* MIB objects that were defined as modifiable in its MIB. The same applied analogously for read access: either *all* MIB objects could be read, or none.

The current version of the SNMP management system for BS2000/OSD enables a community string to restrict read and/or write access to a particular MIB branch. For example, you can define a community string with read access for all MIB-II Objects, but only write access for objects of the system group defined as modifiable in the MIB-II, except for *sysName*.

The access definition can even be defined on the instance level. In this way, you can restrict access to the second instance of the interface table, for example.

Defining access rights to a group of management stations

In earlier versions, it was also possible to restrict the permission to send request to individual management stations. This entailed entering the community string along with the IP address of the host in the configuration file for each host to be granted the permission. To extend the access rights to a subset of management stations, however, you had to enter “0.0.0.0”, which granted access to *all* hosts. Now, in the current version, it is possible to specify a family of IP addresses via a bit mask, which then have access via the defined community string.

Example:

You can define a community string to grant access to the agents of systems with the IP addresses 139.25.104 - 139.25.255.

Detailed trap reception

In earlier security versions, traps were *always* sent to *all* the trap destinations defined in the security configuration. In contrast, the current version allows you to define the IP address of the system according to Enterprise and trap number.

Example

You can specify that all traps with enterprises sni.2.34 are to be sent to system 139.22.22.22.

Request authentication

Management stations that send their request via the SNMPv3 protocol can authenticate their message. This enables the agent to check that the message has neither been changed nor held back in the meantime.

3.3.1.3 Configuration steps

The security configuration for an agent encompasses the following steps:

1. a) User configuration

   The user configuration depends on the security model and is carried out

   – for SNMPv1 by a *communityEntry* (currently standard),

   – for SNMPv3 by an *usmUserEntry*,

   – for HTTP by an *httpUserNameEntry*.

   b) Configuration of the access check

   1. *vacmViewTreeFamiliyEntry* defines an MIB branch.
   2. *vacmAccessEntry* defines a security group and this group is assigned the MIB branches defined at 1) for read and write access and for the trap.
   3. *vacmSecurityToGroupEntry* assigns the user configured at a) to a security group.

   c) *snmpTargetAddressEntry* performs the access check.

2. a) Definition of trap destinations
    1. *snmpNotifyEntry* defines a notify entry.
    2. *snmpTargetAddrEntry* assigned the notify entry to a target address and a target parameter entry.
    3. *snmpTargetParamsEntry* defines a target parameter with community string.

    b) Configuration of the access check
    1. *vacmViewTreeFamiliyEntry* defines an MIB branch.
    2. *vacmAccessEntry* defines a security group and this group is assigned the MIB branches defined at 1) for read and write access and for the trap.
    3. *vacmSecurityToGroupEntry* assigns the community string (a3), with which the trap is to be sent, to a security group.

    c) *snmpNotifySourceEntry* specifies the sender address of the trap.

3.3.1.4 Configuration file *snmpd.cnf*

All information relevant for the security configuration are entered in the form of security entries in the configuration file */etc/snmp/agt/snmpd.cnf*.

Each security entry in *snmpd.cnf* has the following format:

TAG value

Meaning:

– *TAG* specifies the type of security entry
– *value* specifies a valid value for the configuration.

Syntax conventions

The following conventions are used in the section below to describe the security entries:

– Items in **Bold** type must not be changed.
– items in *italics* must be replaced in *snmpd.cnf* by the current values.
– The characters [,], (,), *, | are meta characters and must not be specified in *snmpd.cnf*.
– Items enclosed in square brackets "[...]" must not be specified. Instead, you can specify "-" for these items in *snmpd.cnf*; the default value then applies.
– Item enclosed in round brackets "(...)" and delimited with "|" represent alternatives; exactly one item must be specified.
– Items marked with an asterisk (*) are keywords, which uniquely define the relevant entry.

– Entries in *snmpd.cnf* can extend over several lines, if a backslash (\) is placed at the end of the line to indicate that it continues.

– Blanks, spaces and carriage returns are ignored.

– Character strings that contain blanks must be enclosed in double-quotes (“...”).

3.3.1.5 Definition of access to the agent via SNMPv1 requests

a) Definition of a community string - *communityEntry*

The *communityEntry* defines a community string and assigns it a security group and a transport label.

Tag:	**communityEntry**
Value:	**localSnmpID** *MyCommunity MyGroup* **localSnmpID -** [*MyTransTag*] **nonVolatile**

*MyCommunity**
Community string (String 1..255) to be used for an SNMPv1 request.

MyGroup
Assigned security group (see b2). The security group defines the scope of authorization.

MyTransTag
Assigned transport label. The transport label refers to a list of target tags in *snmpTargetAddrEntry* (see c) and thus defines the systems from which requests are accepted.
Default value: no restriction of authorization

b) Definition of the access check

b1) Definition of the MIB branch - *vacmViewTreeFamilyEntry*

The definition of the MIB branch consists of one or more *vacmViewTreeFamilyEntry* entries. Each *vacmViewTreeFamilyEntry* assigns an OID to the MIB branch or excludes an OID.

Tag:	**vacmViewTreeFamilyEntry**
Value:	*MyMIB MyOID* - (**included** | **excluded**) **nonVolatile**

*MyMIB**
Name of the MIB branch entry (String 1..32)

*MyOID**
OID or symbolic name of the MIB branch to be included or excluded.

included | **excluded**
The MIB branch is to be included or excluded. Only objects that result from all *included* or *excluded* operations remain in the MIB branch.

b2) Definition of the security group - *vacmAccessEntry*

The *vacmAccessEntry* defines a security group and assigns it MIB branches for read and write access.

Tag:	**vacmAccessEntry**
Value:	*MyGroup* **- snmpv1 noAuthNoPriv exact** [*MYRead*) [*MyWrite*] **- nonVolatile**

*MyGroup**
Name of the security group (String 1..32). The security group defines the scope of authorization.

MyRead
Assigned MIB branch for read access (see b1)

MyWrite
Assigned MIB branch for write access (see b1)

b3) Definition of the security entry - *vacmSecurityToGroupEntry*

vacmAccessEntry defines a security group and assigns it MIB branches for read and write access

Tag:	**vacmSecurityToGroupEntry**
Value:	**snmpv1** *MyCommunity MyGroup* **nonVolatile**

*MyCommunity**
Community string (String 1..255) to be used for an SNMPv1 request.

MyGroup
Assigned security group (see b2). The security group defines the scope of authorization.

c) Definition of the address check - *snmpTargetAddrEntry*

snmpTargetAddrEntry specifies the system from which access may be made.

Tag:	**snmpTargetAddrEntry**
Value:	*MyTarget* **snmpUDPDomain** *MyTaddr* **300 0** *MyTagList* **- nonVolatile** *MyAddrMask*

*MyTarget**
Name of the target string (String 1..32)

MyTAddr
Interne address of the target, i.e. the system which may access, in the form **xxx.xxx.xxx.xxx:0**

MyTagList
List of tags (see a). The list must be enclosed in quotes (“...”); the individual list entries must be delimited by *one* blank.

MyAddrMask
Mask in the form **xxx.xxx.xxx.xxx:0**, analogous to a subnet mask.
A sender address is valid if:
(sender address & MyAddrMask) == (MyTAddr & MyAddrMask)

Examples

Example 1

All systems are to have read access with the community string “public”.

– in SNMPv1 security (earlier versions): **community** public 0.0.0.0 read 1

– in SNMPv3 security (current version):

communityEntry localSnmpID public READ **localSnmpID -** - **nonVolatile**
vacmSecurityToGroupEntry snmpv1 public READ **nonVolatile**
vacmAccessEntry READ **- snmpv1 noAuthNoPriv exact** All - - **nonVolatile**
vacmViewTreeFamiliyEntry All dod - **included nonVolatile**

Example 2

System 139.22.22.22 is to have write with the community string “master”.

– in SNMPv1 security (earlier versions): **community** master 139.22.22.22 write 1

– in SNMPv3 security (current version):

communityEntry localSnmpID master WRITE **localSnmpID -** TarTag1 **nonVolatile**
vacmSecurityToGroupEntry snmpv1 master WRITE **nonVolatile**
vacmAccessEntry WRITE **- snmpv1 noAuthNoPriv exact** All All **- nonVolatile**
vacmViewTreeFamiliyEntry All dod **- included nonVolatile**
snmpTargetAddrEntry Ta1 **snmpUDPDomain** 139.22.22.22:0 300 0 TarTag1 **-**\
nonVolatile 255.255.255.255:0

Example 3

The community string “multi” is to have read access from systems with the IP addresses 139.22.104.0 to 139.22.111.255.

– in SNMPv1 security (earlier versions): not settable

– in SNMPv3 security (current version):

communityEntry localSnmpID multi WRITE **localSnmpID -** TarTag2 **nonVolatile**
vacmSecurityToGroupEntry snmpv1 multi WRITE **nonVolatile**
vacmAccessEntry WRITE **- snmpv1 noAuthNoPriv exact** All All **- nonVolatile**
vacmViewTreeFamiliyEntry All dod **- included nonVolatile**
snmpTargetAddrEntry Ta2 **snmpUDPDomain** 139.22.22.22:0 300 0 TarTag2 **-**\
nonVolatile 255.255.248.0:0

Example 4

All systems are to have read access to the system group only with the community string “sysread”.

– in SNMPv1 security (earlier versions): not settable

– in SNMPv3 security (current version):

communityEntry localSnmpID sysread SysAccR **localSnmpID - - nonVolatile**
vacmSecurityToGroupEntry snmpv1 sysread SysAccR **nonVolatile**
vacmAccessEntry SysAccR **- snmpv1 noAuthNoPriv exact** SysTreeR - **-** \
nonVolatile
vacmViewTreeFamiliyEntry SysTreeR system **- included nonVolatile**

Example 5

The community string “syswrite” is to have write access from all systems with Internet addresses 139.22.104.0 - 139.22.111.255. The access should only be permitted to the system group, except for sysName.

– in SNMPv1 security (earlier versions): not settable

– in SNMPv3 security (current version):

communityEntry localSnmpID syswrite SysAccW **localSnmpID** - TarTag2 ****
nonVolatile
vacmSecurityToGroupEntry snmpv1 syswrite SysAccW **nonVolatile**
vacmAccessEntry SysAccW **- snmpv1 noAuthNoPriv exact** SysTreeR SysTreeW **-**\
nonVolatile
vacmViewTreeFamiliyEntry SysTreeR system **- included nonVolatile**
vacmViewTreeFamiliyEntry SysTreeW system **- included nonVolatile**
vacmViewTreeFamiliyEntry SysTreeW sysName **- excluded nonVolatile**
snmpTargetAddrEntry Ta2 **snmpUDPDomain** 139.22.104.0:0 300 0 TarTag2 **- **
nonVolatile 255.255.248.0:0

Figure 6 on the following page outlines the procedure when creating entries for the security configuration.

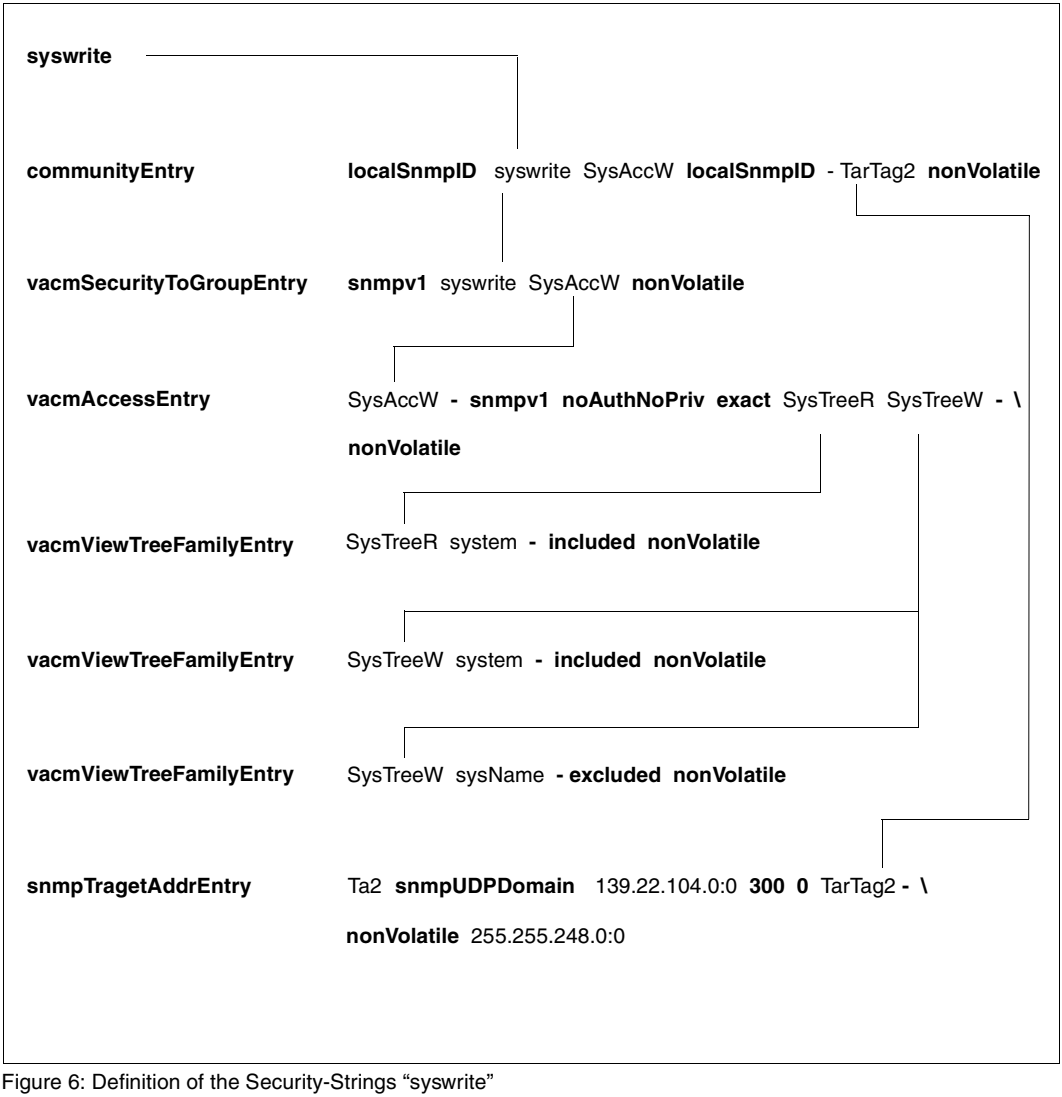


Figure 6: Definition of the Security-Strings “syswrite”

3.3.1.6 Definition of access to agent via SNMPv3 requests

a) Definition of a SNMPv3 user - *usmUserEntry*

usmUserEntry assigns a security group and a transport label to an SNMPv3 user

Tag:	**usmUserEntry**
Value:	**localSnmpID** *MyUser MyAuthProt* **usmNoPrivProtocol nonVolatile** [*MyTransTag*] [*MyAuthKey*]

*MyUser**
User name (String 1..32 for which an SNMPv3 request is to e permitted.

MyMyAuthProt
Authentication protocol:
– usmNoAuthProtocol (no authentication)
– usmHMACMD5AuthProtocol (authentication)

MyTransTagList
Assigned transport label. The transport label points to a target list (see e) and thus defines the systems from which requests are to be accepted.
"-" means that there is no restriction.

MyAuthKey
Password for authentication.+
"-" means that there is no password.

b) Definition of the access check

b1) Definition of the MIB branch - *vacmViewTreeFamilyEntry*

The definition of the MIB branch consists of one or more *vacmViewTreeFamilyEntry* entries. Each *vacmViewTreeFamilyEntry* assigns an OID to the MIB branch or excludes an OID.

Tag:	**vacmViewTreeFamilyEntry**
Value:	*MyMIB MyOID* **-** (**included** | **excluded**) **nonVolatile**

*MyMIB**
Name of the MIB branch entry (String 1..32)

*MyOID**
OID or symbolic name of the MIB branch to be included or excluded.

included | **excluded**
The MIB branch is to be included or excluded. Only objects that result from all *included* or *excluded* operations remain in the MIB branch.

b2) Definition of the security group - *vacmAccessEntry*

vacmAccessEntry defines a security group and assigns it MIB branches for read and write access.

Tag:	**vacmAccessEntry**
Value:	*MyGroup* **- usm noAuthNoPriv exact** [*MyRead*] [*MyWrite*] **- nonVolatile**

*MyGroup**
Name of the security group (String 1..32). The security group defines the scope of authorization.

MyRead
Assigned MIB branch for read access (see b1)

MyWrite
Assigned MIB branch for write access (see b1)

b3) Definition of the security entry - *vacmSecurityToGroupEntry*

vacmSecurityToGroupEntry assigns a security group to the SNMPv3.

Tag:	**vacmSecurityToGroupEntry**
Value:	**usm** *MyUser MyGroup* **nonVolatile**

*MyUser**
User ID (Character-String 1..255) that may make an SNMPv3 request.

MyGroup
Assigned security group (see b2). The security group defines the scope of authorization.

c) Definition of the address check - *snmpTargetAddrEntry*

snmpTargetAddrEntry specifies the system for which access is to be granted.

Tag:	**snmpTargetAddrEntry**
Value:	*MyTarget* **snmpUDPDomain** *myTaddr* **300 0** *MyTagList* **- nonVolatile** *MyAddrMask*

*MyTarget**
: Name of the target entry (String 1..32)

MyTAddr
: Internet address of the target in the form **xxx.xxx.xxx.xxx:0**

MyTagList
: List of tags (see a)). This list must be enclosed in double-quotes (“...”); the individual list elements must be delimited by *one* blank.

MyAddrMask

: Mask in the form **xxx.xxx.xxx.xxx:0**, analogous to a subnet mask.
A sourceAddr is valid if:

(sender address & MyAddrMask) == (MyTAddr & MyAddrMask)

Example

The user “guest” should have access to all objects via password authentication.

In SNMPv3 security:

usmUserEntry localSNMPID guest usmNoAuthProtocol **usmNoPrivProtocol** \
nonVolatile - -
vacmSecurityToGroupEntry usm guest READ **nonVolatile**
vacmAccessEntry READ - **usm authNoPriv exact** All - **- nonVolatile**
vacmViewTreeFamilyEntry All dod **- included nonVolatile**

Figure 7 on the following page outlines the procedure when creating entries for the security configuration.

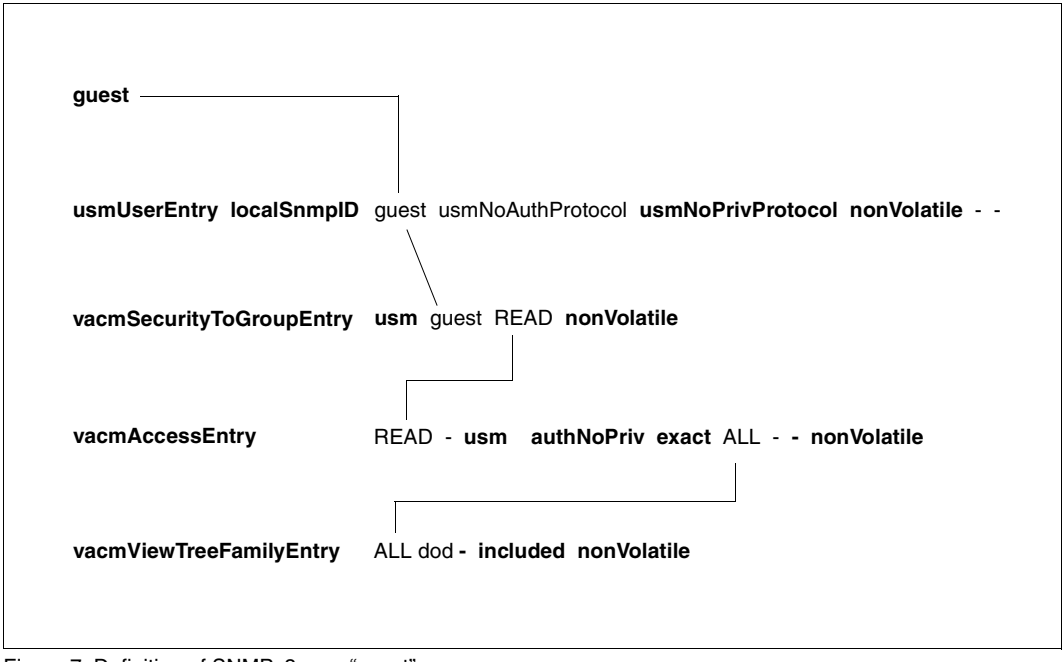


Figure 7: Definition of SNMPv3 user “guest”

3.3.1.7 Definition of access to an agent via HTTP requests

a) Definition of a DR-Web user ID - *httpUserNameEntry*

httpUserNameEntry assigns a security group and a password to a security group.

Tag:	**httpUserNameEntry**
Value:	*MyUserName MyGroup* **- nonVolatile** *MyPaßwort*

*MyUserName**
User ID (String 1..32), under which an HTTP request may be made.

MyGroup
Assigned security group (see b2). The security group defines the scope of authorization.

MyPassword
Password: "-" means that no password is required for the defined user ID.

b) Definition of the access check

b1) Definition of the MIB branch - *vacmViewTreeFamilyEntry*

The definition of the MIB branch consists of one or more *vacmViewTreeFamilyEntry* entries. Each *vacmViewTreeFamilyEntry* assigns an OID to the MIB branch or excludes an OID.

Tag:	**vacmViewTreeFamilyEntry**
Value:	*MyMIB MyOID* **-** (**included** | **excluded**) **nonVolatile**

*MyTrap**
Name of the MIB branch entry (String 32)

*MyOID**
OID or symbolic name of the MIB branch (OID)

included | **excluded**
The MIB branch is to be included or excluded. Only objects that result from all *included* or *excluded* operations remain in the MIB branch.

b2) Definition of the security group - *vacmAccessEntry*

vacmAccessEntry defines a security group and assigns it MIB branches for read and write access.

Tag:	**vacmAccessEntry**
Value:	*MyGroup* **- http authNoPriv exact** [*MYRead*] [*MyWrite*] **- nonVolatile**

*MyGroup**
: Name of the security group (String 1..32). The security group defines the scope of authorization.

MyRead
: Assigned MIB branch for read access (see b1)

MyWrite
: Assigned MIB branch for write access (see b1)

b3) Definition of the security entry - *vacmSecurityToGroupEntry*

vacmSecurityToGroupEntry assigns a security group to the user ID.

Tag:	**vacmSecurityToGroupEntry**
Value:	**http** *MyUserName MyGroup* **nonVolatile**

*MyUserName**
: User ID (String 1..32) with which an http request may be made.

MyGroup
: Assigned security group (see b2). The security group defines the scope of authorization.

Example

The user “guest” is to be granted access to all MIB objects without password authentication.

In DR-Web:

httpUserNameEntry guest READ **- nonVolatile** -
vacmSecurityToGroupEntry http guest READ **nonVolatile**
vacmAccessEntry READ **- http authNoPriv exact** All - **- nonVolatile**
vacmViewTreeFamilyEntry All dod **- included nonVolatile**

The following schematic outlines the procedure for creating the entries in the security configuration.

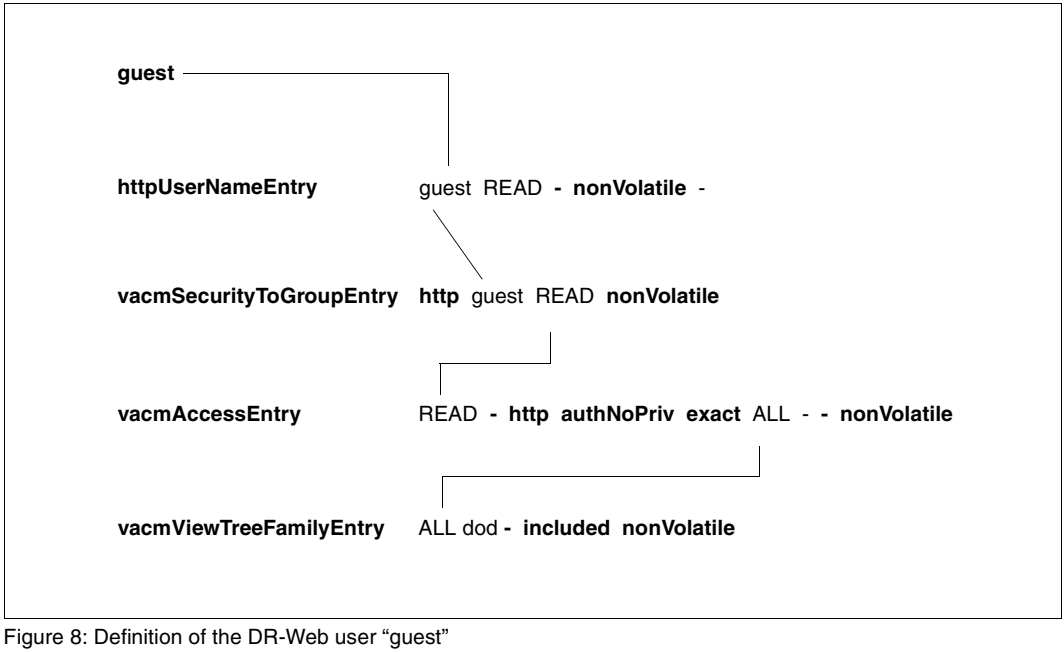


Figure 8: Definition of the DR-Web user “guest”

3.3.1.8 Definition of the trap targets

a) Definition of notify entry and target address

a1) Definition of the notify entry - *snmpNotifyEntry*

Tag:	snmpNotifyEntry
Value:	*MyNotify MyTransTag* **trap nonVolatile**

*MyNotify**
Name of the notify entry (String 1..32)

MyTransTag
Assigned transport label. The transport label points to a target list (see a2) and defines the systems to which the traps are to be sent.

a2) Definition of the target address - *snmpTargetAddrEntry*

snmpTargetAddrEntry assigns the notify entry (see a1) a target address and a target parameter

Tag:	snmpTargetAddrEntry
Value:	*MyTarget* **snmpUDPDomain** *MyTAddr* **300 0** *MyTagList MyTargetParam* **nonVolatile** *MyAddrMask*

*MyTarget**
Name of the target entry (String 1..32)

MyTAddr
Internet address of the target in the form **xxx.xxx.xxx.xxx:p**;
xxx.xxx.xxx.xxx denotes the IP address;
p denotes the port (p=0: default value 162 for traps)

MyTagList
List of tags (see a). The list must be enclosed in quotes (“...”); the individual list entries must be delimited by *one* blank.

MyTargetParam
Assigned parameter entry (see a3)

MyAddrMask
Mask of the form **xxx.xxx.xxx.xxx:0** (analogous to a subnet).

A target address is valid if:
(target address & MyAddrMask) == (MyTAddr & MyAddrMask)

a3) Definition of the Target-Parameter - *snmpTargetParamsEntry*

A target parameter is defined for each *snmpTargetParamsEntry*.

Tag:	snmpTargetParamsEntry
Value:	*MyTargetParam* **0 snmpV1** *MyCommunity* **noAuthNoPriv nonVolatile**

*MyTargetParam**
Name of the target parameter entry (String 1..32)

MyCommunity
Community string with which the trap is to be sent. The community string also defines the security entry (see b3).

b) Definition of the access check

b1) Definition of the MIB branch: *vacmViewTreeFamilyEntry*

The definition of the MIB branch consists of one or more *vacmViewTreeFamilyEntry* entries. Each *vacmViewTreeFamilyEntry** assigns an OID to the MIB branch or excludes an OID.

Tag:	**vacmViewTreeFamilyEntry**
Value:	*MyTrap MyOID* **-** (**included** | **excluded**) **nonVolatile**

*MyTrap**
Name of the MIB branch entry (String 1..32)

*MyOID**
OID or symbolic name of the MIB branch (OID)

included | excluded
The MIB branch is to be included / excluded. Only objects that remain as the result of all included /excluded operations are compared with the Enterprise and the variable bindings of the trap. If the MIB contains Enterprises and variable binding, the trap is sent.

b2) Definition of the security group - *vacmAccessEntry*

vacmAccessEntry defines a security group and assigns it MIB branch for read and write access.

Tag:	vacmAccessEntry
Value:	*MyGroup* **- snmpv1 noAuthNoPriv exact - -** MyTrap **nonVolatile**

*MyGroup**
Name of the security group (String 1..32)

MyTrap
MIB branch assigned to the trap (see b1)

b3) Definition of the security entry: *vacmSecurityToGroupEntry*

vacmSecurityToGroupEntry assigns a security group to the community string

Tag:	vacmSecurityToGroupEntry
Value:	**snmpV1** *MyCommunity MyGroup* **nonVolatile**

*MyCommunity**
Community string (String 1-32) with which the trap is to be sent.

MyGroup
Assigned security group (see b2). The security group defines whether or not access is granted.

c) Definition of the sender address of a trap - *snmpNotifySourceEntry*

snmpNotifySourceEntry defines which sender addresses are to be used for which trap targets. Definition of the sender address is optional.

Tag:	snmpNotifySourceEntry
Value:	*MyNotifySource MyTagList MySourceAddr* **nonVolatile**

*MyNotifySource**
Name of the notifySource entry (String 1..32)

MyTagList
List of tags (see a) on page 46). This list must be enclosed in double-quotes ("..."); the individual list elements must be delimited by *one* blank.

MySourceAddr
Desired sender address for the trap in the form **xxx.xxx.xxx.xxx**

Examples

Example 1

Traps should be sent to host 139.22.22.22 with community string "tcom".

– in SNMPv1 security: **trap** tcom 139.22.22.22

– in SNMPv3 security:

snmpNotifyEntry Nf1 TarTag1 **trap nonVolatile**
snmpTargetAddrEntry Ta1 **snmpUDPDomain** 139.22.22.22:0 **300 0**
TarTag1 Tp1 **nonVolatile** 255.255.255.255.0
snmpTargetParamsEntry Tp1 **0 snmpv1** tcom **noAuthNoPriv nonVolatile**
vacmSecurityToGroupEntry snmpv1 tcom TRAP **nonVolatile**
vacmAccessEntry TRAP **- snmpv1 noAuthNoPriv exact - -** All **nonVolatile**
vacmViewTreeFamiliyEntry All dod **- included nonVolatile**

Example 2

Traps are to be sent to system with the IP addresses 139.22.104.0 to 139.22.111.255 with the community string “tcom”.

– in SNMPv1 security (earlier versions): not settable

– in SNMPv3 security (current version):

snmpNotifyEntry Nf2 TarTag2 **trap nonVolatile**
snmpTargetAddrEntry Ta2 **snmpUDPDomain** 139.22.104.0:0 **300 0**
TarTag2 Tp1 **nonVolatile** 255.255.248.0:0
snmpTargetParamsEntry Tp1 **0 snmpv1** tcom **noAuthNoPriv nonVolatile**
vacmSecurityToGroupEntry snmpv1 tcom TRAP **nonVolatile**
vacmAccessEntry TRAP **- snmpv1 noAuthNoPriv exact - -** All **nonVolatile**
vacmViewTreeFamiliyEntry All dod **- included nonVolatile**

Example 3

Traps with Enterprises sni.2.34 are to be sent to the system with IP addresses 139.22.22.22 with the community string “tent”. The traps with the specific trap number 33 should not be considered.

– in SNMPv1 security (earlier versions): not settable

– in SNMPv3 security (current version):

snmpNotifyEntry Nf3 TarTag3 **trap nonVolatile**
snmpTargetAddrEntry Ta3 **snmpUDPDomain** 139.22.22.22:0 **300 0**
TarTag3 Tp3 **nonVolatile** 255.255.255.255:0
snmpTargetParamsEntry Tp3 **0 snmpv1** tent **noAuthNoPriv nonVolatile**
vacmSecurityToGroupEntry snmpv1 tent EpAcc **nonVolatile**
vacmAccessEntry EpAcc **- snmpv1 noAuthNoPriv exact - -** EpTreeT **nonVolatile**
vacmViewTreeFamiliyEntry EpTreeT sni.2.34 **- included nonVolatile**
vacmViewTreeFamiliyEntry EpTreeT sni.2.34.0.33 **- excluded nonVolatile**

Figure 9 on the following page outlines the procedure when creating entries for the security configuration.

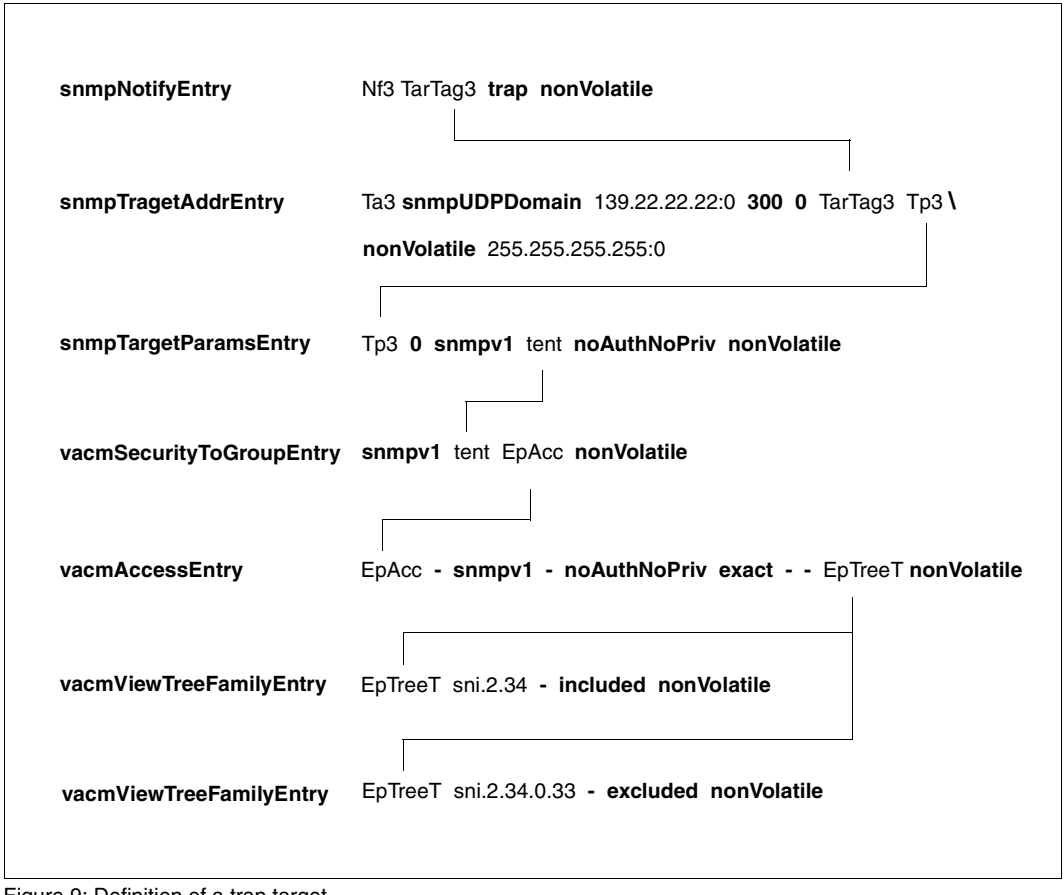


Figure 9: Definition of a trap target

Example 4

Traps to host 139.22.22.22 should be sent with the sender address 112.1.1.1.

– in SNMPv1 security (previous versions): trap public 139.22.22.22 112.1.1.1

– in SNMPv3 security (current version):

snmpNotifyEntry Nf1 TarTag1 **trap nonVolatile**
snmpTargetAddrEntry Ta1 **snmpUDPDomain** 139.22.22.22:0 **300 0**
TarTag1 Tp1 **nonVolatile** 255.255.255.255.0
snmpTargetParamsEntry Tp1 **0 snmpv1** tcom**noAuthNoPriv nonVolatile**
snmpNotifySourceEntry NfS1 TarTag1 112.1.1.1 **nonVolatile**
vacmSecurityToGroupEntry snmpv1 tcom TRAP **nonVolatile**
vacmAccessEntry TRAP **- snmpv1 noAuthNoPriv exact - -** All **nonVolatile**
vacmViewTreeFamiliyEntry All dod **- included nonVolatile**

3.3.1.9 Example

```
# User configuration
communityEntry  localSnmpID public   READ    localSnmpID - -       nonVolatile
communityEntry  localSnmpID master   WRITE   localSnmpID - TarTag1 nonVolatile
communityEntry  localSnmpID multi    WRITE   localSnmpID - TarTag2 nonVolatile
communityEntry  localSnmpID sysread  SysAccR localSnmpID - -       nonVolatile
communityEntry  localSnmpID syswrite SysAccW localSnmpID - -       nonVolatile
usmUserEntry localSnmpID gast usmNoAuthProtocol usmNoPrivProtocol nonVolatile - -
httpUserNameEntry  gast READ - nonVolatile -


snmpNotifyEntry  Nf1 TarTag1 trap nonVolatile
snmpNotifyEntry  Nf2 TarTag2 trap nonVolatile
snmpNotifyEntry  Nf3 TarTag3 trap nonVolatile


# Configuration of the access check
vacmViewTreeFamilyEntry  All      dod           - included nonVolatile
vacmViewTreeFamilyEntry  SysTreeR system        - included nonVolatile
vacmViewTreeFamilyEntry  SysTreeW system        - included nonVolatile
vacmViewTreeFamilyEntry  SysTreeW sysName       - excluded nonVolatile
vacmViewTreeFamilyEntry  EpTreeT  sni.2.34      - included nonVolatile
vacmViewTreeFamilyEntry  EpTreeT  sni.2.34.0.33 - excluded nonVolatile

vacmAccessEntry  READ    - snmpv1 noAuthNoPriv exact All      -        -
nonVolatile
vacmAccessEntry  WRITE   - snmpv1 noAuthNoPriv exact All      All      -
nonVolatile
vacmAccessEntry  SysAccR - snmpv1 noAuthNoPriv exact SysTreeR -        -
nonVolatile
vacmAccessEntry  SysAccW - snmpv1 noAuthNoPriv exact SysTreeR SysTreeW -
nonVolatile
vacmAccessEntry  TRAP    - snmpv1 noAuthNoPriv exact -        -        All
nonVolatile
vacmAccessEntry  EpAcc   - snmpv1 noAuthNoPriv exact -        -        EpTreeT
nonVolatile
vacmAccessEntry  READ    - usm    noAuthNoPriv exact All      -        -
nonVolatile
vacmAccessEntry  READ    - http AuthNoPriv exact All      -        - nonVolatile

vacmSecurityToGroupEntry  snmpv1 public   READ    nonVolatile
vacmSecurityToGroupEntry  snmpv1 master   WRITE   nonVolatile
vacmSecurityToGroupEntry  snmpv1 multi    WRITE   nonVolatile
vacmSecurityToGroupEntry  snmpv1 sysread  SysAccR nonVolatile
vacmSecurityToGroupEntry  snmpv1 syswrite SysAccW nonVolatile
vacmSecurityToGroupEntry  snmpv1 tcom     TRAP    nonVolatile
vacmSecurityToGroupEntry  snmpv1 tent     EpAcc   nonVolatile
vacmSecurityToGroupEntry  usm    gast     READ    nonVolatile
vacmSecurityToGroupEntry  http   gast     READ    nonVolatile
```

```
# Configuration of the address check
snmpTargetAddrEntry  Ta1 snmpUDPDomain 139.22.22.22:0 300 0 TarTag1 Tp1
nonVolatile 255.255.255.255:0
snmpTargetAddrEntry  Ta2 snmpUDPDomain 139.22.104.0:0 300 0 TarTag2 Tp1
nonVolatile 255.255.248.0:0
snmpTargetAddrEntry  Ta3 snmpUDPDomain 139.22.22.22:0 300 0 TarTag3 Tp3
nonVolatile 255.255.255.255:0

snmpTargetParamsEntry  Tp1 0 snmpv1 tcom noAuthNoPriv nonVolatile
snmpTargetParamsEntry  Tp3 0 snmpv1 tent noAuthNoPriv


# Sender address
snmpNotifySourceEntry  NfS1 TarTag1 112.1.1.1 nonVolatile
```

3.3.2 Configuring the master agent and supervisor subagent

The security configuration described in the previous section is the central security mechanism which prevents unauthorized access to the management system and the connected hosts.

Besides the parameters for the security configuration, the configuration file */etc/snmp/agt/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	SNI 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 traps are sent 1 : authentication traps are sent
subagent	If the supervisor subagent is to be started, the name of the library must be entered here: [:<catid>:] [$<userid>.]SYSLNK.SBA-BS2.050 or for RISC machines: [:<catid>:] [$<userid>.]SRMLNK.SBA-BS2.050

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

3.3.3 Configuring the Application Monitor subagent

The Application Monitor subagent allows monitoring of

– user applications
– DCAM applications
– BCAM applications
– subsystems
– job variables and
– log files.

In addition, groups of associated statements can be managed as a unit (object).

The type and extent of the application monitoring are controlled individually via a configuration file. The name of the configuration file is notified to the application monitor subagent in the start command. If there are errors in the configuration file, the start procedure is interrupted. If no configuration file is specified, monitoring is restricted to subsystems.

3.3.3.1 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	59
DCAM application	//ADD-DCAM-APPLICATION-RECORD	60
Subsystem	//ADD-SUBSYSTEM-RECORD	61
Log file	//ADD-LOG-FILE-RECORD	62
Job variable	//ADD-JV-RECORD	64
Group of associated applications	//DEFINE-OBJECT	66
Trap format	//DEFINE-TRAP-FORMAT	68

Example:

The following example can also be found in the SINLIB.SBA-BS2.050 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 = Application1  -
//       ,VERSION = V01.0A00  -
//       ,TYPE = *USER  -
//       ,JV-NAME = MJV1  -
//       ,TRAP-CONDITION = A  -
//       ,WEIGHT=5   -
//       ,ACKNOWLEDGE = *YES -
//
//ADD-APPLICATION-RECORD  -
//        APPLICATION-NAME = Application2  -
//       ,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 = File1  -
//       ,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
```

3.3.3.2 Change in 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 section “Trap structure” on page 180 in the chapter “Functions of the BASIC AGENT”). 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 ADD-DCAM-APPLICATION-RECORD statement states the DCAM and user applications to be monitored cyclically. The monitoring interval for DCAM applications is 60 times the timer setting, i.e. 5 minutes by default.

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> , **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

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 section “Trap structure” on page 180 in the chapter “Functions of the BASIC AGENT”).
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 ADD-SUBSYSTEM-RECORD statement defines the subsystems to be monitored. The monitoring interval for DCAM applications is 5 times the timer setting, i.e. 25 seconds by default.

```
//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

Note:
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 section “Trap structure” on page 180 in the chapter “Functions of the BASIC AGENT”). 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-LOG-FILE-RECORD

The ADD-LOG-FILE-RECORD statement defines the log files to be monitored. By default, the Application Monitor subagent sends a trap for each modification to a job variable. However, it is possible to filter the traps/entries.

//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 section “Trap structure” on page 180 in the chapter “Functions of the BASIC AGENT”).
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 for 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 section “Trap structure” on page 180 in the chapter “Functions of the BASIC AGENT”).
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, logfiles, 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... statement.

If the specifications made for an element of the object in the DEFINE-OBJECT for ICON and ACKNOWLEDGE contradict the corresponding specifications in the ADD... statement, the specifications made in the DEFINE-OBJECT statement apply.

//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>

, **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 that belong to this object
Default value: *NONE

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

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

LOG-FILE = *NONE / list-poss(5): <filename_1 .. 54>
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

Example: Monitoring a MAREN systems

A MAREN system consists of the following components:

– Subsystem MAREN
– Control program MARENCP
– Automatic assignment of free tape MARENUCP

In addition, each VSN reserved by the automatic tape assignment function is automatically stored in the job variable TAPE.FILE.MAREN.

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
```

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

3.3.4 Configuring the Console Monitor subagent

The Console Monitor for monitoring the console interface allows BS2000/OSD console messages to be received on the management station and BS2000/OSD console commands to be input. 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:

Create the operator ID <operator-id>.

```
/ADD-USER USER-ID=<operator-id>,                              -
          PROTECTION-ATTRIBUTE=*PAR(LOGON-PASSWORD=<pass>),   -
          ACCOUNT-ATTRIBUTES=*PAR(ACCOUNT=<account-no>)
```

The logon attributes defined here must be specified in the Console Monitor start statement (see START-SNMP-CONSMON on page 132).

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.

3.3.4.1 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 station

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

Positive message filter

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

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

Trap format selection criterion

The trap format is defined in the message filter file:

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

GENERIC
: Only the Application Monitor-specific trap is used.
GENERIC is the default value

TVCC
: Only the TVCC trap format is used.

ALL
: Both trap formats are used.

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 station. The operator roles are specified in the Console Monitor start statement (see START-SNMP-CONSMON on page 132).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 station are stored in the positive message filter file. The statements

– *msgid*
– *QUESTION*
– *TYPIO*

provide three filter options. 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 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 station. 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] [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 station. 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 page 194).

dev If DEVICE is specified, the Console Monitor subagent sends this trap with the DEVICE entry as Community (see page 194).

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 entry of one or more search patterns.

? :	replaces any single character
* :	replaces a character string of any length
[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.

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=/Hallo*>
<TYPIO PATTERN=/?\?*>
```

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

An example of a message filter can be found in the SINLIB.SBA-BS2.050 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 station are stored in the negative message filter file. Questions cannot be suppressed. 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> [<msgid> ... <msgid>]
```

3.3.4.2 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.

3.3.5 Configuring the AVAS subagent

The AVAS subagent receives the information which AVAS system to monitor via the GENPAR file specified during startup. The subagent uses three methods to communicate with the AVAS system:

– The job variables allocated by AVAS are interpreted in order to monitor the central processes. Since these are not monitor job variables, you should start at least some of the important processes with MONJVs and let the Application Monitor monitor these.

– The subagent obtains information on networks and structure elements via the program interface, which in turn is linked to the AVAS processes via inter-task communication.

– To avoid inefficient access, the subagent can be addressed via an AVAS-RZ exit and thus informed about error states.

When the program interface is first called, further modules are loaded by the AVAS interface module. The reload library is automatically determined via IMON.

If AVAS is not installed using IMON, the default path `$TSOS.SYSLNK.AVAS.030` must be specified in the start procedure `START.AVAS` in the `SYSSPR.SSC-BS2.050` library.

The subagent obtains the following entries from the GENPAR generation file:

AVAS system ID	Identification of the AVAS system.
JVPLAMZD	Job variable for monitoring the PLAM-ZD.
JVUPAMZD	Job variable for monitoring the UPAM-ZD.
JVCENTRAL	Job variable for monitoring the central process.
AVAK	Names of the schedules and the job variables for monitoring the schedules.
USER	Name and password of the user as which the subagent is to log on at the program interface.

AVAS system ID and USER

AVAS system ID and USER are interpreted for LOGON via the program interface. The AVAS system ID identifies the AVAS system to be monitored. The login and password are taken from the USER entry. These LOGON data also determine the authorization scope. The ID used by the subagent is either defined by the USER entry SNMPUSER in the GENPAR file, the first USER entered is selected.

Controlling job variables

The job variables JVPLAMZD, JVUPAMZD, JVCENTRAL and the AVAK schedule are used to monitor the AVAS processes. You must ensure that the USERID that calls the agent has read access to the job variable. If necessary, you must enter the controlling job variables in the GENPAR file specifying :<catid>:$<userid> :

```
:<catid>:$<userid>.<jv-name>
```

The system parameter file SYSPAR used by the AVAS system must be generated from the GENPAR file used by the subagent.

The ID used for LOGON should possess the full scope of authorization rights. It must be granted authorization for access via the program interface (READ-AVAS-LIBRARY).

AVAS-RZ exit

If you want to be informed about error states fast, you can connect a special AVAS-SNMP exit to AVAS exit AVEX0001. For this purpose, the SNMPAV module prepared in the SYSLNK.SSC.BS2-050 library must be linked to the exits. The P.LNK.EXIT procedure is contained in the SINLIB.SSC.BS2-050 library; it may be necessary to adapt the procedure. The SNMPAV module must be linked to the AV03EXIT and AV04EXIT modules. If a separate RZ routine has already been called at exit AVEX0001, it is recommended that you always jump to SNMPAV first, since SNMPAV sets the return code of the exit to X'00' (Ok, record logged). The exit should only be linked to those AVAS systems, which are to be monitored.

3.3.6 Configuring the OMNIS subagent

A configuration file containing the name(s) of the ONMIS to be started must be prepared for the OMNIS subagent. OMNIS names must be padded with blanks up to eight characters. The configuration file is assigned the *<filename>* by the start command (see page 146).

Please refer to the OMNIS “Administration and Programming” Manual, Sections “Declaring the SNMP Managements in OMNIS” and “SNMP Monitoring in OMNIS” for a description of the configuration work required in OMNIS.

3.3.7 Configuring the SESAM subagent

The user must provide the SESAM subagent with a configuration file. The configuration file contains information on the databases attached to this computer and the SESAM DBHs with which these databases are processed. Options for the subagent and for starting the SESAM monitor pool (SESMON) are also stored in the configuration file.

The configuration file must be cataloged in BS2000/OSD. The statements are entered in SDF format. The subagent is passed the name of the file when it is started. Syntax errors in the file cause the subagent to abort.

The configuration file can be modified during a session with either a BS2000/OSD command:

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

and from BS2000/OSD V2.0 using the POSIX command:

```
sesamcmd x "readConfig <filename>"
```

The old file is retained if an error occurs.

An example of a configuration file can be found in the SINLIB.SSC-BS2.50 library.

3.3.7.1 Communication between the SESAM subagent and the SESAM/SQL server

The SESAM subagent communicates with the SESAM/SQL server(s) in order to obtain some of the information the subagent provides (parts of the *rdbmsSrvParamTable*, *rdbmsRelTable*, traps).

The connection between the SESAM subagent and the SESAM/SQL server(s) concerned is set up via the SNMP CONFIGURATION-NAME and DBH-NAME options.
The following two cases must be discriminated:

1. The SESAM configuration designated by the SNMP CONFIGURATION-NAME option is a non-distributed configuration, i.e. without SESAM-DCN.

   In this case, the SESAM subagent can only communicate with a single SESAM/SQL-DBH via the program interface. This SESAM/SQL-DBH is then identified via the two SNMP options CONFIGURATION-NAME and DBH-NAME.

   The information obtained via the program interface can then be made available by the SESAM subagent for this SESAM/SQL-DBH only. The SESAM subagent can also provide information for all remaining SESAM/SQL-DBHs which were specified via the ADD-SERVER-RECORD command.

2. The SESAM configuration designated by the SNMP CONFIGURATION-NAME option is a distributed configuration, i.e. with SESAM-DCN.

   In this case, the SESAM subagent can communicate via the program interface with the SESAM/SQL-DBH identified by the two SNMP options and with all SESAM/SQL-DBHs to which an access path is described in the distribution table of this SESAM configuration.

   It is meaningful to list all databases and their access paths in this distribution rule which are specified in this SESAM-SNMP configuration file via the ADD-SERVER-RECORD command and are not linked to the DBH name.

   The following applies, analogous to the above case: the information gained over the program interface can only be made available by the SESAM subagent for the DBH names and those SESAM/SQL-DBHs to which an access path is described in the distribution table of this SESAM configuration. The SESAM subagent can also provide information on all other SESAM/SQL-DBHs specified via ADD-SERVER-RECORD commands.

3.3.7.2 Configuration file statements

The following statements must be entered in the configuration file.

SET-SNMP-OPTIONS

Defines options for the SESAM subagent.

//SET-SNMP-OPTIONS
CACHE-TIME = 120 / <integer 1..9999> , **CHECK-TIME** = 120 / <integer 1..9999> , **CONFIGURATION-NAME** = ***BLANK** / <alphanum-name1..1> , **DBH-NAME** = ***BLANK** / <alphanum-name1..1>

CACHE-TIME=120 / <integer 1..9999>
Time period for which the data in the subagent cache is valid. Entry is in seconds and the default value is 120 seconds.

CHECK-TIME=120 / <integer 1..9999>
Time interval in which the databases are checked by the subagent. The default value is 120 seconds.

CONFIGURATION-NAME=*BLANK / <alphanum-name1..1>
Name of the SESAM configuration in which the SESAM subagent is running.

DBH-NAME=*BLANK / <alphanum-name1..1>
Default DBH for the SESAM subagent.

SET-SESMON-PARAMETERS

Defines the start parameters for SESMON.

//SET-SESMON-PARAMETERS
CRTE-LIBRARY = <filename_1..54> , **MODULE-LIBRARY** = <filename_1..54> , **SYSOUT** = ***PRIMARY** / <filename_1..54> , **SYSLST** = ***PRIMARY** / <filename_1..54> , **REFRESH-TIME** = 120 / <integer 1..999>

CRTE-LIBRARY=<filename 1..54>
Name of the CRTE library to be used.

MODULE-LIBRARY=<filename 1..54>
Name of the module library containing SESMON.

SYSOUT=*PRIMARY / <filename 1..54>
Assignment of SYSOUT. The default setting is *PRIMARY.

SYSLST=*PRIMARY / <filename 1..54>
Assignment of SYSLST. The default setting is *PRIMARY.

REFRESH-TIME=120 / <integer 1..999>
Time interval in seconds after which the data provided by SESMON is updated. The default value is 120 seconds.

ADD-DATA-BASE-RECORD

Names the database to be monitored by the SESAM subagent.

```
//ADD-DATA-BASE-RECORD

LOGICAL-NAME = <filename_1..18-without-all>
, PHYSICAL-NAME = <filename_1..18-without-all>
, USER-ID = <name_1..8>
, CONTACT = <c-string 1..64-with-low>
, SERVER_ID = *NONE / *PARAMETERS(...)
    *PARAMETERS(...)
        CONFIGURATION-NAME = *BLANK / <alphanum-name 1..1>
        , DBH-NAME = *BLANK / <alphanum-name 1..1>
        , VERSION = <product-version>
```

LOGICAL-NAME=<filename_1..18-without-all>
Logical name of the database.

PHYSICAL-NAME=<filename 1..18-without-all>
Physical name of the database.

USER-ID=<name 1..8>
ID under which the database is cataloged.

CONTACT=<c-string 1..64-with-low>
Identification of the contact person responsible for this database.

SERVER_ID=*NONE / *PARAMETERS(...)
ID of the server concerned.

***PARAMETERS(...)**

CONFIGURATION-NAME=*BLANK / <alphanum-name1..1>
Name of the SESAM configuration in which the SESAM subagent is running.

DBH-NAME=*BLANK / <alphanum-name1..1>
Default DBH for the SESAM subagent.

VERSION=<product-version>
Product version of the server concerned

ADD-SERVER-RECORD

Defines the database server to be monitored by the subagent. Up to 10 servers can be monitored per configuration.

```
//ADD-SERVER-RECORD

IDENTIFICATION = *PARAMETERS(...)
    *PARAMETERS(...)
        | CONFIGURATION-NAME = *BLANK / <alphanum-name1..1>
        | , DBH-NAME = *BLANK / <alphanum-name1..1>
        | , VERSION = <product-version>
, PRODUCT-NAME = <c-string 1..64-with-low>
, USER-ID = <name 1..8>
, CONTACT = <c-string 1..64-with-low>
, PASSWORD = <c-string 0..3>
```

IDENTIFICATION = *PARAMETERS(...)
Identification of the server

***PARAMETERS(...)**

CONFIGURATION-NAME=*BLANK / <alphanum-name1..1>
Name of the SESAM configuration, in which the SESAM subagent runs.

DBH-NAME=*BLANK / <alphanum-name1..1>
Default DBH for the SESAM subagent.

VERSION = <product-version>
Product version of the server

PRODUCT-NAME = <c-string 1..64-with-low>
Product name of the servers

USER-ID = <name 1..8>
ID under which the server runs.

CONTACT = <c-string 1..64-with-low>
Identification of the contact person who is responsible for the database server.

PASSWORD = <c-string 0..3>
PAssword for the administration statements.

ADD-SERVER-PARAMETER

Defines the parameters for a database server. Please refer to the manual SESAM/SQL Server Database Operation (“DBH start statements and options”) for a detailed description of the parameters.

```
//ADD-SERVER-PARAMETER

NAME= *ACCOUNTING / *ADMINISTRATOR / *COLUMNS / *CURSOR-BUFFER / *DBH-IDENTIFICATION /
        *LOG-FILE-OPEN / *MSG-OUTPUT / *OLD-TABLE-CATALOG / *REQUEST-CONTROL /
        *RESIDENT-BUFFERS  / *RETRIEVAL-CONTROL /  *SECURITY /  *SERVICE-TASKS /
        *SESSION-LOGGING-ID / *SPACES / *SQL-DATABASE-CATALOG / *SQL-SUPPORT /
        *STACK-POOL / *SUBORDERS / *SYSTEM-DATA-BUFFER / *THREADS /
        *TRANSACTION security / *TRANSFER-CONTAINER / *USER-DATA-BUFFER / *USERS /
        *WORK-CONTAINER / *TALOG-SUPPORT / *WALOG-SUPPORT / *SESWORK-SUPPORT /
        *CURSOR-MEDIA-SUPPORT-1 / *CURSOR-MEDIA-SUPPORT-2 /
        *CURSOR-MEDIA-SUPPORT-3 / *CURSOR-MEDIA-SUPPORT-4 / *CURSOR-MEDIA-SUPPORT-5
, COMMENT = <c-string 1..128-with-low>
```

NAME = *ACCOUNTING / *ADMINISTRATOR / *COLUMNS / *CURSOR-BUFFER /
***DBH-IDENTIFICATION / *LOG-FILE-OPEN / *MSG-OUTPUT /**
***OLD-TABLE-CATALOG / *REQUEST-CONTROL /**
***RESIDENT-BUFFERS / *RETRIEVAL-CONTROL / *SECURITY /**
***SERVICE-TASKS / *SESSION-LOGGING-ID / *SPACES /**
***SQL-DATABASE-CATALOG / *SQL-SUPPORT / *STACK-POOL /**
***SUBORDERS / *SYSTEM-DATA-BUFFER / *THREADS /**
***TRANSACTION security / *TRANSFER-CONTAINER /**
***USER-DATA-BUFFER / *USERS / *WORK-CONTAINER /**
***TALOG-SUPPORT / *WALOG-SUPPORT / *SESWORK-SUPPORT /**
***CURSOR-MEDIA-SUPPORT-1 / *CURSOR-MEDIA-SUPPORT-2 /**
***CURSOR-MEDIA-SUPPORT-3 / *CURSOR-MEDIA-SUPPORT-4 /**
***CURSOR-MEDIA-SUPPORT-5**

Name of the parameter.

COMMENT = <c-string 1..128-with-low>
Description of the parameter.

3.3.8 Configuring the subagent for storage management

You can use the storage management subagent to monitor disks and pubsets.

The configuration is performed using an input file:

– For monitoring the saturation level of individual public volumes (pubsets), the relevant PVS must be specified in the input file of the subagent. This is done using the ADD-PUBSET-RECORD statement.

– To monitor the reconfiguration state of the individual disks, the relevant disks must be specified in the input file of the subagent. This is done using the ADD-DISK-RECORD statement.

– The //REMARK statement can be used to store comments in the configuration file.

– The last statement in the configuration file should be the //END statement. Any statements that appear after the //END statement are ignored.

– A maximum of 10 pubsets or disks can be monitored.

ADD-PUBSET-RECORD - adding a pubset to be monitored

//ADD-PUBSET-RECORD
PUBSET= <cat_id 1..4> **, CHECK**=**SATURATION-LEVEL** **, TRAP-COMMUNITY**= ***STORAGE** / ***PUBSET-NAME** / <c-string 1..64>

PUBSET=<cat_id 1..4>
CAT-ID of the pubset to be monitored.

CHECK=SATURATION-LEVEL
Object to be monitored; currently on possible to specify the SATURATION-LEVEL (Default value).

TRAP-COMMUNITY=*STORAGE / *PUBSET-NAME / <c-string 1..64>
Community string with which the trap is sent.
If *PUBSET is defined, the <cat-id> is used as the Community Name.
If <c-string 1..64> is specified, this string is sent as the Community Name.
Default value: *STORAGE

ADD-DISK-RECORD - adding a disk to be monitored

//ADD-DISK-RECORD
DISK-MN =<alphanum-name 1..4> **, CHECK=RECONFIGURATION-STATE** **, TRAP-COMMUNITY**= ***STORAGE** / ***DISK-MN** / <c-string 1..64>

DISK-MN=<alphanum-name 1 ..4>
Mnemonic name for the device to be monitored

CHECK=RECONFIGURATION-STATE
Object to be monitored; currently on possible to specify the RECONFIGURATION-STATE (Default value).

TRAP-COMMUNITY=*STORAGE / *DISK-MN / <c-string 1..64>
Community string with which the trap is sent.
If *DISK-MN is defined, the name defined for DISK-MN is used as the Community Name.
If <c-string 1..64> is specified, this string is sent as the Community Name.
Default value: *STORAGE

3.3.9 Configuring the *open*UTM subagent (SSA-OUTM-BS2)

The following section describes the activities required for putting the *open*UTM subagent into operation.

3.3.9.1 Preparation

The subagent communicates with a UTM application via UTM-D-SP or UPIC(BS2000) V1.1. UPIC requires a side information file (upicfile) to link the subagent to the UTM application. This file must be named upicfile and be cataloged in BS2000/OSD to conform to UPIC V1.1. There are four parts to the entry in the *upicfile*:

– An identifier; in this case, HD as the identifier for a link between UPIC(BS2000) and UTM(BS2000).

– The symbolic destination name preset to SNMP4UTM for the currently selected UTM application.

– The partner name defined by the MIB *utmMainBCAMAppl* object.

– The transaction code; this entry is required as the subagent specifies the transaction code with the *Set_TP_Name* call.

The file *upicfile* can be edited under BS2000/OSD. The end-of-line character is represented in BS2000/OSD by a semicolon (";") as there is no <newline> character in BS2000/OSD (see example). This means that if an edited line contains a semicolon, UPIC interprets this as the end of the line and puts the remainder on the next line (up to the next semicolon). This also applies to comment lines.

Example

Side Information file

```
*;
*remote partner applications;
*;
*symbolic destination names for (BS2000/OSD) application ZENTRBS2;
;
HDSNMP4UTM ZENTRBS2;
```

The UTM subagent reports to the UPIC communication system with the local name SNMPUPIC. The name SNMP4UTM is defined as the communication partner of the application with the KDCDEF PTERM or TPOOL statement.

Each application to be monitored must be assigned a BCAM application name with the BCAMAPPL statement:

```
BCAMAPPL ZENTRBS2,T-PROT=ISO
```

A BCMAP entry must be made for each partner application in the *upicfile*:

```
/BCMAP FUNCT=DEFINE,SUBFUNCT=GLOBAL,NAME=(OSI,ZENTRBS2),ES=<BS2000/OSD-M.>
       ,PTSEL-I=(8,'ZENTRBS2 '),PTSEL-N=ZENTRBS2  *)
```

The local name of the UPIC program is also defined via BCMAP:

```
/BCMAP FUNCT=DEFINE,SUBFUNCT=LOCAL,APPL=(OSI,SNMPUPIC)
       ,TSEL-I=(8,C'SNMPUPIC‘),TSEL-N=SNMPUPIC *)
```

*) PTSEL-N and TSEL-N are only required if UPIC(BS2000) and UTM(BS2000) are on one computer.

Since the UTM subagent requires the appropriate authorization to issue UTM administration commands, a UTM user ID must be specified in the LTERM statement which is defined with STATUS=ADMIN or PERMIT=ADMIN.

The UTM subagent uses the KDCWADMI subprogram supplied with UTM to monitor the applications of *open*UTM Version ≥ 4.0. The subprogram is assigned the TAC KDCWADMI. In this case, KDCDEF generation must be expanded to include the following two statements:

```
PROGRAM KDCWADMI,COMP=ILCS
TAC KDCWADMI,ADMIN=Y,PROGRAM=KDCWADMI
```

3.3.9.2 Runtime environment

The UPIC program is controlled via job variables in BS2000/OSD.
UPIC evaluates the following job variables:

Job variable	Link name	Meaning
UPICPATH	*UPICPAT	The job variable UPICPATH defines the file directory under which the side information file is stored. If the job variable is set, the current file directory is used for the search. If the subagent is started under POSIX, the job variable UPICPATH must be supplied the value “BS2/$<userid>”, since UPIC would otherwise try to open the *upicfile* in the POSIX file system.
UPICTRACE	*UPICTRA	The job variable UPITRACE controls the trace generation (see “Diagnostic documents” in the next section).
UPICLOG	*UPICLOG	The job variable UPICLOG defines the name of the logging file. If this definition is missing, the name is “##.USR.TMP.UPICL<tsn>”.

Note that the assignment is lost after LOGOFF.

3.3.9.3 Diagnostic documents

Besides the trace file, of the *open*UTM subagent, there are other files which may be helpful in the event of an error:

– UPIC-Trace file
– UPIC-Logging file
– SYSLOG file

UPIC-Trace file

The carries system UPIC enables trace information to be generated for all interface calls. You can control this procedure by setting the job variable *UPICTRACE*. The call *Enable_UTM_UPIC* interprets the contents of the job variable. If the job variable is set, the parameters ad the user data are logged project-specific in a file up to a size of 128 bytes.

Activating the UPIC trace

The UPIC-Trace is activated as follows:

```
/SET-JV-LINK LINK-NAME=*UPICTRA,JV-NAME=UPICTRACE
/MODIFY-JV UPICTRACE,VALUE='-S[X] [-R wrap] [-Dprefix]'
```

Meaning:

-S Detailed logging of call, the associated arguments and user data up to a maximum length of 128 bytes (mandatory data)

-SX Additional internal information at the interface to the transport system are logged.

-R *wrap* The decimal number specified by *wrap* defines the maximum size of the temporary trace file.
Default value: 128.

-D*prefix* The trace files are created with the following names:
– *prefix*.UPICT<tsn>
– *prefix*.UPICU<tsn>
If *prefix* is not specified, “##.USR.TMP” is used as prefix.

Deactivating the UPIC trace

The UPIC-Trace is deactivated using the following two commands:

```
/DELETE-JV UPICTRACE
/MODIFY-JV UPICTRACE,VALUE=''
```

UPIC logging file

If the *open*UTM application terminates a conversation normally, an *open*UTM error message is written to the UPIC logging file. The UPIC logging file is only opened to write the error message (mode *append*) and then closed again.

SYSLOG file

When an application is started, *open*UTM creates an application-specific log file SYSLOG. This file logs events which occur during the application runtime in the form of *open*UTM-messages.

3.4 Integration in the management platforms

For the management platforms below, Fujitsu-Siemens offers an SMAWsmbs2 integration package for Solaris, as well as an SMBS2 integration package both for Reliant UNIX and Windows NT. These packages enable the automatic integration of BS2000/OSD in the following management platforms:

– Unicenter TNG from the company Computer Associates (Windows NT, Solaris)
– TransView SNMP (Reliant Unix)
– TransView Control Center (Symmetrix monitoring) (Reliant UNIX)
– OpenView NetWork Node Manager from HP (Reliant UNIX)

The integration package SMBS2 or SMAWsmbs2 is not required for implementing the master agent and its subagents.

Besides the integration packages, tailored management applications are also offered for individual subagents.

The integration package and management applications are part of the BS2-SNMP-SO, BS2-SNMP-SX and BS2-SNMP-NT products, and are included on the CD supplied.

The CD contains the following packages:

Package	**Meaning**
SMBS2 / SMAWsmbs2 from BS2-SNMP-SX or BS2-SNMP-NT or BS2-SNMP-SO	SNMP management for BS2000/OSD; package for automatic integration in system management platforms
HNC-SNMP-NT	HNC device management for Windows NT (see “SNMP for HNC” manual)

This section describes the installation and configuration of the SMBS2 or SMAWsmbs2 integration package on the management platforms stated.

Installation and configuration of the management applications CMBS2 or SMAWcmbs2 and PMBS2 or SMAWpmbs2 are described in the section “Installing the management applications” (see page 120).

3.4.1 Integration in CA Unicenter TNG under Windows NT

Unicenter TNG Version 2.2 must be installed and available in order to use SMBS2 on Windows NT. If no Unicenter TNG installation can be found, the installation is aborted. If only the framework of Unicenter TNG is installed, SMBS2 can only be used with restrictions.

In addition to the application-specific MIBs, the SMBS2 package contains the following elements for supplementing the components “World View”, “Enterprise Management” and “Agent Technology” of Unicenter TNG:

- World View

  The host class “SiemensBS2000” and additional object instances of the classes “Method”, “Popup-Menu”, “Icon_2d” and “Icon3d” are created to be able to place BS2000/OSD systems in the network map either manually or with the aid of the automatic Discovery function. The BS2000/OSD MIBs are provided for the “Object View”.

- Enterprise Management

  To improve the monitoring of BS2000/OSD systems at the event console, SMBS2 contains message formats for all traps of the BS2000/OSD-SNMP master agent and its subagents. Actions are linked with message output in the message formats in order to clarify the traps; important events are highlighted.

- Agent Technology

  The host class “Siemens-BS2000” is defined. This class is assigned the following agent classes with DSM policies:

  – The agent classes “Ping” and “MIB2” supplied with Unicenter TNG.

  – New agent classes for monitoring the various subagents in BS2000/OSD: “sieAppMonitor”, “sieAVAS”, “sieHSMS”, “sieOmnis”, “sieRDBMS”, “sieStorage”, “sieSupervisor”.

3.4.1.1 Installing on Unicenter TNG under Windows NT

Since SMBS2 requires the installation of Unicenter TNG, the installation procedure checks whether Unicenter TNG is already installed. Any warning issued does not necessarily mean that the installation conditions are not satisfied.

Installation levels

There are three installation levels of SMBS2:

– "Basic Support" can be used with the installation of the framework of Unicenter TNG. This framework support covers all elements of SMBS2 that supplement the "World View" and the "Enterprise-Management".

– "Full Support" can be used with a full version of Unicenter TNG. It contains additional elements for use of the "Agent Technology".

– "Remote Administration Client" can be used with an installation of the Remote Administration Client of Unicenter TNG. "RemoteAdministrationClient" contains all files for supplementing the user interface.

Files for the installation

SMBS2 creates files in the following subdirectories of the Unicenter TNG installation directory:

– *Config\Abrowser*
– *Icons*
– *Images*
– *Models*
– *Schema\Included*
– *Services\Config\AWS_nsm\Dm*
– *Services\Config\AWS_WVGATE*
– *Services\Config\Mibs*

SMBS2 creates files with scripts for adapting the configuration of Unicenter TNG. These script files are stored in a separate directory to be specified by you during the installation process. The directory C:\SMBS2 is selected by default.

The following files are created:

- *BS2000.tng* file

  Contents:

  – definition of the “SiemensBS2000” object class as a subclass of “Host”

  – a range of additional instances of other object classes required in the Common Object Repository, e.g. the BS2000/OSD icons

  These definitions can be imported into the repository using the following command:

  ```
  TRIX -Q -R=<repository> -U=<SQL-Admin> -W=<password> -G -X <pathname>\BS2000.tng
  ```

- *BS2dbscript.txt* file

  The *Bs2dbscript.txt* file contains the definitions of the new message formats, as well as the definitions of the actions linked to these messages for the “Enterprise Management” event console.

  The message formats and the actions linked to these formats can be loaded using the following command:

  ```
  cautil -F <pathname>\Bs2dbscript.txt
  ```

- Two readme files with the information contained in this section, plus a licensing text.

Additional subdirectories “INST” and “BACKUP” in the installation directory

Two additional subdirectories, “INST” and “BACKUP”, are created in the Unicenter TNG installation directory during the course of the installation:

- The “INST” directory contains the installation log as well as the deinstallation programs *uninstall.exe* and *remove.exe*. The latter should not be called directly, as the functions of SMBS2 can only be used again after a reinstallation.
- The “BACKUP” directory is used to store copies of all files whose contents have changed during the installation.

Further installation steps

After the files have been created, you determine the scope of the subsequent installation steps. You can skip one or more of the following steps by deleting the corresponding marking(s) in the installation dialog window:

1. Import object classes into the repository
2. Activate the message texts and the actions linked to the messages.
3. Modify the DSM configuration (with full support only)
4. ResetDSM (with full support only)

The desired installation level determines which of these steps can be performed:

– Steps 1) and 2) must be performed for a Basic Support installation.

– Steps 1) through 4) must be performed for a Full Support installation.

– There is no postprocessing for a Remote Administration Client installation.

Each individual step is performed independently of the other steps.

It is only practical to disable individual installation steps if you are reinstalling SMBS2 at a later stage, e.g. to recover lost files.

For a first-time installation, all installation steps (with the possible exception of ResetDSM) must be performed. If individual steps are omitted, SMBS2 will not be fully functional. The omitted steps must then be performed manually or by reinstalling the entire package. Performing the installation steps manually at a later point in time is a difficult operation that may lead to inconsistencies.

3.4.1.2 Configuring Unicenter TNG

After installing SMBS2, you can add BS2000/OSD systems to the network map of the “World View”. The Unicenter TNG documentation describes how to incorporate icons into the network map.

Integration of BS2000/OSD systems

Use the following commands to integrate individual BS2000/OSD systems using automatic discovery:

```
dscvrone –n <name of BS2000 system>
```
or

```
dscvrone –i <IP address of BS2000 system>
```
or

```
dscvrbe –r <name of repository> –7 <IP address or name of BS2000 system>
```

Displaying message texts on the event console

To improve the assignment of messages on the event console, it is advisable to include the “Facility” column in the display.

Configuring and starting the agent technology

Following the installation, the DSM policies are not always active for all subnets. It may be necessary to call the DSM Wizard in order to integrate the subnets into the monitoring facility.

The pop-up menu of the icons for the object class “SiemensBS2000” contains three entries at the end for calling the management applications for the subagents Console Monitor, Performance Monitor and BCAM-Monitor. To function properly, these menu commands require installation of the products CMBS2, PMBS2 or BMBS2. If you install these products at a later stage, it is advisable to reinstall SMBS2 as otherwise the menu commands cannot be used.

Observe the following when carrying out installation:

– If the Unicenter TNG services are stopped during the installation phase, the event console must be reconfigured again using the *opreload* command after completion of all installation steps in order to activate the message formats and all actions linked to these formats.

– The DSM policies are only activated if the command *resetdsm* has been executed. The *resetdsm* command is executed as the last installation step, provided that it has not been deactivated by the user. After execution of the command *resetdsm*, the “Agent Technology” must be restarted with the command *awservices start*.

– If BS2000/OSD-MIBs are not available in the MIB browser of the “Agent Technology”, they can be loaded later with the batch processing program *install_siemibs.bat*.

Displaying the agent objects in the map together with the subobjects

The icons of ten different agent classes assigned to the “SiemensBS2000” class can be displayed in the unispace of BS2000/OSD systems. The particular subagents active in BS2000/OSD determine the classes for which objects are created.

The following two agent icons contain areas with further icons:

– The supervisor icon (“sieSupervisor”) contains an area with objects of the “sieBS2000Subagent” class. An object is created for each monitored subagent contained in the supervisor table.

– The Application Monitor icon (“sieAppMonitor”) contains an area with objects from the classes “sieBCAMApplication”, “sieDCAMApplication” and “sieUserApplication”. These objects contain areas into which the objects of the monitored BCAM, DCAM and user applications are placed.

Pollset settings

The most important parameters for communication between the management station and the agent are defined by pollsets in Unicenter TNG. Communities for reading and setting object values are defined in the pollset, amongst other things. These definitions must match the settings on the agent.

SMBS2 contains a pollset with the name “SiemensBS2000”. This is valid for all objects of the “SiemensBS2000” class, as long as the user does not make any changes or add new pollsets that can be used on the BS2000/OSD systems in the network.

Pollsets can be modified or newly defined using the pollset browser. The *aws_orb* service is required in order to use the pollset browser. This service is started with the pollset browser if it is not already active. However, it is not terminated when the pollset browser terminates.

The online help provides more details on using the pollset browser. For more information on pollsets and the pollset browser, please see the Unicenter TNG documentation.

Carry out the following steps to modify or define a new pollset:

1. Call the pollset browser:

   The pollset browser is activated via the following menu command:

   `Start -> Programs -> Unicenter TNG -> Agent Technology-> Pollset Browser`

2. Select the pollset:

   The pollset browser lists all the pollsets defined in the repository.
   Two buttons are displayed beside the list of pollsets:

   – Click the “D” button to delete the pollset.

   – Click the “C” button to change the pollset. The contents of the pollset definition are transferred to the top line and two buttons labeled “yes” and “no” are displayed on the left-hand side.

3. Define the pollset

   In the top line you can enter the contents of a new pollset or of a pollset you want to change:

   – In the case of a new pollset, enter the name in the first box (name field).

   – If you want to change an existing pollset, the name field already contains the name of this pollset and cannot be edited.

   Enter the information on the pollset application area in the “Agent” and “Host” columns:

   – In the “Host” column, specify any of the following values:

     `<hostname>` or `<IP address>` or `<subnet mask>` or *

   – In the “Agent” column, specify:

     `<name of an agent object instance>` or
     `<name of an agent object class>` or *

4. Confirm the pollset definition

   When you click the “yes” button, the modified pollset is transferred to the repository. Click the “+” button for a new pollset.

5. Test the pollset

   The test function of the pollset browser is activated by clicking the “magnifying glass” button in the toolbar or via the “View” menu. A form then appears in which you can enter information on an object in the map. Click “Go” to display the list of pollsets that can be applied to the object. The valid pollset appears at the top of the list.

3.4.1.3 Deinstallation

During deinstallation, the package files are deleted and an attempt made to undo the changes made to the Unicenter TNG configuration, if possible. The object class “SiemensBS2000” added to the repository cannot be removed automatically if objects still exist for this class. The other WorldView classes and objects created during the installation are deleted in the deinstallation process.

After deinstallation, the user can perform the following step manually:

1. Delete all objects of the “SiemensBS2000” class in the network map, including their subobjects. It is advisable to use the Object Browser to obtain an overview of all objects in the “Siemens2000” class and to delete the objects together with their subobjects.
2. The “Siemens2000” object class must be deleted in the class wizard.

3.4.2 Integration in CA Unicenter TNG under Solaris

Unicenter TNG Version 2.2 must be installed and available in order to use SMAWsmbs2 on Solaris. The installation procedure of SMAWsmbs2 checks whether or not and the extent to which Unicenter TNG is installed. If only partial components of Unicenter TNG are installed, SMAWsmbs2 can only be installed to a limited extent. The installation dialog shows which Unicenter TNG components relevant to SMAWsmbs2 are installed on the system. Installation can then be performed for the components displayed. The scope of the SMAWsmbs2 installation can be restricted further by selecting specific components.

- TNG Base Managers - WorldView Components and EM Java GUIs

  The “SiemensBS2000” host class and additional object instances of the classes “Method”, “Popup-Menu”, “Icon_2d” and “Icon_3d” are created so that BS2000/OSD systems can be placed in the network either manually or using the automatic discovery function. The BS2000/OSD MIBs are provided for the object view.

- TNG Base Managers - Enterprise Management

  To improve the monitoring of BS2000/OSD systems on the event console, SMAWsmbs2 contains message formats for all traps of the BS2000/OSD SNMP master agent and its subagents. The message formats are linked with actions that clarify the meaning of the traps and highlight important events.

- TNG Agent Technology Managers and Agents

  The “SiemensBS2000” host class is defined. The following agent classes with DSM policies are assigned to this host class:

  – The agent classes “Ping” and “MIB2” supplied with Unicenter TNG.

  – New agent classes set up by SNMP to monitor the various subagents in BS2000/OSD: “sieAppMonitor”, “sieAVAS”, “sieHSMS”, “sieOmnis”, “sieRDBMS”, “sieStorage”, “sieSupervisor”.

3.4.2.1 Installation on Unicenter TNG under Solaris

During the installation process, the services of Unicenter TNG must be active so that entries can be made in the database. The services are started by the installation procedures, but are not stopped when the installation has concluded. You can use the *unicntrl stop all* command to stop the services “by hand”.

Files for the installation

SMAWsmbs2 creates files in the following subdirectories of the Unicenter TNG installation directory:

– *browser/images/wvicons*
– *schema/included*
– *atech/services/config/aws_nsm/dm*
– *atech/services/config/aws_wvgate*
– *atech/services/config/mibs*

A subdirectory called *SMAWsmbs2* is created by default in the *opt/SMAW* directory. *SMAWsmbs2* contains the subdirectories *scripts*, *docs, bin, include*.

The *skripts* directory contains the following files:

- Files *Bs2000.tng*, *Bs2000del.tng* and *Bs2000Agtdel.tng*

  Contents of the *BS2000.tng* file:

  – Definition of the new *SiemensBS2000* object class as a subclass of *Host*

  – Definitions of objects for the “World View” repository: BS2000 icon and pop-up menu for the BS2000 icon.

  The specified definitions are imported to the repository using the following command:

  ```
  trix –f <pathname>/Bs2000.tng
  ```

  The files *BS2000del.tng* and *Bs2000Agtdel.tng* are needed for deinstallation.

- Files *BS2dbscript.txt* and *BS2dbscriptdel.txt*

  Contents:

  Definitions of the new message formats (message records), as well as the definitions of the actions linked to these messages (message actions) for the event console of Enterprise Management.

  The message formats and the actions linked to these formats can be loaded using the following command:

  ```
  cautil –f <pathname>/Bs2dbscript.txt
  ```

  The *DB2dbscriptdel.txt* file is needed for deinstallation.

- File *Bs2TrapAnalyse.c*

  Contents:

  Source code of the *Bs2TrapAnalyse* object file (see below)

  Contents:

  Compilation procedure for *BS2TrapAnalyse.c*

- Subdirectory *include*

  Contents:

  Header files *Bs2Msg.h* and *Bs2VarBindOid.h*, needed for compiling *BS2TrapAnalyse*.

The *bin* directory contains the following file:

- File *Bs2TrapAnalyse*

  The *Bs2TrapAnalyse* program is called by the newly defined message actions in order to create and output message texts.

The *docs* directory contains four readme files with the information provided in this section, plus a licensing text.

Further installation steps

After the files have been created, the subsequent installation steps that must be performed are displayed in the course of the installation dialog.

The maximum installation of SMAWsmbs2 comprises the following steps:

1. Importing the object classes to the repository (World View)
2. Activating the message texts and the actions linked to the messages (Enterprise Management)
3. Modifying the DSM configuration (Agent Technology)
4. ResetDSM and importing the agent classes (Agent Technology)

Each individual step is implemented independently of the other steps. It is thus only practical to omit individual installation steps if you are reinstalling SMAWsmbs2 at a later stage, e.g. to restore lost files.

In the case of a first-time installation, you should certainly perform all the installation steps offered for selection. If individual steps are omitted, SMAWsmbs2 will not be fully functional. The omitted steps must subsequently be performed manually or by reinstalling the entire package. Performing the installation steps manually at a later stage is difficult and may lead to inconsistencies.

3.4.2.2 Configuring Unicenter TNG

Two different user interfaces are available for working with Unicenter TNG on Solaris:

– A web interface implemented by Java applets that can be accessed using a web browser

– an interface offered on a Windows NT system via a remote administration client

  In this case, SMBS2 must be installed on the remote administration client (installation mode “administration client”. A detailed description of installing and implementing SMBS2 in conjunction with Unicenter TNG on Windows NT can be found in the section “Integration in CA Unicenter TNG under Windows NT” on page 92 ff.

3.4.2.3 Configuring the trap distributor

If other applications that are to receive traps are started in addition to Unicenter TNG, you can configure the *catrapmux* trap distributor such that it also distributes traps to other ports. To this end, append a line of the following form to the end of the *$CAIGLBL0000/snmp/config/catrapmux.conf* file:

```
<application-name>:<port-number>
```

The trap distributor *catrapmux* must then be stopped using the *unicntrl snmp stop* command and restarted using the *unicntrl snmp start* command.

3.4.2.4 Deinstallation

When a deinstallation is performed, the package files are deleted and the changes made to the configuration of Unicenter TNG are revoked as far as possible. The SiemensBS2000 object class inserted in the repository cannot be removed if objects for this class still exist.

The user can perform the missing steps manually following the deinstallation:

1. All objects of the SiemensBS2000 class in the network map must be deleted, including subobjects. It is advisable to obtain an overview of all the objects in the SiemensBS2000 class using the Object Browser, and to delete the objects together with their subobjects
2. The SiemensBS2000 object class must be deleted with the Class Wizard.

3.4.3 Integration in TransView SNMP

In addition to the following application-specific MIBs, the SMBS2 package contains the following components for use on TransView SNMP:

– Object view which describe the supplementary menus and tables required to display the values of the MIB objects. Only objects that can have several entities are displayed in the tables.

– Image files for displaying scalable objects offer an easier way to list objects and values in a form.

– The *.map_SMBS2* network map file, which can be used to integrated network maps with applications to be monitored into an existing network configuration.

– Bitmap files with icons for displaying the SNMP applications in BS2000/OSD.

– Two files with help texts:

  These files are text files that mainly contain textual descriptions of the MIBs supported and a file with keywords to be added to the help index. A list of explanations for all objects of the MIB branch, to which the objects shown in the window belong, is provided as help text for each window.

– Procedures to support installation and deinstallation, which are not included in TransView SNMP.

Requirements

SMBS2 is not required in order to use the master agent and its subagents in BS2000/OSD. If you want to use SMBS2, you will require TransView SNMP ≥ V4.0 and optionally TransView Control Center ≥ V4.0. TV SNMP can be started during the installation procedure; however, the SMBS2 installation does not become effective until you restart SNMP. If a predecessor version of SMBS2 is installed, you should deinstall it prior to installing Version 3.0 of SMBS2.

Files with the same names that are overwritten during installation are saved under the name *<file-name>.usr*.

In order to be able to install SMBS2, *perl* must be available. If it is not available, the installation is aborted. Important functions for post-editing the TransView SNMP configuration files are carried out using *perl* scripts. Since TransView SNMP is also requires *perl* for its installation procedure, you should make sure that this requirement is satisfied.

3.4.3.1 Installation on TransView SNMP

At the start of installation SNMP, all platforms on which the SMBS2 package have been installed are listed. Versions as of Version 3.0A are taken into account, older versions are not considered. Then, the installation procedure searches for the installed platforms suitable for SMBS2 and displays them in a list that indicates the version and the basic path name. You can now select a platform for installation form this list.

It is possible to install SMBS2 on all of the platforms displayed one after the other. Whereas SMBS2 sets up only one platform for each installation, SMBS2 is simultaneously removed from all platforms during deinstallation. The platforms on which SMBS2 is installed are displayed at the start of deinstallation. The same restriction applies here as for the display during installation: versions that are older than 3.0A are not included. A deinstallation is carried out for all platforms listed. On platforms with older versions, only the files installed with the SMBS2 package are deleted. This can cause malfunctions in the management platforms; however, these can be corrected by reinstalling SMBS2 on the platforms affected (please refer to the Release Notes for the relevant predecessor versions of SMBS2).

The installation of SMBS2 is carried out in the *home* directory of TV SNMP:

– the MIBs in the *asn1* subdirectory

– the files for the object views in the *views* subdirectory

– the image files for displaying scalable objects, icons and the help text files in the *lib* subdirectory

– the network map file *.map_SMBS2* in the *maps* subdirectory

TransView-SNMP-MIB

The central TransView-SNMP-MIB contains the definitions of all the devices that can be managed. It is formed using the *parse* function from the manufacturer-specific MIBs of the devices and applications to be monitored by TransView SNMP.

During the installation procedure you can specify whether the newly installed ASN.1 files are to be added to the central MIB in the BS2000/OSD system management, provided that it has been installed in the *home* directory of TransView SNMP. If a file already exists, it is saved as *mib.org*. It is now possible to integrate a new MIB at any time, independently of the installation of SMBS2.

Object view in *object.views*

The specifications in the *object.views* file map the values of the object attributes obtained from the agents to displays for the user. They cover the menu description for calling the window to display the MIB object groups, as well as the descriptions for the tables and diagrams. You can specify whether the new object views installed with SMBS2 are to be added to an existing *object.views* file, provided that TransView SNMP has been installed in the home directory. The *rfc1514-host. obj* file is used for the Host Resources MIB. This file is included in the scope of delivery of TransView SNMP. The original file is saved as *object.views.org*.

Help text files

Except for the Host Resources MIB, SMBS2 includes help text files for all MIBs. These are generated from the description texts for the MIB objects. The help texts for the Host Resources MIB are missing because the object view file contained in TransView SNMP is used.

Additional notes for installation on TransView SNMP V3.1

The domain directories are omitted as of TransView Control Center Version 4.0A. Instead, the domains can be assigned files from the installation subdirectories of TransView SNMP. The domain-specific installation is therefore no longer required in order to make the SNMP extensions available in the domains. If you are installing TransView Version 3.1, you can also use the domain directories for installation is you are using the TransView Control Centers. All existing path names of the form <*home directory*>*/tcc/*/<*domainname*> */maps* are searched to determine the domain names. Please observe that when the installation is carried out in a different directory than the home directory with ID *tvsnmp*, this directory is also searched for domain names. If domain directories are found, the user is given three options:

– installation in none of the domains (standard option)
– installation in all domains found or
– installation in a selection of domains.

The third option outputs a numbered list of the domain names found. The number sequence defines a subset of domains for installation.

The MIB and *object.views* files in a domain directory are only modified if the *lib* subdirectory that contains these files actually exists, i.e. is not just a symbolic link. If symbolic links are used, you must make sure that the files referenced are also adapted.

3.4.3.2 Configuring on TransView SNMP

An existing network configuration can be extended with the network map file *map_SMBS2*, which is written to the *<home-tvsnmp>/maps* directory, to include the BS2000/OSD systems and applications to be monitored. The network map file includes:

– one icon for each BS2000/OSD computer and

– one icon for each subagent to be monitored for AVAS, file transfer, Spool&Print services, storage management, SM2 (performance measurement), *open*UTM, SESAM, BCAM, OMNIS and HIPLEX. The other subagents are represented by BS2000/OSD icons.

– one connection each between the icons of the BS2000/OSD computer and the application icons. These connections are assigned an object entity on the BS2000/OSD side. The assignment is a line for the subagent taken from the subagent table.

– the definition of the attribute groups for the icons and links. In addition to the MIBs, the attribute groups provide the management station with an information base for the agent object that supports a device or an application represented by an icon. At the same time, the attributes define the valid range for alarms and polls. This is done by assigning attribute groups to the connections.

Incorporate the new network map into your network configuration as follows:

1. Start TransView SNMP.
2. Switch to the desired network map.
3. Select *Add* in the *File* menu. Enter the file name *.map_SMBS2* and path where the file is stored (default: *<tvsnmp-home>/maps*). Press *Accept* to confirm the dialog.

   The icon of a BS2000/OSD system is shown in your network map and this may be moved as required. You can repeat this process, thereby entering a number of BS2000/OSD systems into the network map. A double-click on the icon opens a new window with a BS2000/OSD computer and the icons of applications to be monitored. If the TransView Control Center is running, the menu item *Accept* may be blocked with some versions. Please call TransView SNMP directly in this case.

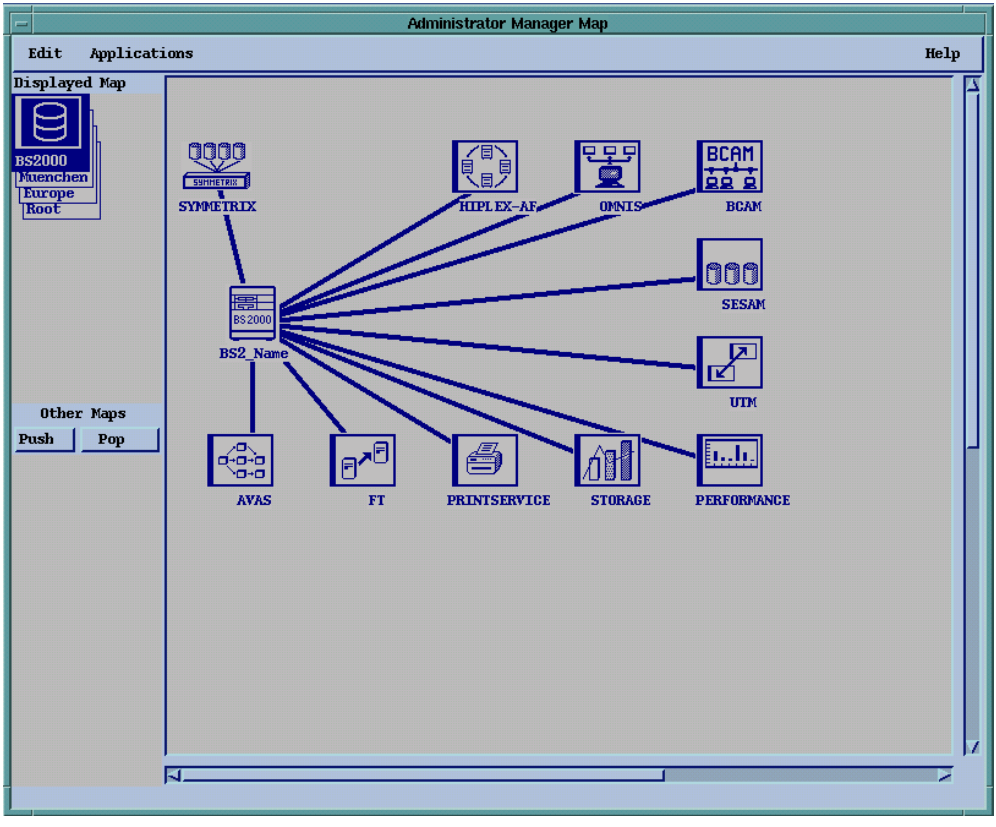


Figure 10: Administrator network card

4. Open the *Object overview* for BS2000/OSD by double-clicking on the BS2000/OSD computer icon. The network map contains the icons for the BS2000/OSD system and the applications.
5. Open the *Object overview* for BS2000/OSD by double-clicking on the icon of the BS2000/OSD host.
6. Input the name and Internet address of the computer into the fields provided, according to your hardware configuration.

7. Correct the community string according to the configuration of the agent system.

   Repeat steps 5 - 7 for all BS2000/OSD applications (subagents).

   The address of the system on which the application is running is to be input as the application Internet address. The agents of the BS2000/OSD applications whose MIB objects have write access require a community string with write access to be able to use the full functionality. It is recommended that different community strings are used for the BS2000/OSD computer and the applications. The community strings of the applications should thereby not be trap receivers (cf. page 55).

   i The community strings of the applications must be the same as the application names in the network map.

8. To activate the alarms, you must set all icons and links to the state *managed*. This value can be changed in the object overview windows.
9. Save the changes with *Save* in the *File* menu.

To simplify this procedure, the *chn_map_SMBS2.pl* script is available in the *bin/TV-SMBS2* subdirectory as of SMBS2 Version 3.0A. The script is called according to the following format:

```
chn_map_SMBS2.pl <dir> <ip-address> <suffix>
```

<*dir*> denotes the absolute or relative path where the original *.map_SMBS2* subnet file resides.

<*ip addr.*> defines the IP address to be entered for all nodes in the subnet map.

<*Suffix*> is added as suffix to the name of the subnet map, the name of the network node and to the line names. Since the community and name must correlate for all nodes except for *BS2_Name*, the community names are also matched here. The first two parameters are mandatory. <*Suffix*> may be omitted, in which case the IP address is used as suffix. The copy generated is assigned the file name: *.map_SMBS2*<*Suffix*> or *.map_SMBS2.*<*ip addr.*>.

Object views

If one of the subagents is not to be run, the icon for its application can be removed from the network map. Select the relevant icon, press the “Menu” mouse button (right button) and choose *Delete* in the pop-up menu. Remove the connection line to the deleted icon in the same way.

3.4.3.3 Deinstallation

When deinstalling, you should restore the state of TransView SNMP and/or TransView Control Center as far as possible before installing SMBS2. At the start of deinstallation, the platforms on which SMBS2 is installed are listed. Versions of SMBS2 that are older than V3.0A are not listed. Deinstallation is carried out only for the platforms listed. If files of older versions installed are deleted, without reconfiguring the platform affected, reinstallation must be carried out to ensure trouble-free operation. If versions of TransView SNMP are affected, it is absolutely necessary to install SMBS2 in order to restore deleted files that are required.

During Deinstallation of SMBS2, you can decide whether the *mib* and *object.views* are to be returned to their original states, i.e. TransView SNMP without SMBS2. If this is not done, the applications of the TransView-SNMP-MIB and the *object.views* must be reset manually. During the deinstallation process, all daemon directories to be processed are searched. Prerequisite for editing the files is the availability of *perl* during the deinstallation process.

The deinstallation routines do not check whether the SMBS2 subnet maps are defined for a BS2000/OSD system in the network map at the time of deinstallation. Removing the icon files makes some of the components in the subnet map invisible and thus prevents them from being managed by the user. They can therefore no longer be deleted until SMBS2 has been reinstalled.

3.4.4 Integration in TransView Control Center

The following components are offered for use on the TransView Control Center:

– the configuration file *bs2symm.def*, which contains the definitions:

  a) of the *BS2-Symmetrix* application

  b) of events of all reference codes for the Symmetrix messages,

  c) of the *Symmetrix* node and

  d) of relations between the *BS2- Symmetrix* application, the defined events and the *Symmetrix* node.

  The event definitions refer to the TV-SNMP alarms. Therefore they can only be displayed in one network map of TransView SNMP.

– The *bs2symm.cnf* file that contains a pattern for four lines to be added to the configuration file of the Console Monitor on the BS2000/OSD system

– A procedure for accepting the data from the *bs2symm.def* configuration file into the domains of the TransView Control Centers.

3.4.4.1 Installing on TransView Control Center

A *SMBS2* subdirectory is created in the installation directory of the TransView Control Centers (generally: */opt/tcc*) to store the files:

bs2symm.def

The configuration data from this file can be taken over to a TCC domain either with the *tccadd* command or using the *Update Domain ...* menu interface.

bssymm.cnf

This file of not intended for the management platform, but for the agent side. It contains the following four lines:

```
<NJD0010 SOURCE=SYMMETRIX DEVICE=SYMMETRIX>
<NJD0011 SOURCE=SYMMETRIX DEVICE=SYMMETRIX>
<NJD0012 SOURCE=SYMMETRIX DEVICE=SYMMETRIX>
<NJD0013 SOURCE=SYMMETRIX DEVICE=SYMMETRIX>
```

These lines must be added to the configuration file of the Console Monitor subagent so that it can forward the Symmetrix messages as traps (cf. page 69). The value of the DEVICE parameter must also be matched. This parameter value also indirectly determines the name of the node in the network map of the management station. The community name

and the node name in the network map must be identical for the TransView Control Center. The value used for the DEVICE parameter can therefore only be a unique node name that is used for the entire network map.

upd-domain

This is a procedure file that is stored in a *bin* subdirectory of the *SMBS2* directory. It makes it easy to adopt the definitions from the *bs2symm.def* file for the TCC domains. First of all, all the domain names are listed so that the user can select the ones for which the *tccadd* command is to be executed. This procedure is also called when installing in the SMBS2 post-installation procedure.

Deinstallation

The deinstallation procedure for TransView CC largely corresponds to the one described for TransView SNMP (cf. page 112); however, please observe the following additions. The *TV-SMBS2* subdirectory in the CC installation directory and all its files are deleted. Similarly, the *BS2-Symmetrix* application with the associated event definitions are removed from the TransView Control Centers configuration. The *Symmetrix* generated during installation is retained.

3.4.4.2 Configuring on TransView Control Center

The SMBS2 installation created a *SMBS2* subdirectory in the CC installation directory (generally in the path */opt/tcc*) containing the *bs2symm.def* and *bs2symm.cnf* files, as well as the *bin* subdirectory with the *upd-domain* procedure.

If the Symmetrix events notified to the console are to be monitored at a BS2000/OSD system, you must perform the following steps.

1. The configuration file of the Console Monitors must be extended to include four lines of the *bs2symm.cnf* file. The DEVICE designation must be matched to ensure that this designation is a unique node name on the management station side in the network map. The SOURCE parameter should remain unchanged, since it is part of the application name in the TransView Control Center.
2. A subnet map with a Symmetrix icon is created for the BS2000/OSD system in the network map of the management station. The procedure for setting up this icon corresponds to description on page 107.
3. The node name must correspond to the community for the Symmetrix icons in the subnet. The community is defined by the DEVICE parameter of the application in the configuration file of the Console Monitor (cf. 1).

4. The *Update Control Center* function must take up the new Symmetrix node and the node for the BS2000/OSD system, which is located in the center of the subnet, in the management by the TCC domains. The BS2000/OSD node should be set up as a real device to which the events are assigned in the event manager, and the Symmetrix node as virtual device, where the events are displayed at the icons.
5. If the configuration data have not be incorporated in the TCC domains provided from the *bs2symm.def* file, this must be carried out later. You can do this using the *tccadd* command or the *upd-domain* script. The latter option allows you to apply the definitions from *bs2symm.def* simultaneously to several domains. The data can also be taken up using the *Update-Domain* function in the *Integrated Applications* window at the dialog interface.
6. The newly set up Symmetrix node must be linked to the application and its linked events in the *Activate Events and Reactions* window, provided that it does not have the name *Symmetrix*. First select *Applications* as the basic assignment list and mark the *BS2-Symmetrix* application there. The nodes and events linked to this application are also marked. When the new Symmetrix nodes have been marked in the node list, press the *create* button to create the required link.

3.4.5 Integration in the OpenView Network Node Manager

SMBS2 contains an OpenView application for BS2000/OSD which is integrated into the user interface of the network node manager. OpenView, consisting of version 3.3 or 4.1 of the software packages OV-IC, OV-NNMGR and OV-SNMPRN is required for correct installation and error and problem-free operation of OV-SMBS2.
SMBS2 is not required for using the SNMP master agent and its subagents in the BS2000/OSD system. It is only integrated on the OpenView management station.

SMBS2 contains the application-specific MIBs and the following files to supplement the OpenView NNMGR interface:

– The application file which, together with the MIB files, forms the core of the package OV-SMBS2. It contains the definitions for the menus and windows which are added to the OpenView user interface. The application in the package only uses the standard windows which are also available in the application builder. These are forms for object lists, tables and graphics.

– Three files containing help texts with information on the three window types used.

– Bitmap files for displaying a BS2000/OSD host in the network map (with six different icon sizes). Each bitmap file has six mask files.

– The configuration files consisting of one symbol file, one field file and two files containing the lines which must be added to the *oid_sym* and *oid_to_type* configuration files.

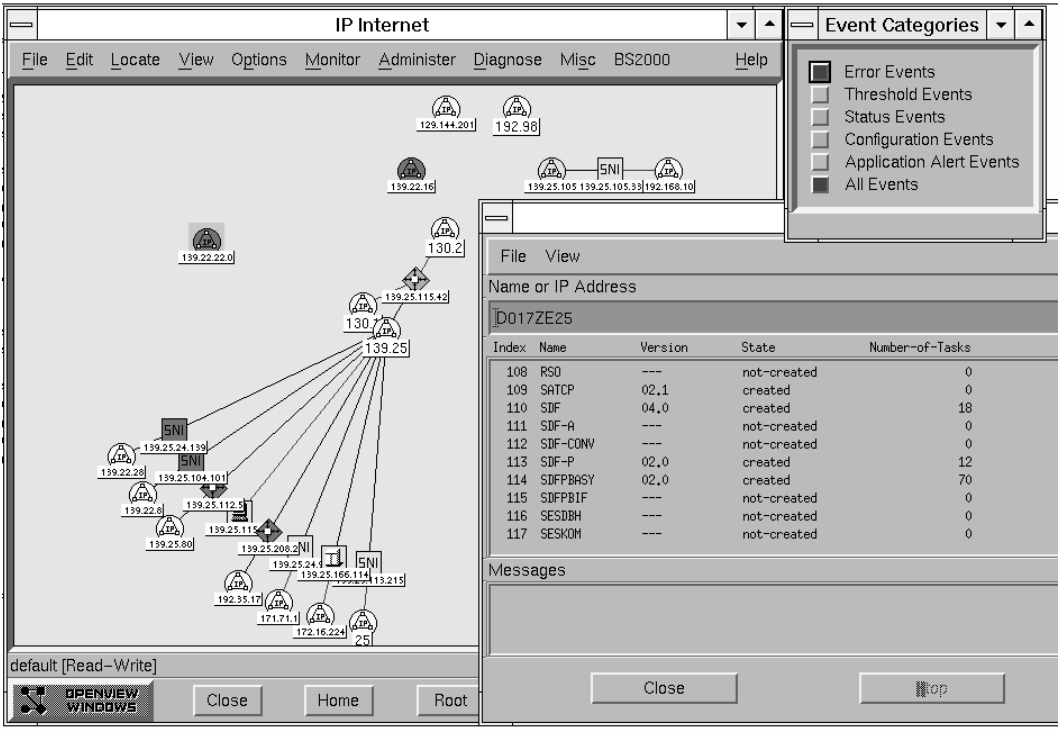


Figure 11: OpenView network map

3.4.5.1 Installing on OpenView

In contrast to TransView SNMP, where it is possible to install different versions in parallel, SMBS2 can only be installed for the last installed and actually used version with OpenView NNMGR. If several versions of OpenView NNMGR are found under different path names, these are displayed. When you select one of these versions, however, the installation is attempted for the last installed version. For this reason, the user is shown the SMBS2 path names used for confirmation during installation.

SMBS2 installation depends on the OpenView version because different directories are used. There is no difference in the general procedure. The main part of SMBS2 installation is the distribution of files into the different directories. These directories are searched and the files found are interpreted when OpenView is loaded. Since the function of a file is defined by the directory in which it resides for OpenView, it is mandatory that all files in a directory have the same format. For example, all files in the */usr/OV/registration/C/ovmib* directory are interpreted as by OpenView V 3.3 as application files. If other files are found in this directory, error messages are output when OpenView is loaded. The same applies for the field, symbol and bitmap files.

The following directories are used in the two supported OpenView versions:

File	Version 3.3	Version 4.1
MIB files	/usr/OV/snmp_mibs	/var/opt/OV/share/snmp_mibs
Application file	/usr/OV/registration/C/ovmib	/etc/opt/OV/share/registration
Help files	/usr/OV/help/C/ovmib/OVW/Functions	/var/opt/OV/share/help/C/ ovmib/OVW/Functions
Bitmap files	/usr/OV/bitmaps/C	/etc/opt/OV/share/bitmaps/C
Symbol file	/usr/OV/symbols/C	/etc/opt/OV/share/symbols/C
Field file	/usr/OV/fields/C	/etc/opt/OV/share/fields/C
Configuration file	/usr/OV/conf/C	/etc/opt/OV/share/conf/C

Any existing files with the same name are overwritten. SMBS2 cannot be installed in the OpenView NNMGR environment under a different path name. The path names are loaded by the *ov.envvars.sh* script in NNMGR Version 4.1.

postinstall procedure

Three processing steps must be executed in the following *postinstall* procedure.

1. The contents of the MIB files must be transferred into the database. This is done by calling the *xnmloadmib* command for each MIB. An update may not be necessary if the MIBs have already been loaded by a previous installation. The operator is therefore asked during installation if he wishes to reload the MIBs. The default response is *YES*. The loaded MIBs are overwritten in this case. If the MIBs are not reloaded, this may lead to errors when displaying objects due to incompatibilities between MIBs and window definitions.
2. A line for the BS2000/OSD symbol must be added to both the *oid_to_sym* and *oid_to_type* assignment files.
3. The entries in the field file must be transferred into the OpenView database with the *own-fields* command. This last step is irreversible. The field entries can only be removed from the database by exchanging the complete database.

The SMBS2 installation is only fully effective when OpenView is started for the first time after installation completion. Only loading the MIBs has an effect on OpenView calls in the current session. The MIB structure is known to the MIB browser immediately after they are loaded to enable it to read and set the object value.

3.4.5.2 Configuring on OpenView NNM

After SMBS2 has been successfully installed, there is an entry in the menu bar for the BS2000/OSD user interface. The functions of the installed application can, however, only be used if a BS2000/OSD system has been registered with the management by placing a BS2000/OSD icon in the network map. This can be done automatically if the *netmon* detects a BS2000/OSD system with an active SNMP agent when it scans the network. Otherwise, the icon is added with the following steps:

- Start the OpenView Network Node Manager.
- Change to the required network map.
  It may be necessary to create a new network map for the subnetwork containing the BS2000/OSD system. Refer to the OpenView manual for the necessary procedure.
- Select the *Add object* function in the *Edit* menu.
- A window titled *Add object: Palette* appears. Various icons are depicted in the upper half of the window below the title *Symbol classes*. You must select the *Computer* icon by clicking on it with the left mouse button. The field below it is then filled with icons. These are the *Symbol subclasses for class computer*. One of them is the BS2000/OSD icon.
- Grab the BS2000/OSD icon by pressing the center mouse button and drag it into the network map while holding the center mouse button down.
- Releasing the mouse button places the BS2000/OSD icon on the marked position and opens the *Add object* window. In this window you can input the attributes for the new symbol in the network map.
  The most important are the *Label,* the *Selection name* and the object attributes which define the relationship to the real object. The object attributes are split into three groups which are handled in different dialog boxes. The window for an attribute group is opened by activating the *Set object attributes...* button. This button is only selectable if an attribute group is marked in the adjacent list.

The three object attribute groups have the following meaning:

– Capabilities

  Nothing can be entered in this window. All switches in the window are shaded, i.e. they are locked. All attribute values shown are fixed to the BS2000/OSD symbol and are therefore shown but cannot be changed. It is important that the BS2000/OSD icon satisfies the properties *isSNI* and *isBS2000* as this is required by the selected symbol as the execution condition for the separate BS2000/OSD application actions.

– General Attributes

  This group comprises four attributes which, in contrast to the first group, can also be set. These are the attributes *isSNMPSupported*, which is set, *isSNMPProxied*, which is not set and the two attributes *Vendor* and *SNMPAgent* which have the values *SNI* and *SNI BS2000-SNMP-Agent*. The values of these attributes should not be changed. The first two attribute values also belong to the execution condition of the BS2000/OSD application actions.

– IP map

  This group of attributes is the most important of the three because the network connection data to the displayed device is entered here. This is the host name, the IP address and the subnetwork mask. The data must be verified before it is accepted. This check is either automatic or it must be started explicitly by activating the *Verify* button. The *OK* button is unlocked if the check is successful. The window can be closed with this and the data entered in the window is transferred into the database.

The window is closed with the *OK* button after all attributes have been entered.

- A specific SNMP configuration is used for communication with an agent as default. A different SNMP configuration must be defined if a special port number or another community is to be used. The dialog box for this can be opened with *SNMP configuration...* in the *Options* menu. Special OpenView SNMP parameters can be entered in the lower half of the dialog box for a single target device or a group of them.

3.4.5.3 Deinstallation

During deinstallation, the status of OpenView NNM should be restored as far as possible before installing SMBS2. At the start of deinstallation, all platforms on which the SMBS2 package have been installed are listed. Versions that are older than V3.0A cannot be detected. Deinitialization is carried out only on the platforms displayed. Since deinstallation also deletes the files installed by older versions, without reconfiguring the platforms affected, you must perform reinstallation on these platform to ensure correct operation.

All package files are deleted and the entries for the BS2000/OSD icon removed from the allocation tables *oid_to_sym* and *oid_to_type*. The new field entries can no longer be removed from the database, since it is not possible to modify the database as a rule. The user may choose not to unload the BS2000/OSD MIBs. In this case they are retained in the MIB browser.

3.5 Installing the management applications

Installation of the *tclset* software package is required in order to use the BMBS2, CMBS2 and PMBS2 management applications on UNIX and Windows NT.

The SMAWtcl software package must be installed in order to use the management applications from the packages SMAWbmbs2, SMAWcmbs2 and SMAWpmbs2 on Solaris.

The following sections describe the installation of the interpreter and of the applications on Solaris and Reliant UNIX, as well as on Windows NT.

When using the operating systems Solaris and Reliant UNIX, the trap server can also be installed.

The supplied CD has the following contents:

tclset / SMAWtcl	Interpreter for Tcl/Tk (required for the applications BMBS2, CMBS2 and PMBS2 or SMAWbmbs2, SMAWcmbs2, SMAWpmbs2)
BMBS2 / SMAWbmbs2	BCAM monitor for BS2000/OSD
CMBS2 / SMAWcmbs2	Console Monitor for BS2000/OSD
PMBS2 / SMAWpmbs2	Performance Monitor for BS2000/OSD
trpsrv / SMAWtrpsv	Trap server for Reliant UNIX or Solaris

A detailed description of the configuration and operation of CMBS2 and PMBS2 can be found in the chapter “Operating the management station” (see page 333).

3.5.1 Installing on Solaris and Reliant UNIX

Installing the interpreter for Tcl/Tk applications

The following must be installed, depending on the operating system used:

– Tcl-Set interpreter for Solaris: SMAWtcl
– Tcl-Set interpreter for Reliant UNIX: tclset

Installing the Tcl-Set interpreter for Tcl/Tk applications on Solaris

The SMAWtcl packages comprises:

– *bin* directory with the main program
– *lib* directory with the Tcl scripts and dynamic libraries

During the installation procedure, you set the installation directory for *SMAWtclset*; the default directory for this purpose is *opt/SMAW*. This directory must already exist. A directory called *SMAWtcl* is created with the subdirectories named above.

Installation is not permitted in the root directory or under */var*.

Installing the Tcl-Set interpreter for Tcl/Tk applications on Reliant UNIX

The tclset package comprises:

– *bin* directory with the main program
– *lib* directory with the Tcl scripts and dynamic libraries

During the installation procedure, you set the installation directory for *tclset*. The default directory for this purpose is */usr/local*; this directory must already exist. A directory called *tcl* is created with the subdirectories named above.

Installation is not permitted in the root directory or under */var*.

Installing the trap server

The trap server for Solaris (*SMAWtrpsv* package) and Reliant UNIX (*trpvsrv* package) is installed with *pkgadd* in accordance with package procedures. Further information on the trap server for Solaris and Reliant UNIX can be found in the chapter “Trap server for Solaris and Reliant UNIX” on page 431.

Installing the applications

During the installation procedure, you set the installation directory for the management applications. The suggested directory is the base directory of the Tcl-Set interpreter. If this directory does not exist, it is created. Other specified directories must already exist.

The subdirectories *bin*, *lib*, *help*, *asn1, config* and *bitmaps* are created in the directory *Bmon* (BMBS2), *Cmon* (CMBS2) or *Pmon* (PMBS2).

The necessary environment variables are set in the routine that activates the main program.

Configuring CMBS2 (Solaris and Reliant UNIX)

The *tclset* package contains a trap daemon called *nmtrapd*. The package *SMAWtrpsv* (Solaris) or *trpsrv* (Reliant UNIX) contains another, more comprehensive trap distributor that should be given priority over *nmtrapd* (see chapter “Trap server for Solaris and Reliant UNIX” on page 431).

A trap distributor fulfils two tasks:

- It multiplies the traps arriving in the system
  - for the various Console Monitor applications
  - for other applications that receive traps
- It also permits non-privileged applications to listen to the root-privileged port 162

It thus makes sense to use a trap distributor if the Console Monitor application is not started exclusively under *root*. In all other cases, no trap distributor is needed.

You can use the environment variable TNM_TRAPD to define:

– whether a trap distributor is to be used, and if so
– which trap distributor is to be used

The environment variable TNM_TRAPD can accept the following values:

Value	**Meaning**
NMTRAPD	The trap distributor *nmtrapd* is used.
TRPSRV	The trap distributor *trpsrv* is used.
TRPTCC	The TransView trap distributor *trd_distr* is used.
NO	No trap distributor is used; the port is set using the CMBS2 interface (default value: 162).

3.5.2 Installing on Windows NT

Installing the interpreter Tcl-Set for Tcl/Tk applications

The *tclset* package is the basic package for using the management applications BCAM Monitor, Performance Monitor and Console Monitor on Windows NT.

The package includes:

– the *Bin* directory with the main program and dynamic libraries,
– the *Inst* directory containing the installation procedures,
– the *Lib* library with the Tcl scripts.

During the installation procedure, you can specify the directory where the product is to be installed. The default directory is *C:\Programme\Tcl*. The above-mentioned subdirectories *Bin*, *Inst* and *Lib* are created here.

Installing the applications

The *tclset* package version ≥ 05.0A.00, which contains the interpreter, is required for operating BMBS2, CMBS2 and PMBS2.

The application includes

– the *Asn1* directory with the MIB files,
– the *Bin* directory with the main program,
– the *Bitmaps* directory with the bitmap files,
– the *Help* directory with the help texts,
– the *Inst* directory with the installations and deinstallation programs and the installation logfile,
– the *Lib* directory with the Tcl scripts,
– the *config* directory with the configuration file *cmon_cnf.prt* or *pmon_cnf.prt*.

During the installation procedure, you can specify the directory where the BMBS2, CMBS2 or PMBS2 is to be installed. The installation path specified must not contain any blanks.

The basic product subdirectory *tclset* under the *appl* directory is offered for installation. This directory is created if it does not exist.

The above-mentioned subdirectories *Asn1*, *Bin*, *Bitmaps*, *Help*, *Inst* and *Lib* are created in the BMBS2, CMBS2 or PMBS2 directory. If no *cmon.cnf* file exists in the installation directory, the *cmon_cnf.prt* prototype file is copied to *cmon.cnf*. The same applies to the *pmon.cnf* file.

4 Operation

The delivery unit SBA-BS2 V5.0 contains the master agent, the supervisor subagent, the HTML subagent, the Application Monitor subagent and the Console Monitor subagent. A set of subagents for BS2000/OSD specific management tasks is supplied with SSC-BS2 V5.0. Two additive subagents, SSA-SM2-BS2 and SSA-OUTM-BS2, are also available for performance monitoring and *open*UTM application monitoring. *open*Net Server provides both a MIB-II subagent that conforms to RFC 1213 and a subagent with a BCAM specific MIB. This chapter describes startup and shutdown of the separate components in BS2000/OSD as well as commands used to send traps. The final section provides information on the procedure in the event of errors.

4.1 Startup and shutdown

The subagents are only functional if the master agent is running. Except for the supervisor subagent, they can be started and stopped separately at any time.

Prerequisites for starting the agents are:

– an operational LAN1 connection between BS2000/OSD computer and management platform
– a started POSIX subsystem
– an installed SNMP subsystem
– privileges (on next page).

The following privileges are required to start the relevant agents:

Command	Privilege
START-SNMP-MASTER	NET-ADMINISTRATION
START-SNMP-APPMON	NET-ADMINISTRATION
START-SNMP-CONSMON	NET-ADMINISTRATION
START-SNMP-HTML	NET-ADMINISTRATION
START-SNMP-AVAS	NET-ADMINISTRATION
START-SNMP-FT	FT-ADMINISTRATION
START-SNMP-HIPLEX-AF	NET-ADMINISTRATION
START-SNMP-HOSTRES	NET-ADMINISTRATION
START-SNMP-HSMS	HSMS-ADMINISTRATION
START-SNMP-OMNIS	NET-ADMINISTRATION
START-SNMP-PRINTSERVICE	PRINT-SERVICE-ADMINISTRATION
START-SNMP-SESAM	NET-ADMINISTRATION
START-SNMP-STORAGE	NET-ADMINISTRATION
START-SNMP-PERFMON	SW-MONITOR-ADMINISTRATION
START-SNMP-UTM	NET-ADMINISTRATION
START-SNMP-MIB-MIB2	NET-ADMINISTRATION
START-SNMP-MIB-BCAM	NET-ADMINISTRATION

In order to start the master agent, the BS2000/OSD ID must possess the POSIX User ID 0 (SYSROOT).

Either the TSOS ID must be used to stop the agent or the same command as was used for the start command.

rc scripts

As of version 5.0, rc scripts are installed, which permit an automatic start of the agent on startup of POSIX or an automatic stop when POSIX is terminated. These procedures are stored in the */etc/rc0.d* or */etc/rc2.d* directories. Apart from the one for starting the master agent, all calls are commented out. In this way, it is possible to customize the procedures to match your configuration.

Name of the procedure	Function
S90snmpsba	Starting the agents of SBA-BS2
S91snmpssc	Starting the agents of SSC-BS2
S91snmputm	Starting the *open*UTM agent (SSA-OUTM-BS2)
S91snmpsm2	Starting theSM2 agent (SSC-SM2-BS2)
K10snmpsba	Stopping the agent of SBA-BS2
K11snmpssc	Stopping the agent of SSC-BS2
K11snmputm	Stopping the *open*UTM agent (SSA-OUTM-BS2)
K11snmpsm2	Stopping the SM2 agent (SSC-SM2-BS2)

Trace files

During operation of the agents, trace files are generated under the ID under which the agent was started. The trace files are generated with the names *SYSTRC.SNMP.*<*agent*>.<*date*>.<*time*> and can be deleted after the relevant agent has been stopped, provided they are no longer required. The messages from the agents are stored as default in these trace files in the BS2000/OSD file system.

The name of the trace file *SYSTRC.SNMP.*<*agent*>.<*date*>.<*time*> is made up as follows:

<agent>	the name of the agent program.
<date>	the current date in the form: YYYY-MM-DD
<time>	the current time in the form: HHMMSS

Example:

```
/FS SYSTRC.SNMP.
%       9 :2DC1:$DC14.SYSTRC.SNMP.AVASAGT.1999-02-19.101643
%       9 :2DC1:$DC14.SYSTRC.SNMP.AVASAGT.1999-03-07.165625
%      78 :2DC1:$DC14.SYSTRC.SNMP.AVASAGT.1999-03-07.171451
```

4.1.1 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 55).

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

```
snmpdm
```

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

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.

4.1.2 BASIC AGENT subagents

In addition to the supervisor subagent for monitoring the subagents logged on at the master agent, the Application Monitor subagent, Console Monitor subagent and the HTML subagent are further subagents of the BASIC-AGENT. The Application Monitor subagent is used to monitor subsystems, BCAM and user applications, as well as job variables and logging files. The Console Monitor subagent is used to monitor the console. It can be used to display console messages and also allows console commands to be input. The HTML subagent is required when SNMP-based management information is to be provided in custom pages via the WWW.

4.1.2.1 Supervisor subagent

The close link to the master agent means that there is no separate start or stop command for the supervisor subagent. The supervisor subagent start is initiated by a corresponding entry in the */etc/snmp/agt/snmpd.cnf* file. While this entry is present, the supervisor subagent is always started and terminated automatically with the master agent.

4.1.2.2 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=**<u>*STD</u>** / <product-version>

or in the POSIX:

```
appmoncmd T
```

Description of operands:

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

MONJV=<u>*NONE</u> / <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=<u>*STD</u> / <integer 1 .. 32767> / *NO
Maximum CPU runtime in seconds. If *STD is specified, the generated default value is used.

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

FILE-NAME=<u>*NONE</u> / <filename 1..54 without-gen-vers>
You may specify a configuration file when you start the Application Monitor subagent (see page 56). 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=<u>5</u> / <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.
The monitoring interval for subsystems is calculated as five times the value set for the timer interval, i.e. 25 seconds in the standard case.
It may happen that changes of state in applications or job variables may not be not notified until the timer interval has expired.
The monitoring interval for DCAM applications is calculated as 60 times the value set for the time interval, i.e. 5 minutes in the standard case.

4.1.2.3 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>
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.

4.1.2.4 HTML subagent

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

1. Starting in BS2000/OSD:

/START-SNMP-HTML
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>

2. Starting in the POSIX shell:

```
htmlagt [-t <int>]
```

The HTML subagent is stopped (irrespective of the environment in which its was started) in BS2000/OSD:

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

or in the POSIX shell:

```
htmlcmd 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.

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.

4.1.3 Subagents in the STANDARD COLLECTION

In Version 5.0, the SNMP STANDARD-COLLECTION-BS2000 contains the subagents for AVAS, FT-BS2000 or *open*FT (BS2000), HIPLEX-AF, HSMS, OMNIS, Host Resources, PrintService, SESAM, storage management and for SM2-based performance monitoring. Detailed descriptions of these subagents can be found in chapter 6.

Additional subagents are supplied with *open*Net Server (MIB-II and Private MIB) or as additive subagents (SM2 and UTM). Information on these subagents can be found on page 155.

4.1.3.1 AVAS subagent

Configuration work required prior to starting the AVAS subagent is described on page 76.

Starting the AVAS subagent in BS2000/OSD:

/START-SNMP-AVAS
VERSION=***STD** / <product-version> , **MONJV**=***NONE** / <filename 1 .. 54> , **CPU-LIMIT**=***STD** / <integer 1 .. 32767> / ***NO** , **JOB-CLASS**=***STD** / <name 1 .. 8> , **FILE-NAME**=<filename 1 .. 54> , **DELAY-TIME**=60 / <integer 0 .. 3600> , **TIMER-INTERVAL** = 5 / <integer 1 .. 32767>

Stopping the AVAS subagent in BS2000/OSD:

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

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.

FILE-NAME=<filename 1 .. 54>
The Name of the AVAS generation file GENPAR is assigned with FILE-NAME.

DELAY-TIME=60 / <integer 0 .. 3600>
If the subagent has no connection to AVAS, it attempts to set one up when new requests are issued. DELAY-TIME specifies the time (in seconds) which may elapse between connection attempts. You can use this to suppress additional connection attempts by setting DELAY-TIME=0.

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.

4.1.3.2 Subagent for *open*FT

Starting the FT subagent in BS2000/OSD:

/START-SNMP-FT
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>

or in the POSIX shell:

```
ftagt [-t <int>]
```

The subagent is terminated, independent of the environment in which it was started, with the BS2000/OSD command:

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

or in the POSIX shell as of BS2000/OSD V2.0:

```
ftcmd 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.

4.1.3.3 Subagent for HIPLEX-AF

The HIPLEX-AF subagent is started in the POSIX shell or in BS2000/OSD.

1. Starting in BS2000/OSD:

/START-SNMP-HIPLEX-AF
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> , **USER-ID**=**TSOS** / <name 1..8> , **TIMER-INTERVAL** = 5 / <integer 1 .. 32767>

2. Start in the POSIX shell:

```
hiplexAFagt [-t <int>][ -u <user-id>]
```

The HIPLEX-AF subagent terminated, independent of the environment in which it was started, with the BS2000/OSD command:

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

or in the POSIX shell:

```
hiplexAFcmd 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.

USER-ID=TSOS / <name 1..8>
User ID required as process ID for HIPLEX AF.

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.

4.1.3.4 HSMS subagent

The hsms subagent can be started in the BS2000/OSD or POSIX shell.

1. Starting in BS2000/OSD:

/START-SNMP-HSMS
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> , **HSMS-LIBRARY =** ***STD** / <filename 1 .. 54>

2. Starting in the POSIX shell:

```
hsmsagt [-t <int>] -l <HSMS-library>
```

Regardless of the environment in which it was started, the HSMS subagent is terminated in BS2000/OSD with:

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

or in the POSIX shell with:

```
HSMScmd T
```

Description of operands:

VERSION=*STD / <product-version>
Defines the agent version to be started or stopped. This specification is not evaluated at present.

MONJV=*NONE / <filename 1..54 without-gen-vers>
Name of the job variable in which agents are to be monitored. The default setting is *NONE - no monitoring by a job variable.

CPU-LIMIT=*STD / <integer 1 .. 32767> / *NO
Specifies the 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.

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

HSMS-LIBRARY=*STD / <full-filename 1..54>
Path name of HSMS-SYSLIB. If *STD is specified, IMON is used to determine the name.

4.1.3.5 Host Resources subagent

The subagent can be started in BS2000/OSD or in the POSIX shell.

Starting in BS2000/OSD:

/START-SNMP-HOSTRES
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>

or in the POSIX shell:

```
hostresagt [-t <int>]
```

Stopping the subagent in BS2000/OSD:

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

or in the POSIX shell:

```
hostrescmd 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 at five seconds.

4.1.3.6 Subagent for OMNIS

The subagent can be started in BS2000/OSD or in the POSIX shell.

Starting in BS2000/OSD:

/START-SNMP-OMNIS
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> , **CONFIGURATION-FILE** = <filename 1 .. 54>

or in the POSIX shell:

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

Stopping the subagent in BS2000/OSD:

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

or in the POSIX shell:

```
omniscmd 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.
After a period six times the timer interval (i.e. 30 seconds in the standard case), all OMNIS messages present at the DCAM interface are collected and sent to the management station as a trap.

CONFIGURATION-FILE-=<filename 1 .. 54>
At the start, the subagent must be assigned the file designated by CONFIGURATION-FILE=<*filename*>, which contains the name(s) of the OMNIS system(s) to be monitored.

4.1.3.7 Subagent for SESAM

The subagent can be started in BS2000/OSD or in the POSIX shell (BS2000/OSD ≥ V2.0).

Starting in BS2000/OSD:

/START-SNMP-SESAM
VERSION = ***STD** / <product-version> , **MONJV** = ***NONE** / <filename 1 .. 54> , **CPU-LIMIT** = ***STD** / <integer 1 .. 32767> / ***NO** , **JOB-CLASS** = ***STD** / <name 1 .. 8> , **FILE-NAME** = <filename 1 .. 54> , **TIMER-INTERVAL** = 5 / <integer 1 .. 32767>

or in the POSIX shell:

```
sesamagt –f <inputfile>
        [–t <int>]
```

Stopping the subagent in BS2000/OSD:

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

or in the POSIX shell:

```
sesamcmd 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.

FILE-NAME=<filename 1 .. 54>
When the subagent is started, the configuration file assigned with FILE-NAME=<*filename*> or <*inputfile*> must be specified. The configuration file must always be cataloged in the BS2000/OSD file system independently of the start command page 78.

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 relation between the server and the database is checked each time the timer interval expires.

4.1.3.8 Subagent for Spool & Print Service

Starting in BS2000/OSD:

/START-SNMP-PRINTSERVICE
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>

or in the POSIX shell:

```
printagt [-t <int>]
```

Stopping the subagent in BS2000/OSD:

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

or in the POSIX shell:

```
printcmd 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.

4.1.3.9 Subagent for storage management

The subagent can be started in BS2000/OSD or in the POSIX shell.

Starting in BS2000/OSD:

/START-SNMP-STORAGE
VERSION=***STD** / <product-version> , **MONJV**=***NONE** / , **CPU-LIMIT**=***STD** / <integer 1 .. 32767> / ***NO** , **JOB-CLASS**=***STD** / <name 1 .. 8> , **FILE-NAME**=***NONE** / <filename 1 .. 54> , **TIMER-INTERVAL** = 5 / <integer 1 .. 32767>

In the POSIX shell:

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

The subagent is stopped, regardless of the environment in which it was started, with the BS2000/OSD command:

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

or in the POSIX shell:

```
storagecmd 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.

FILE-NAME=<filename 1 .. 54>
Name of the input file if pubsets or disks are to be monitored.

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 subagent monitors pubsets or disks are monitored for the storage management at intervals 6 * TIMER-INTERVAL seconds.

4.1.3.10 Subagent for basic performance monitoring with SM2 (PerfMonB)

Prerequisite for starting the subagent for performance monitoring with SM2 (PerfMonB) is a started SM2 subsystem (SM2 must be started explicitly in BS2000/OSD V2.0)

Starting the SM2 subagent in BS2000/OSD:

/START-SNMP-PERFMON
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>

Stopping the SM2 subagent in BS2000/OSD:

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

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.

4.1.4 Additive subagents

Additional subagents for SNMP-based management in BS2000/OSD are included in *open*Net Server on the one hand, and are available as additive subagents for SM2 and *open*UTM with the products SSA-SM2-BS2 and SSA-OUTM-BS2.

4.1.4.1 Subagents for *open*Net Server and *inter*Net Services

Two subagents are supplied with the BS2000/OSD transport system *open*Net Server.The first is the MIB II subagent, which supports read access to the MIB-II according to RFC1213. The second is the BCAM subagent, which uses the private MIB to provide information on BCAM-specific values and settings (see the manual “SNMP Management for *open*Net Server”).

Starting the MIB-II subagent in BS2000/OSD:

/START-SNMP-MIB-MIB2
VERSION=***STD** / <product-version> , **MONJV**=***NONE** / <filename 1 .. 54> , **CPU-LIMIT**=***STD** / <integer 1 .. 32767> / ***NO** , **JOB-CLASS**=***STD** / <name 1 .. 8>

or in the POSIX shell:

```
mib2agt
```

Stopping the MIB-II subagent in BS2000/OSD:

/STOP-SNMP-MIB-MIB2
VERSION=***STD** / <product-version>

or in the POSIX shell:

```
mib2cmd T
```

Starting the BCAM subagent in BS2000/OSD:

/START-SNMP-MIB-BCAM
VERSION=***STD** / <product-version> , **MONJV**=***NONE** / <filename 1 .. 54> , **CPU-LIMIT**=***STD** / <integer 1 .. 32767> / ***NO** , **JOB-CLASS**=***STD** / <name 1 .. 8>

or in the POSIX shell:

```
bcamagt
```

Stop the BCAM subagents in BS2000/OSD:

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

or in the POSIX shell:

```
bcamcmd 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.

4.1.4.2 SSA-SM2-BS2 subagent for performance monitoring (with SM2)

The following prerequisites are required for starting the subagent for performance monitoring with SM2 (PerfMonF):

– A started SM2 subsystem (SM2 must be started explicitly in BS2000/OSD V2.0)
– Complete and successful installation of SSA-SM2-BS2

Example

```
/EXEC $SM2
*call-admin-part
%//set-periodic-task-parameter log-tasks=*none
%//start-measurement-program per
%//start-measurement-program utm
%//call-eval-part
*end
```

SM2 measurement must be activated in *open*UTM for measuring *open*UTM applications:

```
kdcappl sm2=on
```

Starting the performance subagent in BS2000/OSD:

/START-SNMP-PERFMON
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>

or in the POSIX shell:

```
perfagt
```

Stopping the performance subagent in BS2000/OSD:

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

or in the POSIX shell:

```
perfcmd 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.

4.1.4.3 Subagent SSA-OUTM-BS2 for *open*UTM applications

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

/START-SNMP-UTM
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>

or in the POSIX shell:

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

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 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.

4.2 Trap send commands

The trap send commands contained in SBA-BS2 can be used to send traps from BS2000/OSD to a target computer. The ID used to send the command requires the NET-ADMINISTRATION privilege.

4.2.1 START-SNMP-TRAPSEND

The START-SNMP-TRAPSEND command sends any desired SNMP trap to a target computer.

START-SNMP-TRAPSEND

VERSION=***STD**
, **MONJV**=***NONE** / <filename 1 .. 54 without-gen-vers>
, **CPU-LIMIT**=***STD** / <integer 1 .. 32767> / ***NO**
, **SOURCE-IP-ADDR**=***NONE** / <c-string 1..15>
, **DESTINATION-IP-ADDR**=***NONE** / <c-string 4..15>
, **COMMUNITY**=<c-string 1..15_with-lower-case>
, **GENERIC-TRAP**=**0** / <integer_0..6>
, **SPECIFIC-TRAP**=**0** / <integer 0..2147483647>
, **ENTERPRISE**=***1.3.6.1.4.1.231.1.6** / <c-string 3..55 _with-lower-case>
, **SYSUPTIME**=**0** / <integer -2147483648..2147483647>
, **TRAP-VARIABLE-NAME**=***NONE** / <c-string 3..1800_with-lower-case>
, **TRAP-VARIABLE-TYPE**=**D**/ <c-string 1..3_with-lower-case>
, **TRAP-VARIABLE-VALUE**=***NONE** / <c-string ..1800_with-lower-case>

Description of operands:

VERSION=*STD
Defines the version of the program. This operand is currently not interpreted.

MONJV=*NONE / <filename 1 .. 54 without-gen-vers>
Name of the job variable that monitors the program. 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.

SOURCE-IP-ADDR=*NONE / <c-string 1..15>
IP address of the source host. If the host has several IP addresses, the first one is used per default.

DESTINATION-IP-ADDR=*NONE / <c-string 4..15>
IP address of the target computer. If no specification is made (*NONE), the trap is sent to all the hosts defined in the */etc/snmp/mgr/trap.cnf* file.

COMMUNITY=<c-string 1..15_with-lower-case>
Community string with which the trap is to be sent.

GENERIC-TRAP=0 / <integer 0..6>
Generic trap (0 - 6), corresponds to the specification in RFC1215

SPECIFIC-TRAP=0 / <integer 0..2147483647>
Specific trap number if GENERIC-TRAP=6 was specified.

ENTERPRISE=*1.3.6.1.4.1.231.1.6 / <c-string 3..55_with-lower-case>
Object Identifier of ENTERPRISE

SYSUPTIME=0 / <integer -2147483647..2147483647>
SYSUPTIME can be used to identify the trap in greater detail.
Specified in seconds.

TRAP-VARIABLE-NAME=*NONE / <c-string 3..1800_with-lower-case>
Object identifier to be included.

TRAP-VARIABLE-TYPE=i / <c-string 1..3_with-lower-case>
Defines the type of the value included. The default is: i. Possible values:

i	Integer
o	Octet String
d	Object identifier
a	IP address
D	Display string

TRAP-VARIABLE-VALUE=*NONE / <c-string 1..1800_with-lower-case>
Value included, corresponding to the TRAP-VARIABLE-TYPE.

Entries in the trap.cnf file

The *trap.cnf* file contains the following entries:

```
trap <community-string> <IP address (target)>
```

Meaning:

– <community-string>: freely selectable community string
– <IP address(target)>: IP address of the target computer

4.2.2 SEND-TCC-MSG

The SEND-TCC-MSG command sends an SNMP trap to a target computer in the format of the *tccGenTrap* object. The *tccGenTrap* object is described in the MIB *Tcc-MIB* defined by the TV Control Center. The command is linked to the TV Control Center and CMBS2 management stations, which require this trap.

SEND-TCC-MSG

VERSION=***STD**
, **MONJV**=***NONE** / <filename 1 .. 54 without-gen-vers>
, **CPU-LIMIT**=***STD** / <integer 1 .. 32767> / ***NO**
, **SOURCE-IP-ADDR**=***NONE** / <c-string 1..15>
, **DESTINATION-IP-ADDR**=***NONE** / <c-string 4..15>
, **OBJECT**=***NONE** / <c-string 1..1800_with-lower-case>
, **APPLICATION**=<c-string 1..1800_with-lower-case>
, **TEXT**=<c-string 1..1800_with-lower-case>

Description of operands:

VERSION=*STD
Defines the version of the program. This operand is currently not interpreted.

MONJV=*NONE / <filename 1..54 without-gen-vers>
Name of the job variable that monitors the program. 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.

SOURCE-IP-ADDR=*NONE / <c-string 1..15>
IP address of the source host. If the host has several IP addresses, the first one is used per default.

DESTINATION-IP-ADDR=*NONE / <c-string 1..15>
IP address of the target host. If no specification is made (*NONE), the trap is sent to all the hosts defined in the */etc/snmp/mgr/trap.cnf* file.

OBJECT=*NONE / <c-string 1..1800_with-lower-case>
Name of the object in the network map. This is used as the community and entered in $DEVC$.

APPLICAITON=<c-string 1..1800_with-lower-case>
Name of the application that is entered for $SOURCE$.

TEXT=<c-string 1..1800_with-lower-case>
Text entered for MSG.

Entries in the trap.cnf file

The *trap.cnf* file contains entries of the form:

```
trap <community-string> <IP address(target)>
```

Meaning:

– <community-string>: freely selectable community string
– <IP address(target)>: IP address of target computer

4.3 Procedure in the event of errors

Each agent keeps trace files in which error messages are logged as standard (see page 127). Each agent is provided with its own command program for managing the trace functionality. These separate command programs may only be started if the relevant agent has also been successfully started. The functionality provided by the command program allows you to display and close trace files and terminate the agent concerned.

Defining the trace extent

Only error messages are logged in the trace file as default. If this is not sufficient for diagnosis, the agent concerned must be terminated and then restarted with the normal start command supplemented with *TRACE=*APALL*. The detailed trace is requested in POSIX with the *-apall* switch. Additional trace output to the terminal is requested with *-termout*.

Starting the command program

The agent concerned must be successfully started before the command program can be started. The following table lists the start statements for the separate command programs:

BS2000/OSD commands	POSIX commands	Agent
/START-MASTERCMD	snmpdmcmd	Masteragent / Supervisor
/START-APPMONCMD	appmoncmd	Application Monitor
/START-CONSMONCMD	consmoncmd	Console Monitor
/START-HTMLCMD	htmlcmd	HTML subagent
/START-HOSTRESCMD	hostrescmd	Host Resources subagent
/START-AVASCMD	avascmd	AVAS subagent
/START-FTCMD	ftcmd	FT subagent
/START-HIPLEX-AFCMD	hiplexAFcmd	HIPLEX-AF subagent
/START-HSMSCMD	hsmscmd	HSMS subagent
/START-OMNISCMD	omniscmd	Subagent for OMNIS
/START-SESAMCMD	sesamcmd	SESAM subagent
/START-PRINTCMD	printcmd	Subagent for print and spool management
/START-STORAGECMD	storagecmd	Subagent for storage management
/START-PERFMONCMD	perfmoncmd	Subagent for the performance monitor
/START-UTMCMD	utmcmd	UTM subagent
/START-MIB2CMD	mib2cmd	MIB-II subagent (*open*Net Server)
/START-BCAMCMD	bcamcmd	BCAM subagent (*open*Net Server)

The parameters VERSION, MONJV and CPU-LIMIT apply only for BS2000/OSD commands.

BS2000/OSD commands	POSIX commands	Agent
VERSION=***STD** / <product-version> , **MONJV**=***NONE** / <filename 1 .. 54> , **CPU-LIMIT**=***JOB-REST** / <integer 1 .. 32767>		

The parameters VERSION, MONJV and CPU-LIMIT apply only for BS2000/OSD commands.

When the command program is started, the current subagent data is output first. The following example shows the data of an AVAS subagent.

Example

```
INFO of AVAS subagent
Version :   50A00        FileOut : Yes             Errors  :        0
PID     :     517        TermOut : No              Warnings:        1
LogLevel: APERROR
LogFile : ':2DC1:$DC14.SYSTRC.SNMP.AVASAGT.1999-02-19.101643'
```

Element	Meaning
Version	Subagent version (V4.1A00)
PID	POSIX subsystem process ID (517)
LogLevel	Message classes to be traced (APERROR or APALL)
LogFile	Absolute path name of the current trace file
FileOut	Trace output to file activated/deactivated
TermOut	Trace output to the terminal activated/deactivated
Errors	Number of ERROR messages until now
Warnings	Number of WARNING messages until now

The following trace options are offered when the command program has been started and the RETURN key is pressed in the MAIN MENU:

s	show trace file	Display the trace file
c	save trace file	Save the trace file
x	agent executes command	This function is currently only used by the master agent and by the following subagents: Application Monitor subagent Console Monitor subagent SESAM subagent
T	terminate agent	Terminate the agent
q	quit command program	Terminate the command program

If, for example, the AVAS subagent has not been started, the command program concerned (in this case *avascmd*) terminates with the following error message:

```
--- avascmd: ERROR  14:03:23  08/03/1999
Agent is not running
```

If the following message is not output when a subagent is started, although the subagent has already been terminated, please call the relevant command program and input *-CLEAN* at the input prompt.

```
--- avasagt: ERROR  16:04:23  08/03/1998
Another Agent is probably yet running
Please use command program to terminate that agent correctly
```

5 Functions of the BASIC AGENT

5.1 System and SNMP management (master agent)

The master agent supports system and SNMP management with the aid of two MIB-II groups:

– the group for monitoring the system
– the group for SNMP monitoring

It also provides a series of standardized and proprietary MIBs for SNMP administration.

The master agent supports individual objects of other MIBS which are relevant for SNMP management in BS2000/OSD.

5.1.1 MIB-II values for the system group

MIB definition	definition	Meaning
sysDescr	read-only	sysDescr defines the name of the device, the software version and the hardware type. The description is given only in ASCII characters.
sysObjectID	read-only	sysObjectID defines the exact position of the device to be managed in the SMI Enterprise Subtree.
sysUpTime	read-only	sysUpTime defines the time (in 1/100 seconds) since the last reinitialization of the network management software.
sysContact	read-write	sysContact contains the text string for the contact person and contact address, who is responsible for this manageable node.
sysName	read-write	sysName contains a logical name for the device to be managed, which also corresponds to the full domain name.

Group for system monitoring

MIB definition	definition	Meaning
sysLocation	read-write	sysLocation describes the location of the device.
sysServices	read-only	sysServices defines precisely the services (ISO Layers), which this device supports. The service object corresponds to the value of a sum (starting at the basic value 0).

Group for system monitoring

The host name defined in BCAM is automatically stored as the *sysName*. The values for *sysContact* and *sysLocation* are specified in the Initial System Group (see page 55).

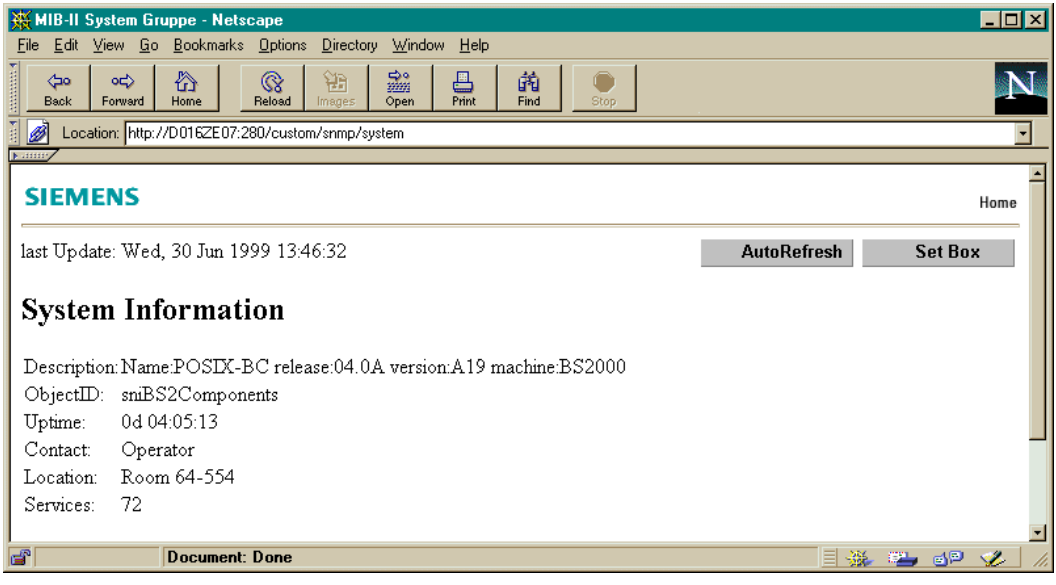


Figure 12: Display of MIB-II values for the system group

5.1.2 MIB-II values for the SNMP group

Object Name	Access	Meaning
snmpInPkts	read-only	Total number of messages sent to the SNMP Entity by the transport service.
snmpOutPkts	read-only	Total number of SNMP messages sent from the SNMP Protocol Entity to the transport service.
snmpInBadVersions	read-only	Total number of SNMP messages sent to the SNMP Protocol Entity and for which a non-supported SNMP version was configured.
snmpInBadCommunityNames	read-only	Total number of SNMP messages sent to the SNMP Protocol-Entity with an SNMP community name that is unknown to this entity.
snmpInBadCommunityUses	read-only	Total number of SNMP messages sent to the SNMPProtocol Entity, which represent an SNMP operation that was not recognized by the SNMP Community stated in the message.
snmpInASNParseErrs	read-only	Total number of ASN.1 or BER errors detected by the SNMP-Protocol-Entity while decoding the SNMP messages received.
snmpInTooBigs	read-only	Total number of SNMP-PDUs sent to the SNMP Protocol Entity and for which the error status field has the value “tooBig”.
snmpInNoSuchNames	read-only	Total number of SNMP-PDUs sent to the SNMP Protocol Entity and for which the error status field has the value “noSuchName”.
snmpInBadValues	read-only	Total number of SNMP-PDUs sent to the SNMP Protocol Entity and for which the error status field has the value “badValue”.
snmpInReadOnlys	read-only	Total number of invalid SNMP-PDUs sent to the SNMP Protocol Entity, and for which the error status field has the value “readOnly”. You should observe that the creation of SNMP-PDUs with the value “readOnly” in the error status field represents an error since such objects serve as an aid to detecting invalid SNMP implementations.
snmpInGenErrs	read-only	Total number of SNMP-PDUs sent to the SNMP Protocol Entity and for which the error status field has the value “genErr”.

SNMP group

Object Name	Access	Meaning
snmpInTotalReqVars	read-only	Total number of MIB objects, which can be called successfully by the SNMP-Protocol-Entity as a result of reception of the valid SNMP-Get-Request and Get-Next-PDUs.
snmpInTotalSetVars	read-only	Total number of MIB objects, which can be modified successfully by the SNMP-Protocol-Entity as a result of the reception of valid Set-Request-PDUs.
snmpInGetRequests	read-only	Total number of SNMP-Get-Request-PDUs accepted by the SNMP Protocol Entity and processed.
snmpInGetNexts	read-only	Total number of SNMP-Get-Next-PDUs accepted by the SNMP Protocol Entity and processed.
snmpInSetRequests	read-only	Total number of SNMP-Set-Request-PDUs accepted by the SNMP Protocol Entity and processed.
snmpInGetResponses	read-only	Total number of SNMP-Set-Response-PDUs accepted by the SNMP Protocol Entity and processed.
snmpInTraps	read-only	Total number of SNMP-Trap-PDUs accepted by the SNMP Protocol Entity and processed.
snmpOutTooBigs	read-only	Total number of SNMP-PDUs created by the SNMP Protocol Entity and for which the value “tooBig” is displayed.
snmpOutNoSuchNames	read-only	Total number of SNMP-PDUs created by the SNMP Protocol-Entity, and for which has the error status field has the value “noSuchName”.
snmpOutBadValues	read-only	Total number of SNMP-PDUs created by the SNMP Protocol-Entity, and for which has the error status field has the value “badValue”.
snmpOutGenErrs	read-only	Total number of SNMP-PDUs created by the SNMP Protocol-Entity, and for which has the error status field has the value “genErr”.
snmpOutGetRequests	read-only	Total number of SNMP-Get-Request-PDUs created by the SNMP-Protocol-Entity.
snmpOutGetNexts	read-only	Total number of SNMP-Get-Next-PDUs created by the SNMP-Protocol-Entity.
snmpOutSetRequests	read-only	Total number of SNMP-Set-Request-PDUs created by the SNMP-Protocol-Entity.
snmpOutGetResponses	read-only	Total number of SNMP-Get-Response-PDUs created by the SNMP-Protocol-Entity.

SNMP group

Object Name	Access	Meaning
snmpOutTraps	read-only	Total number of SNMP-Trap-PDUs created by the SNMP-Protocol-Entity.
snmpEnableAuthenTraps	read-write	Indicates whether the SNMP agent processing is able to create alarm messages due to authorization errors. The value of this object overwrites all configuration data. It thus provides a way of deactivating all alarm messages that result from authorization errors. Input: enabled (1) - unauthorized access cause a trap to be sent. disabled (2) - unauthorized access does not cause a trap to be sent.
snmpSilentDrops	read-only	Total number of GetRequest-PDUs, GetNextRequest-PDUs, GetBulkRequest-PDUs, SetRequest-PDUs and InformRequest-PDUs forwarded to the SNMP entity, which were lost unnoticed because the scope of reply of the corresponding GetResponse-PDU that contained an empty list of variable bindings violated a local restriction or was larger than the agreed message length for the sender of the Get...Request
snmpProxyDrops	read-only	Total number of GetRequest-PDUs, GetNextRequest-PDUs, GetBulkRequest-PDUs, SetRequest-PDUs and InformRequest-PDUs forwarded to the SNMP entity, which were lost unnoticed because message transfer to the proxy server failed due to a timeout, with the result that no GetResponse-PDU could be returned.

SNMP group

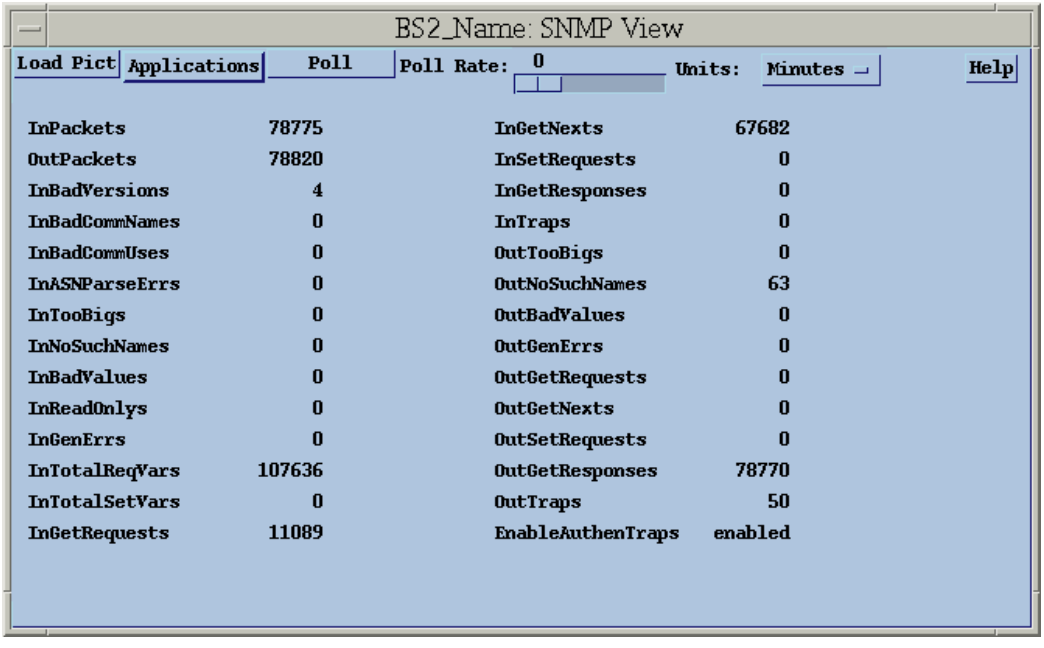


Figure 13: Display of the MIB-II values of the system group

5.1.3 SNMP framework MIB (SNMP engine)

Object name	Access	Meaning
snmpEngineID	read-only	Unique administrator name for the SNMP engine
snmpEngineBoots	read-only	Number of (re-)initializations of the SNMP engine since its start configuration
snmpEngineTime	read-only	Number of seconds since the last incrementation of the *snmpEngineBoots* object by the SNMP engine
snmpEngineMaxMessageSize	read-only	Maximum length (in bytes) of an SNMP message, which can be sent / received and processed by this SNMP engine.

SNMP Engine

5.1.4 Objects of other MIBs supported by the master agent

Besides the MIBs for the system and SNMP management, the master agent also supports the other objects of the MIBs relevant for SNMP management in BS2000/OSD listed below.

Standardized MIBs

SNMP-MPD-MIB	RFC 2272	snmpModules.3
SNMP-TARGET-MIB	RFC 2273	snmpModules.12
SNMP-NOTIFY-MIB	RFC 2273	snmpModules.13
SNMP-USER-BASED-SM-MIB	RFC 2274	snmpModules.15
SNMP-VIEW-BASED-ACM-MIB	RFC 2275	snmpModules.16

Private MIBs

SR-COMMUNITY-MIB	snmpResearchMIBs.33
TGT-ADDRESS-MASK-MIB	snmpResearchMIBs.36
HTTPSEC-MIB	srExperimentalMIBs.1

5.2 SNMP management for subagents (supervisor subagent)

The Supervisor Subagent is used to monitor all the subagents logged on at the master agent.

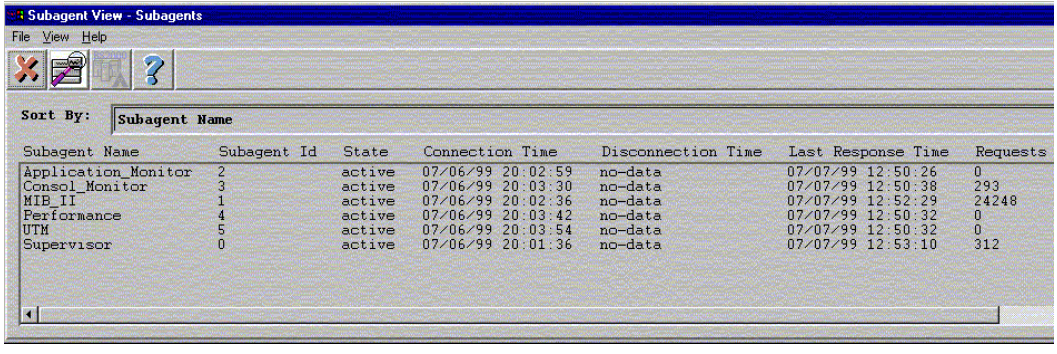


Figure 14: Subagent monitoring

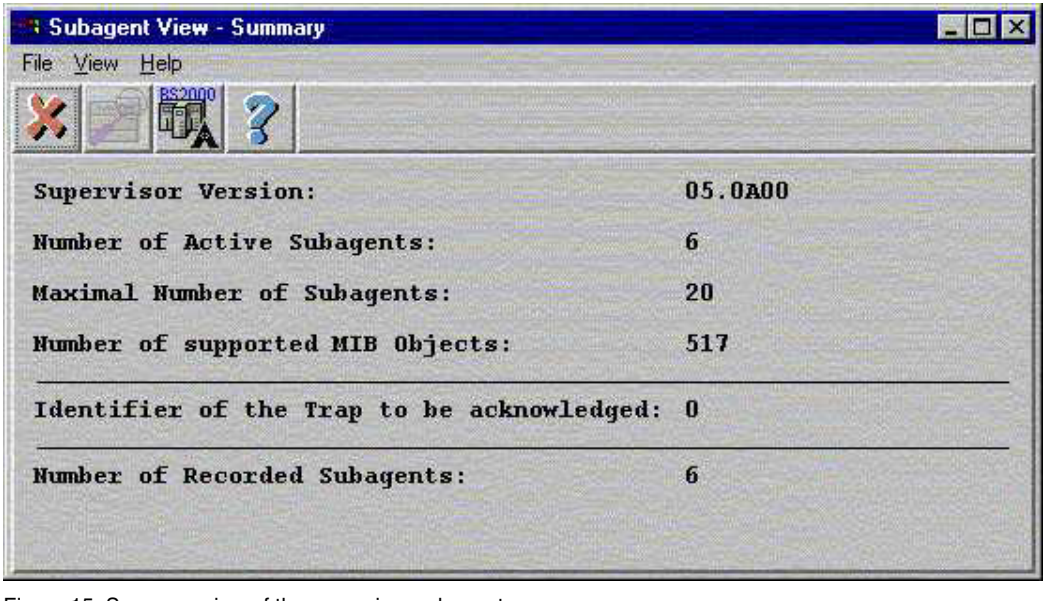


Figure 15: Summary view of the supervisor subagent

Object name	Access	Meaning
superVisVersion	read-only	Version of supervisor subagent
superVisActiveNumber	read-only	Number of active subagents
superVisMaxSubagent Number	read-only	Maximal number of subagents
superVisObjectNumber	read-only	Number of objects currently supported by the BS2000 SNMP agent
superVisTrapAckId	read-write	ID of last trap to be acknowledged by manager. Setting this object with its current value means acknowledgment of the last trap
superVisSubagentNumber	read-only	Number of entries in subagent table
Subagent Table		
superVisSubagentName	read-only	Name of the subagent (for BS2000-Subagents only)
superVisSubagentSID	read-only	SID of subagents: Index used by master agent for the management of the subagent
superVisSubagentStatus	read-only	Status of subagent: active (1): The subagent is connected and works normally. disconnected (2): The subagent has sent a disconnect event. It is no longer available. undefined (3): No answer from the subagent has been received since a request has been sent after 5 minutes. There was no disconnect event sent by this subagent.
superVisSubagentConnTime	read-only	Time when subagent was connected last
superVisSubagent Disconn Time	read-only	Time when subagent was disconnected last
superVisSubagentLast ResponseTime	read-only	Time when subagent last answered
superVisSubagentRequests Done	read-only	Number of requests processed by the subagent
superVisSubagentTrapsSent	read-only	Number of traps caused by the subagent
superVisSubagentOID	read-only	First object identifier supported by the subagent
superVisSubagentProcessID	read-only	Process identifier of the subagent
superVisSubagentUserId	read-only	User ID that belongs to the process identifier

Object name	Access	Meaning
superVisSubagentCpuTime	read-only	CPU time used by the subagent
superVisSubagentCommand	read-only	Command string found in the output of ps command that belongs to the process identifier

TrapAcknowledge group

Object name	Access	Meaning
superVisTrpAckState	read-write	Status of trap acknowledgment on agent: active(1): acknowledgment mechanism is active inactive(2): acknowledgment mechanism is not active undefined(3)
superVisTrpAckId	read-write	ID of last trap from manager requiring acknowledgment. If this object is set with the current value, the last trap is acknowledged.
superVisTrpAckQueueCnt	read-write	Number of traps currently queued. Any superfluous traps are removed from the queue.

Trap objects

Object name	Trap No	Meaning
Enterprise = 1.3.6.1.4.1.231.2.34.2		
superVisSubAgentConnected	301	The trap signifies that subagent has connected.
superVisSubAgent Disconnected	302	The trap signifies that master agent has disconnected.
superVisSubAgentNoAnswer	303	The trap signifies that subagent has not answered to a request for 5 minutes.

5.3 Application Monitor subagent

The Application Monitor subagent allows monitoring of:

– user applications
– BCAM applications
– DCAM applications
– subsystems
– job variables and
– log files

Logically associated components of a process (applications, log files, subsystems and job variables) can be monitored together as a group.

The term applications is taken to mean programs and tasks here. The type and extent of application monitoring is controlled individually with the configuration file. Please see the corresponding section on page 56 for information on configuration file creation.

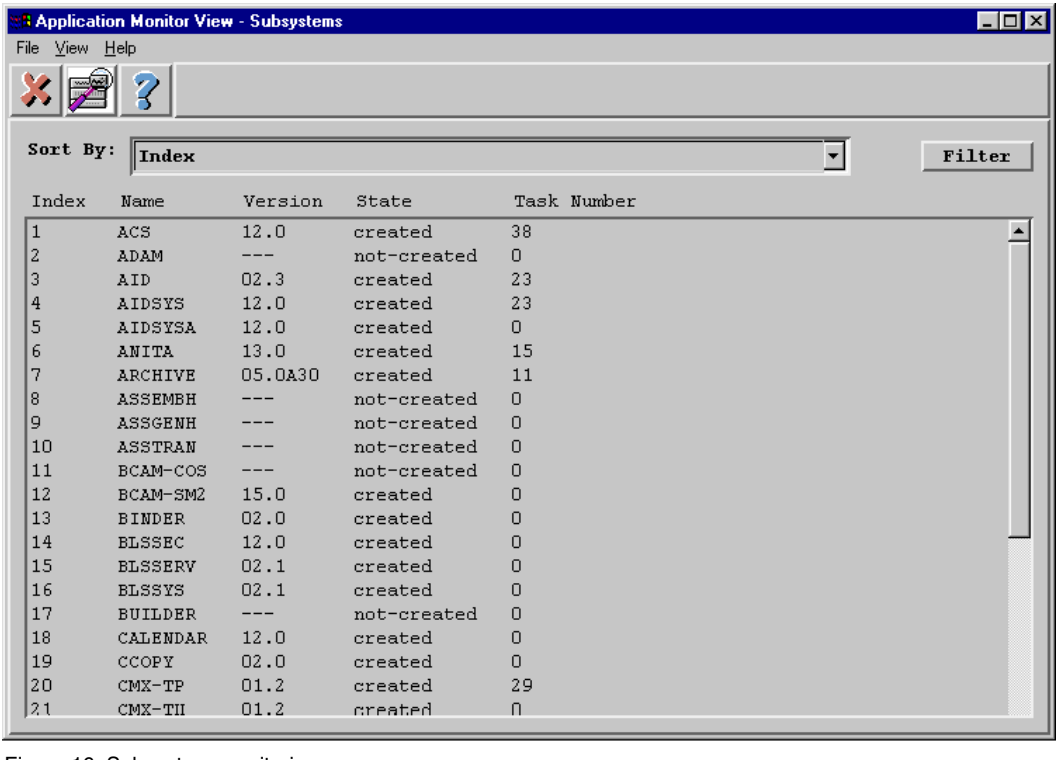


Figure 16: Subsystem monitoring

5.3.1 Private MIB of the Application Monitor subagent

The job variables software package is required for monitoring applications (programs and tasks) and job variables. The applications and tasks to be monitored must be entered in the configuration file. If the Application Monitor is started without a configuration file, only the subsystems are displayed.

Global data of the Application Monitor

The following objects display the global data of the Application Monitor:

MIB definition	Access	Meaning
appMonVersion	read-only	Version of the Application Monitor
appMonConfFile	read-write	Pathname of configuration file
appMonTrapFormat	read-write	Trap structure used

5.3.1.1 Trap structure

The Application Monitor subagent supports the following traps:

– Application Monitor-specific trap
– trap from the TCC-MIB (still supported for compatibility reasons)

The Application Monitor-specific trap is supported by default.

Structure of the Application Monitor-specific trap

1. Community: appMonDevice or <standard>

2. Enterprise: 1.3.6.1.4.1.231.2.23.20.2 (trap need not be acknowledged)
or
1.3.6.1.4.1.231.2.23.20.3 (trap must be acknowledged)

3. Trap number: 6 / <weight>

4.Variable binding:

appMonSource	1.3.6.1.4.1.231.2.23.20.1.1 (OCTET STRING)
appMonDevice	1.3.6.1.4.1.231.2.23.20.1.2 (OCTET STRING)
appMonMsg	1.3.6.1.4.1.231.2.23.20.1.3 (OCTET STRING)
appMonWeight	1.3.6.1.4.1.231.2.23.20.1.4 (INTEGER)

appMonSource, appMonDevice and appMonMsg depend on the type of application being monitored (see page 182 - page 191).

Structure of the trap from the TV-CC-MIB:

1. Community: <device/community>

2. Enterprise: 1.3.6.1.4.1.231.2.14

3. Trap number: 6 / 1

4.Variable binding:

tccTrapString: 1.3.6.1.4.1.231.2.14.1.1.3 (OCTET STRING)

"$DATE$: <date> $HOST$: <system> $SOURCE$: <source> $DEVC$: <device> MSG: <msg>"
<source> and <device> depend on the relevant data in the configuration file. <msg> depends on the type of application. See sections below!

tccApplTrapAck: 1.3.6.1.4.1.231.2.14.1.1.2.1.6.1 (INTEGER)

0 the trap does not have to be confirmed.
1 the trap must be confirmed.

The value also depends on the specifications in the configuration file.

5.3.1.2 Monitoring the BCAM and user applications

BCAM group

MIB definition	Access	Meaning
appMonBcamApplTabNum	read-only	Number of elements in the table
appMonBcamApplIndex	read-only	Index
appMonBcamApplName	read-only	Name
appMonBcamApplVersion	read-only	Version
appMonBcamApplState	read-only	Status*
appMonBcamApplMonJV	read-only	Name of the monitor job variable

*Please refer to the table on the next page for the corresponding status values.

User application group

MIB definition	Access	Meaning
appMonUserApplTabNum	read-only	Number of elements in the table
appMonUserApplIndex	read-only	Index
appMonUserApplName	read-only	Name
appMonUserApplVersion	read-only	Version
appMonUserApplState	read-only	Status*
appMonUserApplMonJV	read-only	Name of the monitor job variable

*Please refer to the table on the next page for the corresponding status values.

State of the separate BCAM and user applications

Value	Meaning
running	The application is running.
terminated	The application was terminated normally.
aborted	The application was terminated abnormally.
scheduled	The task is still in the wait loop.
unknown	The current state of the application cannot be determined.

The change of state of an application can lead to a trap being sent to the management station, depending on the trap conditions defined for the application in the configuration file.

Variable binding

With Application Monitor-specific trap format

appMonSource	appMonDevice	appMonMsg
"BS2-MonJV"	<appl-name>	"Application has entered state: <jv-contents>"

With TV-CC-MIB trap format

The trap string has the following structure:

Trap string
...$SOURCE$: <appl-name> $DEVC$: MSG: Application has entered state: <jv-contents>

5.3.1.3 Monitoring of DCAM applications

The following values are supplied for DCAM applications:

MIB definition	Access	Meaning
appMonDcamApplTabNum	read-only	Number of table elements
appMonDcamApplIndex	read-only	Index
appMonDcamApplName	read-only	Name
appMonDcamApplHost	read-only	Processor
appMonDcamApplState	read-only	Status*

*The corresponding status values are given in the table below.

Status of the individual DCAM applications

Value	Meaning
running	DCAM application is running.
terminated	DCAM application terminated normally.
unknown	Unable to determine the current status of the DCAM application.

The change of state of a DCAM application can lead to a trap being sent to the management station, depending on the trap conditions defined for the application in the configuration file.

Variable binding

With Application Monitor-specific trap format

appMonSource	appMonDevice	appMonMsg
"BS2-DCAM"	<appl-name>	"DCAM application no longer available. Reason <fdb1> <fdb2>"*

With TV-CC-MIB trap format

The trap string has the following structure:

Trap string
...$SOURCE$: <appl-name> $DEVC$: MSG: application no longer available. Reason <fdb1> <fdb2>*

*) <fdb1> and <fdb2> correspond to the first two bytes of the DCAM return code (see page 457).

5.3.1.4 Monitoring subsystems

The following values are supplied for subsystems:

MIB definition	Access	Meaning
appMonSubsysTabNum	read-only	Number of table elements
appMonSubsysIndex	read-only	Index
appMonSubsysName	read-only	Name
appMonSubsysVersion	read-only	Version
appMonSubsysState	read-only	Status*
appMonSubsysTasks	read-only	Number of tasks

*The corresponding status values are given in the table of the next page.

The version display is dependent on the entries in the configuration file. If information is requested about one or all subsystems without explicit version information, the highest version is displayed per defined subsystem (state not equal to “not-created”). If all versions of a subsystem are in the “not-created” state, an entry without version is returned. If the information for a subsystem is requested with version data, the display is independent of the state.

Subsystem states

Value	Meaning
created	The specified subsystem is loaded and initialized.
not-created	The specified subsystem is declared but currently not activated.
in-delete	The specified subsystem is in the deactivation process, after a STOP-SUBSYSTEM command. Processes connected to the subsystem will be completed.
in-create	The specified subsystem is currently in the loading and initialization phase.
in-resume	The specified subsystem is in the resume phase after a RESUME-SUBSYSTEM command. Reinitialization has not been completed.
in-hold	The specified subsystem was stopped by a HOLD-SUBSYSTEM command. the deinitialization is not completed. Processes connected to the subsystem will be completed.
not-resumed	The specified subsystem was stopped by a HOLD-SUBSYSTEM command. Deinitialization has been completed.
locked	A non-recoverable error has occurred while the specified subsystem was active or was activated, deactivated, resumed or stopped. A further attempt to execute the corresponding commands will be rejected.
unknown	The specified subsystem or version does not exist.

Subsystem state information can be requested from the management station. A trap is sent if the subsystem changes to the state defined by the TRAP-CONDITION. By default, the subsystems are checked by the subagent every 25 seconds.

Variable binding

With Application Monitor-specific trap format

appMonSource	appMonDevice	appMonMsg
"BS2-Subsys"	<subsystem>	"Subsystem has entered state: <state>.

With TV-CC-MIB trap format

The trap string has the following structure:

Trap string
...$SOURCE$: <ss-name(vers)> $DEVC$: MSG: Subsystem has entered state: <state>

5.3.1.5 Monitoring job variables

The job variables to be monitored must be made known in the configuration file using ADD-JV-RECORD (see page 64).

The following values are supplied for each job variable:

MIB definition	Access	Meaning
appMonJVName	read-only	Name of the job variable
appMonJVAppl	read-only	Name of the application
appMonJVValue	read-only	Current value of the job variable
appMonJVPattern	read-only	Pattern for which a trap is to be sent

If the contents of a job variable change, a trap is created. The trap contains the date, host and application name (if specified), and a message indicating the job variable state.

Variable binding

With Application Monitor-specific trap format

appMonSource	appMonDevice	appMonMsg
"BS2-JV"	<appl-name>	"Job variable has changed to: <jv-contents>"

With TV-CC-MIB trap format

The trap string has the following structure:

Trap string
...$SOURCE$: <appl-name> $DEVC$: MSG: Jobvariable has changed to: <jv-contents>

5.3.1.6 Log file monitoring

Monitoring via log files is provided for those applications that cannot send a trap to the management station themselves. Instead of this, these applications deposit messages in a log file that is monitored by the Application Monitor. The Application Monitor subagent evaluates these messages and sends each entry to the management station as a trap.

BS2000 log files must be of type ISAM and must be shareable (SHAREUPD=YES). NFS or POSIX log files can be in ASCII or EBDIC format. The default format is EBDIC, ASCII format must be marked in the configuration file. The file name entry in the configuration file must contain the user ID for BS2000 or the absolute path name in the case of NFS / POSIX. The subagent cannot otherwise discriminate between BS2000 and NFS / POSIX files.

The following objects can be displayed for each monitored log file:

MIB definition	Access	Meaning
appMonLogfName	read-only	Path name of the file
appMonLogfAppl	read-only	Name of associated application
appMonLogfState	read-write	Indicates whether or not monitoring is active
appMonLogfPattern	read-only	Pattern for which a trap is to be sent

The *appMonLogfState* object can also be set:

Value	Meaning
deactive	Terminates monitoring
start-begin	Activates monitoring at the start of the file
start-new	Activates monitoring at the start of the file without taking account of the original contents
start-end	Activates monitoring at the end of the file

By default, log files are checked every 5 seconds by the subagent. This value can be changed in the start command using the TIMER-INTERVAL operand, and with the *appmoncmd* command during operation. If the subagents detect changes to the files, a trap is sent to the management station for each new message.

Variable binding

The structure of this trap string depends on the contents of an entry in the log file. If a new message begins with `$<DEVICE=devc>$`, the Application Monitor subagent uses <devc> for Device.

With Application Monitor-specific trap format

appMonSource	appMonDevice	appMonMsg
"BS2-LogF"	<appl-name>	<logfile entry>
"BS2-LogF"	<devc>	<logfile entry>

With TV-CC-MIB trap format

DEVC is assigned <devc> in the trap string itself.

Entry in the file	Trap string
$<DEVICE=devc>$ text	...$SOURCE$: <appl-name> $DEVC$: <devc> MSG: text
text	...$SOURCE$: <appl-name> $DEVC$: MSG: text

5.3.1.7 Controlling file monitoring

The /START-APPMONCMD (BS2000/OSD) or appmoncmd (POSIX) command offers the following options for file monitoring in addition to the standard options:

i	print information	Output the information
a	activate monitoring	Activate the monitoring
d	deactivate monitoring	Deactivate the monitoring
p	set time period [sec]	Define the time period

Options for file monitoring

Option	Meaning
[**i**]	Outputs information on the log files to stderr.
[**a**] <logfile-id> <position>	Activates the monitoring of the log file with the ID <logfile-id>. <position> positions the file pointer in the file: b Start of file p Last read position if monitoring was already active e End of file Monitoring is activated only temporarily, the configuration file is not changed.
[**d**] <logfile-id>	Monitoring is deactivated only temporarily, the configuration file is not changed.
[**p**] <seconds>	Defines the cycle (in seconds) for checking the log files.

Example:

```
ID APPL      S FSYS PATHNAME              FILE    ERR  SIZE MTIME
1  ROBAR.025 A  ISAM $SYSROBAR.LOGFILE3   open      0    78 15:41:05 11.Oct(UTC)
2  APPL1     A  UFSE /tmp/logfile4        closed    8     0 unknown
3  APPLICATI D  ISAM $TSOS.APPL2.LOGFILE  closed    0     0 unknown
4  unknown   A   UFS /home/snmp/Logfile2  open      0   196 16:42:35 12.Oct
5  SNMP      A  ISAM :2OS6:$DC14.SNMP.LOG open      0   335 17:48:57 12.Oct(UTC)
```

Output	Meaning
ID	The Application Monitor subagent assigns a number to each log file. This ID is specified with activation/deactivation of file monitoring.
APPL	The first nine characters of the application name.
S	Monitoring state: "A" for activated and "D" for deactivated.
FSYS	File format: "UFS" (Unix File System, ASCII file) "UFSE" (Unix File System, EBCDIC file) "ISAM" (BS2000 file)
PATHNAME	The first 20 characters of the log file name.
FILE	File state: possible values are open or closed.
ERR	Error number, if an error occurred during file access. See /usr/include/sys/errno.h for meaning.
SIZE	File size in bytes
MTIME	Time and day of the last change to the file Note: the UTC is output for BS2000 file (identification: (UTC)).

5.3.1.8 Monitoring groups of associated elements

Logically associated components of a process (applications, log files, subsystems and job variables) can be combined into a group (object) and monitored together (DEFINE-OBJECT, see page 66).

All elements (applications, subsystems, etc.) combined to form an object must be configured with the appropriate instructions in the configuration file.

The following values are supplied for objects:

MIB definition	Access	Meaning
appMonObjectName	read-only	Name of the object
appMonObjectBcamAppl	read-only	Name of all BCAM applications belonging to this object
appMonObjectUserAppl	read-only	Name of all user applications
appMonObjectDcamAppl	read-only	Name of all DCAM applications
appMonObjectSub	read-only	Name of all subsystems
appMonObjectLogfile	read-only	Name of all log files
appMonObjectJV	read-only	Name of all job variables

If an event occurs for which the Application Monitor Subagent sends a trap to the Management Station, the trap contents are structured as described, with the following deviations:

– For Application Monitor-specific trap, appMonSource is supplied with BS2-Object: <object> in the variable binding.

– In the trap string of the TV-CC trap, DEVC is assigned the object name:

Trap string
.... $SOURCE$: ... $DEVC$:<obj-name>MSG: ...

5.4 Console Monitor subagent

The Console Monitor subagent is the subagent for monitoring the console interface. It is used for acquiring console messages and entering console commands. The Console Monitor is assigned its own management application, SMAWcmbs2 (Solaris) or CMBS2 (Reliant UNIX, Windows NT). Please refer to the description on page 348 if you wish to use SMAWcmbs2 or CMBS2.

5.4.1 Acquiring console messages

Console messages are received by the Console Monitor subagent and sent to the management station singly with the computer name and time as trap. The number of messages you will have to deal with will depend on the number, utilization and size of the computers monitored by the Console Monitor subagent. It will, however, very seldom be meaningful to pass all console messages on to the management station. The Console Monitor subagent therefore provides two options for filtering console messages. Positive and negative message filters are provided.

positive message filter

1. Each console message is assigned a specific routing code. You define the messages to be output on the management station by selecting specific routing codes which are defined in operator roles.
2. The message codes of console messages, query or TYPE I/Os can also be used as selection criteria to define which messages are sent to the management station. The relevant message codes are stored in a message filter file and this is evaluated by the Console Monitor subagent when it is started and during a session if the file is updated

negative message filter

The Console Monitor subagent allows you to suppress certain messages when logging on to UCON.

Creating the message filter file, which must be specified when the subagent is started, is described on page 69. The message filter file can be modified during a session by writing the *consMonMsgFilter* MIB object (positive message filter) and via the command program; the negative message filter *consMonNegMsgFilter* cannot be modified during operation.

If the newly assigned message filter file cannot be opened, it is rejected with the return code *General error* and the old file is used. No traps are sent to the management station if the message filter file contains either no message codes or no valid ones. The negative message filter *consMonNegMsgFilter* cannot be modified during a session.

The Console Monitor subagent MIB contains the following internally used objects:

Object name	**Access**	**Meaning**
consMonVersion	read-only	Version of the Console Monitor
consMonMsgFilter	read-write	Name of the message filter file
consMonNegMsgFilter	read-only	Name of the negative message filter file
consMonCmdFreeIndex	read-only	Next free index in the command table
consMonCmdTabNum	read-only	Number of entries in the command table
Command table		
consMonCmdIndex	read-only	Command index
consMonCmd	read-write	BS2000/OSD console command
consMonCmdResult	read-only	Result of the BS2000/OSD console command
consMonCmdMainRetco	read-only	Main return code of the BS2000/OSD console command
Table of command outputs		
consMonOutCmdIndex	read-only	Command index of the associated command
consMonOutLineNo	read-only	Number of lines for a command
consMonOutContents	read-only	A line of the command input
consMonBS2Ans	read-write	Response to a BS2000/OSD console message

To operate the Console Monitor subagent, please use the management application SMAWcmbs2 or CMBS2, which is described on page 338.

Trap structure

The Console Monitor subagent supports the following trap formats:

– Application Monitor-specific (generic) trap

– TV Control Center trap (trap from the TV-CC-MIB)

You define the trap format you want to use for the Console Monitor in the message filter file of the Console Monitor subagent (see page 70, “Trap format selection criterion”).

The Application Monitor-specific trap is used by default.

Application Monitor-specific trap

The Application Monitor-specific trap is described on page 180. Variable binding is performed in accordance with the entries in the configuration file.

TV Control Center trap (TV-CC-MIB trap)

The structure of the TV Control Center trap string depends on the entries for SOURCE and DEVICE in the configuration file:

SOURCE=	**DEVICE=**	**Trap string**	**COMMUNITY**
<src>	<dev>	...$SOURCE$: BS2-<src> DEV: <dev> MSG:...	<dev>
---	<dev>	...$SOURCE$: BS2Console DEV: <dev> MSG:...	<dev>
<src>	---	...$SOURCE$: BS2-<src> DEV: MSG:...	*std
---	---	...$SOURCE$: BS2Console DEV: MSG:...	*std

5.4.2 Symmetrix monitoring

Symmetrix is a powerful disk control subsystem from EMC Corporation, which allows you to easily manage and monitor disk I/O units of various systems, whether mainframes or Open Systems. In addition to this, the additional functions provided by Symmetrix let you set up a corporate memory management concept.

Functionality

A BS2000 system receives event messages from each Symmetrix, which are then converted to console messages (NJD0010 to NJD0013). The Console Monitor subagent filters these messages and sends corresponding traps to the management station. The SMBS2 package contains elements for a simple monitoring mechanism, which signals an incoming console message to a Symmetrix event by changing the color in the network map. The monitoring function consists of the expansions of the TransView Control Center described on page 107 and the alarms described on page 344.

The Symmetrix events notified to the console are divided into four classes, corresponding to the four message codes:

NJD0010 Connection to the service processor or to the EMC Customer Support Center (CSC) lost. Symmetrix should still operate normally.

NJD0011 An error has occurred, which may cause a malfunction or loss of data.

NJD0012 An error has occurred, which can be corrected by a redundant function.

NJD0013 Symmetrix signals an event. This is in part the correction of a problem state.

The message text contains various additional information to describe the device affected and the type of event in greater detail:

– technical name of the device address formed from the channel, control and device.
– control type
– serial number of the control
– reference code as precise designation of the problem or event that occurred
– an indicator for repeat messages

Reference codes for messages NJD0010 - NJD0013

Reference code	Message and meaning	Message code	Components affected
460	*dynamic spare invoked* Spare disk activated	NJD0012	Disk
461	*resynchronization completed* Resynchronization between a disk and its mirror disk completed.	NJD0013	Disk
462	*resynchronization completed* Similar meaning to x461.	NJD0013	Disk
463	*dual initiator failed* The control has failed for a disk. The second control takes over.	NJD0012	Disk controller
464	*data migration for all volumes completed* File migration has been completed on all devices.	NJD0013	Disk
465	*resynchronization started* A disk resynchronization has been started. The message generally appears after message x460.	NJD0013	Disk
466	*dynamic spare invoked for remote disk* A spare disk had to be activated at a partner Symmetrix. The message is received in parallel to message x460 from Symmetrix controls connected by SRDF.	NJD0012	Partner Symmetrix
467	*error/event posted by SRDF partner box* A partner Symmetrix has signalled an error or an event. This message is received in parallel from the Symmetrix controls connected by SRDF.	NJD0011	Partner Symmetrix
46D	*remote links not operational* All SRDF connections to the partner Symmetrix are lost. The connected disks can no longer be kept synchronized.	NJD0012	SRDF connection
46E	*all remote links operational again* All SRDF connections are operational again.	NJD0013	SRDF connection
470	*over temperature* The control is too hot and is about to fail.	NJD0011	Control
471	*low battery / high charge state* Problem with power supply and battery.	NJD0011	Control
472	*power subsystem alarm* Problem with power supply. The control is about to fail.	NJD0011	Control

Reference code	Message and meaning	Message code	Components affected
473	*local mirrored device not ready* A mirror disk has failed. Disk availability is restricted.	NJD0012	Disk
474	*local mirrored device write disabled* A mirror disk cannot be written. The problem is similar to x474, but the damage to the disk is more serious.	NJD0012	Disk
475	*remote mirrored device not ready* One of the problems x474 or x475 has occurred at a partner Symmetrix.	NJD0012	Partner Symmetrix
476	*service processor not responding* The service processor has failed. No more messages can be sent to the EMC Customer Support Center.	NJD0010	Service processor
477	*autocall (to EMC-CSC) failed* The service processor cannot connect to the EMC Customer Support Center.	NJD0010	Service processor
478	*12 V on* External problem with power supply	NJD0011	Control
479	*environment cable missing* Physical external connection not available.	NJD0011	Control
47A	*AC line failure/interruption* Failure in AC power supply.	NJD0011	Control
47B	*battery, clock, director without power* Problem with the power supply of individual components.	NJD0011	Control
47C	*latched alarm* Problem with control environment.	NJD0011	Control
47D	*remote link not operational* One of the SRDF connections has failed. The disks connected there can no longer be synchronized.	NJD0012	SRDF connection
47E	*remote link(s) operational again* An SRDF connection that has failed is operation again.	NJD0013	SRDF connection

Example

The example below illustrates the conversion of console message NJD0013 with reference string 46E into a trap string:

```
% P26-000.144244  %  NJD0013 -INFORMATION- #EZM  5100
MT=3860-43 SER=03-00434 REFCODE=146E-1E134-0000
- READ HELP TEXT FOR DETAILLED INFORMATION ABOUT REFCODE
```

Trap string:

```
$DATE$: Feb 16 14:42:50 $HOST$: D016ZE07 $SOURCE$: BS2-SYMMETRIX $DEVC$:
 Symmetrix $MSG$: <000> % P26-000.144244  %  NJD0013 -INFORMATION- #EZM
 5100 MT=3860-43 SER=03-00434 REFCODE=146E-1E134-0000 - READ HELP TEXT FOR
 DETAILLED INFORMATION ABOUT REFCODE
```

The the event manager screen shot below shows the first line of the error message with reference code 46E:

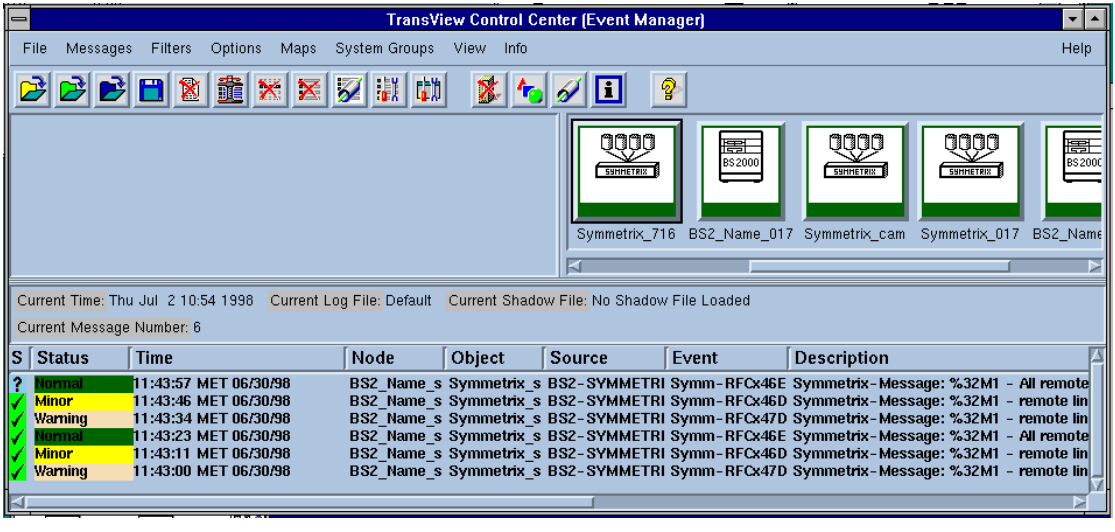


Figure 17: Symmetrix monitoring

5.5 Custom pages (HTML subagent)

The HTML subagent is required for processing custom DR-Web pages (custom pages, see pages 410 and 419) using SNMP requests. The information about custom DR-Web pages is stored in the variables of the HTML-MIB.

The HTML-MIB contains the following groups and tables:

– htmlGlobals group
– htmlPages group
– htmlPageTable
– htmlPageParameterTable
– htmlPageContentTable

htmlGlobals group

MIB definition	Access	Meaning
htmlConsistencyCheck	read-write	*htmlConsistencyCheck* is used to clear the memory of non-referenced lines in the *htmlPageParameterTable* and *htmlPageContentTable*. When the variable is read, a value *consistent* (1) is always returned. Setting the variable to the value *verify* (2) initiates the check.

htmlPages group

The objects in the *htmlPages group* contain information about the custom pages supported by the BS2000/OSD web agent (DR-Web Entity). Access to custom pages is possible via the menu page of the DR-Web interface (see page 399).

MIB definition	Access	Meaning
htmlPageSetSerialNo	read-write	*htmlPageSetSerialNo* implements a lock integrated in the HTML-MIB, which enables numerous management stations to cooperate with each other and avoid mutual interference.

htmlPageTable

The table htmlPageTable contains general (meta)information about the custom pages. Each custom page that exists on a hyperlink in the DR-Web menu page is assigned to a line in the htmlPageTable.

MIB definition	Access	Meaning
htmlPageTitle	read-create	*htmlPageTitle* specifies the title of the associated HTML page. The text in this display string object is added to the HTML page between the tags <title>...<\title>. A string of length 0 means that the custom page does not have a title.
htmPageAddressInfo	read-create	*htmlPageAddressInfo* specifies the address information of the associated HTML page. The text in this display string is added to the HTML page between the tags <address>...<\address>. A string of length 0 means that the custom page does not contain address information.
htmlPageLastUpdated	read-create	*htmlPageLastUpdated* specifies the time of the last change too the HTML page. This object has no special format.
htmlPageBodyArgs	read-create	*htmlPageBodyArgs* contains the arguments for the <body> tag. For example, this display string object can have the value bgcolor='EFEFEF', which corresponds to the <body> tag <body bgcolor='#EFEFEF'. The default value of *htmlPageBodyArgs* is the string of length 0. *htmlPageBodyArgs* is also used to set the refresh time (see page 406).
htmlPageOwner	read-create	*htmlPageOwner* specifies the owner of the custom page. This object helps to coordinate access to the custom page from several management platforms. Any data can be stored in *htmlPageOwner*. An instance of an object should contain at least the snmpID of the SNMP manager and the user ID. When creating a new table line or modifying a existing line, the manager should use the *htmlPageSetSerialNo* object of the *htmlPages* group to control access to the entire *htmlPageTable*.
htmlPageStorageType	read-create	*htmlPageStorageType* specifies how the associated table line is to be stored. This object is of the type StorageType and described as TEXTUAL-CONVENTION in RFC 1903.

MIB definition	Access	Meaning
htmlPageStatus	read-create	*htmlPageStatus* contains the status of the instance in the *htmlPageTable*. This object is of type *RowStatus* and is described as TEXTUAL-CONVENTION in RFC 1903. Note that deleting a line does not affect the associated lines in the *htmlPageTable* and *htmlPagContentTable*. Rather, the memory is cleared using the *htmlConsistencyCheck* of the *htmlGlobals*-group.

htmlPageParameterTable

The objects of the *htmlPageParameterTable* contain information about the parameters that can be referenced by the contents of the associated custom pages. The *htmlPageParameterTable* is found via the name *htmlPageName* of the custom page that references the parameter.

MIB definition	Access	Meaning
htmlPageParameterName	read-create	Name of the parameter in the HTML page
htmlPageParameterDefault	read-create	Default value for a particular parameter in a particular HTML page
htmlPageParameterStorageType	read-only	*htmlPageParameterStorageType* how the associated table line is to be stored. This object is of the type StorageType and described as TEXTUAL-CONVENTION in RFC 1903.
htmlPageParameterStatus	read-only	Status of the associated instance in *htmlPageParameterTable.*

htmlPageContentTable

The objects of *htmlPageContentTable* contain information about the contents of the custom pages. The *htmlPageContentTable* is found via the name of the custom page specified by *htmlPageName* that references the parameter.

MIB definition	Access	Meaning
htmlPageContentIndex	not-accessible	Index for a text section to be displayed on an HTML page. Note that the index values in *htmlPageContentTable* do not have to be ordered sequentially. For example, table lines may be incremented in steps of 5 or 10. This makes sense because the space gained from the next non-assigned index values can be used for additional lines. This makes it easier to modify and add to a custom page.
htmlPageContentText	read-create	*htmlPageContentText* specifies a text section to be displayed on a custom page.
htmlPageContentStorageType	read-create	*htmlPageContentStorageType* specifies how the associated table lines is to be stored.
htmlPageContentStatus	read-create	*htmlPageContentStatus* contains the status of the associated table line.

5.6 Trap security

The asynchronous notification of problems to the agent via traps is an extremely efficient procedure because it reduces the network load to a minimum. However, a problem is posed by the fact that information is lost if no management station is active at the time a trap is sent or there is fault in the communication to it. The concept of trap acknowledgment described here brings a considerable improvement in this behavior.

A trap declared to be *to be acknowledged* is assigned internal information. The management station detects this information and automatically sends a set request to the agent. If the agent receives the request, the trap is deemed to be acknowledge.

This concept requires a TransView Control Center Version 4.3 or higher in addition to the version of the agent described in this manual. The Console Monitor application can also confirm traps (see page 350).

Only traps of the Console Monitor subagent and of the Application Monitor subagent, which are sent in Application Monitor-specific format can be declared to be *to be acknowledged*.

Functionality

Traps that are to be acknowledged are sent in a strictly sequential order, i.e. a trap that is to be acknowledged is not sent by an agent until the acknowledgment for the previous trap has been received.

Traps that are not be acknowledged are sent in each case.

Traps that are still waiting to be acknowledged are stored temporarily in the agent.

If the acknowledgment is not received within the tolerance period of 30 seconds, an information trap with the message

```
<number> messages left in SNMP Master-Agent agent buffer;
```

is sent. Here, <number> denotes the number of traps that have not been acknowledged or could not be sent because the acknowledgment was not received. This information trap is also a trap that must be acknowledged. Unlike a user trap, an information trap is not buffered. It is repeated at 90-second intervals until acknowledged. This serves to check the integrity of the communication periodically.

If an acknowledgment is received by the agent, the buffered traps are sent again in their original format, i.e. with the *to be acknowledged* flag.

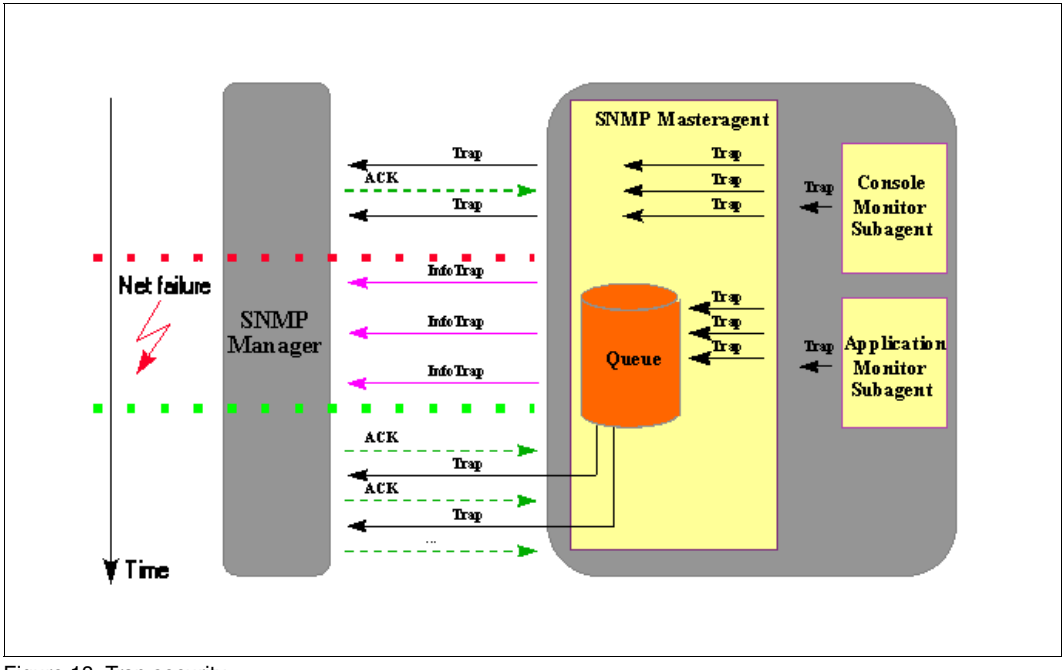


Figure 18: Trap security

Configuration

The traps to be confirmed b the manager are defined

– in the Console Monitor by specifying `ACKNOWLEDGE=YES` in the message filter file (see page 73) and

– in the Application Monitor by specifying `ACKNOWLEDGE=YES` in the records of the configuration file (see page 58).

This allows you to differentiate when specifying the acknowledgment down to the level of message numbers or applications.

Controlling the trap security

The trap security is activated on the BS2000 side as soon as the Supervisor subagent is started. You can control the entire acknowledgment process using the command program of the master agent and of the Console Monitor application SMAWcmbs2 (Solaris) or CMBS2 (Reliant UNIX) (see section “Trap confirmation window” on page 366):

/START-MASTERCMD	
x	agent executes command
getTrapAckState	1 or 2: Acknowledge activated 3 or 4: Acknowledge deactivated
startTrapAck	activated the acknowledgment procedure.
stopTrapAck	deactivates the acknowledgment procedure.

1. It is recommended that you only send a few important traps with the *to be confirmed* flag.
2. Confirmation by a management station is sufficient for the agent.
3. If none of the management stations is able to acknowledge, all the traps declared as *to be acknowledged* are deferred and placed in a buffer.

6 Functions of the STANDARD COLLECTION

The subagents in the STANDARD COLLECTION are described in this chapter. It includes functional descriptions and complete lists of the MIBs.

6.1 SNMP management for AVAS

The AVAS MIB is made up of four groups for the following tasks:

– information on basic data

– monitoring of the central process and run control systems

– monitoring the networks

– monitoring the network structure elements

Basic data

MIB definition	Access	Meaning
avasagtVersion	read-only	Version of the AVAS subagents
avasSystemID	read-only	AVAS system ID

Processes and process control

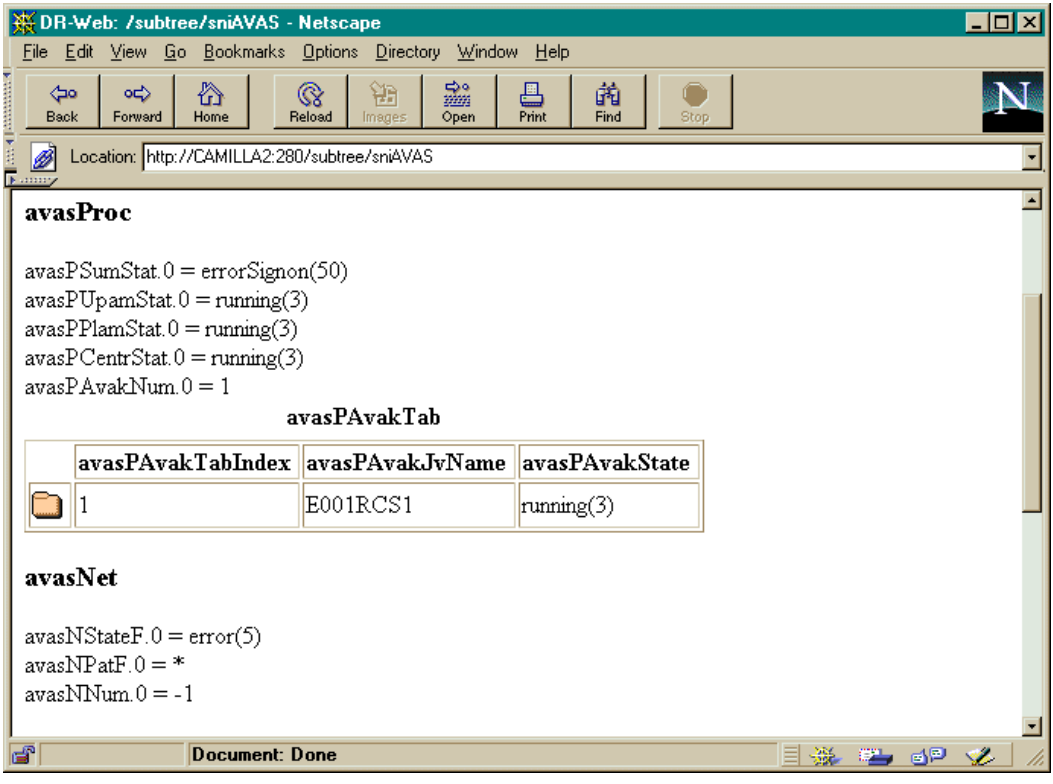


Figure 19: AVAS: Status of the central processes

Group for monitoring the central processes and schedules

MIB definition	Access	Meaning
avasPSumStat	read-only	System State AVAS*
avasPUpamStat	read-only	Process State UPAM-ZD
avasPPlamStat	read-only	Process State PLAM-ZD
avasPCentrStat	read-only	Central Process State
avasPAvakNum	read-only	Number of Run Control Systems
Table:		
avasPAvakTabIndex	read-only	Index
avasPAvakJvName	read-only	Name
avasPAvakState	read-only	Status

* Please refer to the table below for the corresponding values.

The AVAS system state is formed from a combination of the different process states, and supplies complete information on the state of the AVAS system.

Value	Meaning
missing	One of the UPAM-ZD or PLAM-ZD processes is in neither the "ready" nor the "running" state.
ready	The UPAM-ZD and PLAM-ZD processes are in the "ready" state.
running	The UPAM-ZD and PLAM-ZD processes are in the "ready“ state and at least 1 RCS is in the "running" state.
error-net	At least 1 network is in the "error" state.
error-system	One of the UPAM-ZD and PLAM-ZD processes has failed.
error-signon	Connection setup via the program interface has failed.

System status AVAS

Information is additionally supplied about the central process state, PLAM-ZD and UPAM-ZD process states and the number of run control systems. Please see the relevant AVAS manuals for information on interpreting the possible outputs.

Displaying the job networks

The network monitoring group supplies tabular information about AVAS networks. The network state or name are used as the selection criteria for the extent of the information supplied. If a network state is specified, only those objects that are in this state are displayed. The network name must be input as the selection criterion in uppercase and the entry can be terminated with an asterisk (*). This asterisk is the default setting for the network name. Entry of the user group is not required. Network state restrictions are ignored if entry of the network name is fully qualified.

Group for monitoring the AVAS networks

MIB definition	Access	Meaning
avasNStateF	read-write	Restriction by Net State
avasNPatF	read-write	Restriction by Net Name
avasNNum	read-only	Number of Nets
Table:		
avasNTabIndex	read-only	Table index
avasNName	read-only	Network name
avasNState	read-only	Network status*
avasNStateOfError	read-only	Network status switch : Error
avasNStateOfRestart	read-only	Network status switch : Restart
avasNStateOfCondwait	read-only	Network status switch : Condwait
avasNStateOfHold	read-only	Network status switch : Hold
avasNAvak	read-only	Schedule that belongs to the network

*The relevant values are given in the following table.

Please see the relevant AVAS manuals for information on interpreting the possible outputs.

State flag for restricting the AVAS networks display

Value	Meaning
problem	The networks that are in the “error”, “hold”, “running”, “waiting” or “condwait” state are displayed. This is the default value
error	The networks in the "error" state are displayed.
hold	The networks in the "hold" state are displayed.
running	The networks in the "running" state are displayed.
waiting	The networks in the "waiting" state are displayed.
condwait	The networks in the "condwait" state are displayed.

AVAS network state

Displaying the structure elements

The structure elements monitoring group supplies tabular information about AVAS structure elements. Possible selection criteria for the extent of the information supplied are as follows:

– network state
– network name
– element state
– element type or
– element function.

If a network state is specified, those elements whose networks are in this state are displayed. The network name must be input as the selection criterion in uppercase and the entry can be terminated with an asterisk (*). This asterisk is the default setting for the network name. Entry of the user group is not required. Network state restrictions are ignored if entry of the network name is fully qualified. If the selection criterion is the element state, information about elements that are in the specified state is returned (see table). If selection is made by element type, the elements of this type are displayed (default value = “all”). Selection can also be restricted to element function by entering this as the selection criterion (default value = “all”). Please see the relevant AVAS manuals for a detailed description of the element type and function values.

Group for monitoring AVAS structure elements

MIB definition	Access	Meaning
avasENStateF	read-write	Restriction by Net State [1)]
avasENPatF	read-write	Restriction by Net Name
avasEEStateF	read-write	Restriction by Element State [2)]
avasEEFuncF	read-write	Restriction by Element Function
avasEETypeF	read-write	Restriction by Element Type
avasENum	read-only	Number of elements
Table:		
avasETabIndex	read-only	Table index
avasEName	read-only	Name of the structure element
avasEFu	read-only	Function of the structure element
avasEType	read-only	Type of structure element
avasEInd	read-only	Index level of structure element
avasESynInd	read-only	Synchronization level of structure element
avasEState	read-only	Status of the structure element
avasENet	read-only	Associated network
avasEDelSolution	read-only	DELAY solution for the structure element
avasELatest	read-only	Latest start time

[1)] Please see table “Network status of AVAS structure elements” on next page.
[2)] Please see table “Element status of AVAS structure elements” on next page.

Please see the relevant AVAS manuals for interpretation of the entries.

Network status flag for restricting the display of AVAS structure elements

Value	Meaning
problem	Only those elements for which the network status is “error”, “hold”, “running” or “condwait” are displayed. This is the default setting.
error	Only those elements for which the network status is “error” are displayed
hold	Only those elements for which the network status is “hold” are displayed
running	Only those elements for which the network status is “running” are displayed
skipped	Only those elements for which the network status is “skipped” are displayed
waiting	Only those elements for which the network status is “waiting” are displayed
condwait	Only those elements for which the network status is “condwait” are displayed

Network status for AVAS structure elements

Element status flag for restricting the display of AVAS structure elements

Value	Meaning
all	There is not restriction via the status. This is the default setting
ended	Only those elements that have the state “ended” are displayed
error	Only those elements that have the state “error” are displayed
hold	Only those elements that have the state “hold” are displayed
running	Only those elements that have the state “running” are displayed
skipped	Only those elements that have the state “skipped” are displayed
waiting	Only those elements that have the state “waiting” are displayed
noOccure	Only those elements that have the state “noOccure” are displayed

Element status for AVAS structure elements

Traps

Object name	Trap No.	Meaning
avasLastMsg		Last Trap Message
avasStateTraps (Enterprise = 1.3.6.1.4.1.231.2.24.11.10)		
avasMissing	301	(UPAMZD || PLAMZD) not ready
avasReady	302	(UPAMZD && PLAMZD) ready
avasRunning	303	min1 RCS (ready || running)
avasErrorSystem	304	(UPAMZD || PLAMZD) abended
avasErrorNet	305	min1 net in error
avasErrorSignon	350	SIGNON != ok
avasProblemTraps (Enterprise = 1.3.6.1.4.1.231.2.24.11.11)		
avasNetAbended	311	Net abended
avasNetError	312	Net error
avasNetRestarted	313	Net Restarted
avasNetCancelled	314	Net Cancelled
avasJobAbended	321	Job abended
avasJobError	322	Job error
avasJobRestarted	323	Job restarted
avasJobCancelled	324	Job cancelled
avasProced	331	Procedure abended
avasProcError	332	Procedure error
avasProcRestarted	333	Procedure restarted
avasProcCancelled	334	Procedure cancelled
avasUJobAbended	341	Unix or NT Job abended
avasUJobError	342	Unix or NT Job error
avasUJobRestarted	343	Unix or NT Job restarted
avasUJobCancelled	344	Unix or NT Job cancelled

6.2 SNMP management for *open*FT (BS2000)

The file transfer subagent is used for

- starting and stopping *open*FT (BS2000)
- acquiring system parameter information
- changing the encryption public key
- statistic data output
- diagnosis control
- outputting partner information

The proprietary MIB for *open*FT (BS2000) offers objects for the above management tasks. The objects for starting and stopping, public-key encryption and diagnosis control also allow write accesses.

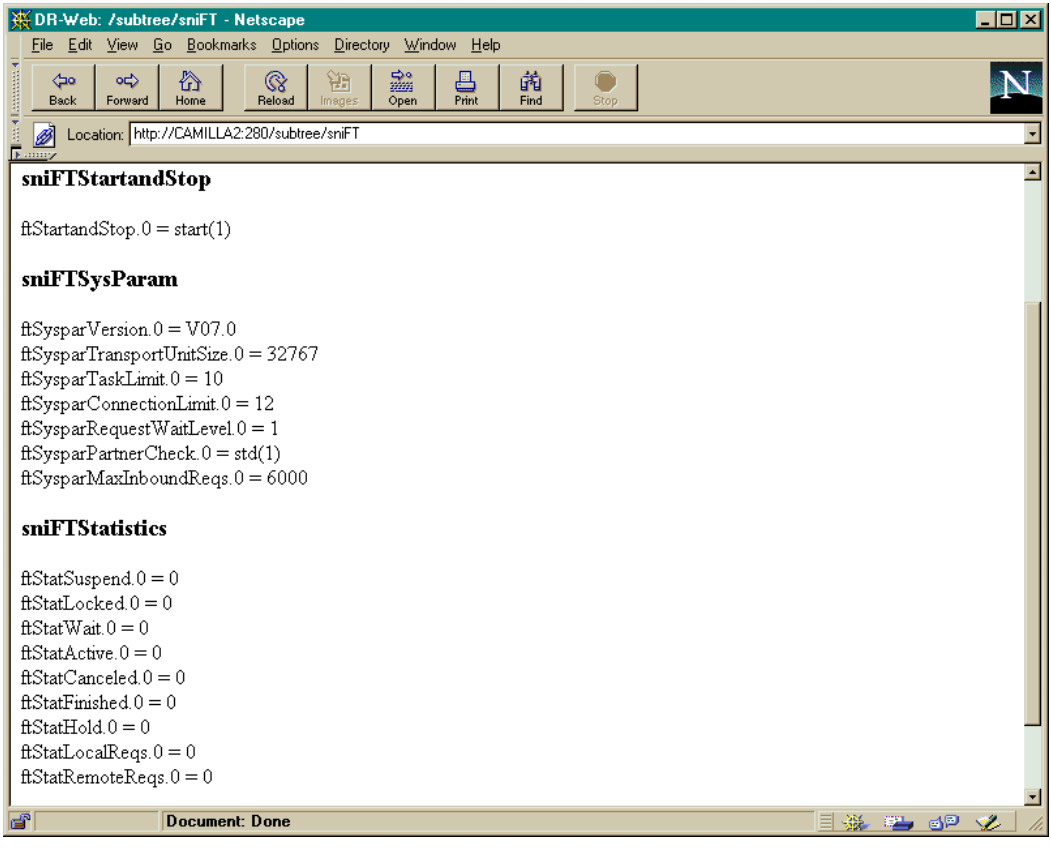


Figure 20: *open*FT subagent: overview

Starting and stopping FT

MIB definition	Access	Meaning
ftStartandStop	read-write	Start / Stop start (1) stop (2) on (3) off (4) undefined (255)

FT is started and stopped via the FT subagent by setting the value *START* or *STOP* respectively.

– Read access supplies information on the current state of FT (STARTED / NOT STARTED).

– With write access, FT can be started or stopped.

System parameters

MIB definition	Access	Meaning
ftSysparVersion	read-only	Version
ftSysparTransportUnitSize	read-write	Transport unit size
ftSysparTaskLimit	read-write	Task limit
ftSysparConnectionLimit	read-write	Maximum number of transport connections that can be reserved for executing FT requests.
ftSysparRequestWaitLevel	read-write	Number of waiting requests needed to set up a connection to the remote system (Request Wait Level)
ftSysparPartnerCheck	read-write	Partner check
ftSysparProcessorName	read-write	Processor name of the target system
ftSysparStationName	read-write	Station name of the target system
ftSysparMaxInboundReqs	read-write	Maximum number of inbound requests per partner system
ftSysparMaxLifeTime	read-write	Maximum lifetime (in days) in the request queue (maximum 400 days)

A description of the output values can be found in the *open*FT (BS2000) System Administrator Guide.

Public key for encryption

MIB definition	Access	Meaning
ftEncryptKey	write-only	The entry “create-new-key” or “1” causes a new public key to be created.

Statistical information

MIB definition	Access	Meaning
ftStatSuspend	read-only	Requests in status: SUSPEND
ftStatLocked	read-only	Requests in status: LOCKED
ftStatWait	read-only	Requests in status: WAIT
ftStatActive	read-only	Requests in status: ACTIVE
ftStatCanceled	read-only	Requests in status: CANCELED
ftStatFinished	read-only	Requests in status: FINISHED
ftStatHold	read-only	Requests in status: HOLD
ftStatLocalReqs	read-only	Async requests in local system
ftStatRemoteReqs	read-only	Requests in remote system

A description of the output values can be found in the *open*FT (BS2000) System Administrator Guide in the section dealing with the SHOW-FILE-TRANSFER command.

Diagnosis control

MIB definition	Access	Meaning
ftDiagStatus	read-write	Diagnosis Management

The following table shows the input options - syntax or integer - for FT diagnosis control:

Input:		Meaning for:		
Syntax	**Integer**	**FTNEA=**	**FTAM=**	**SESSION=**
off	1	TRACE=OFF - disables all FT traces		
snn	2	*STD	*NONE	*NONE
pnn	3	*BY-PARTNER	*NONE	*NONE
mnn	4	*MINIMUM	*NONE	*NONE
nnn	5	*NONE	*NONE	*NONE

FT diagnosis control inputs

Input:		Meaning for:		
Syntax	**Integer**	**FTNEA=**	**FTAM=**	**SESSION=**
ssn	6	*STD	*STD	*NONE
sns	7	*STD	*NONE	*STD
sss	8	*STD	*STD	*STD
psn	9	*BY-PARTNER	*STD	*NONE
pns	10	*BY-PARTNER	*NONE	*STD
pss	11	*BY-PARTNER	*STD	*STD
msn	12	*MINIMUM	*STD	*NONE
mns	13	*MINIMUM	*NONE	*STD
mss	14	*MINIMUM	*STD	*STD
nsn	15	*NONE	*STD	*NONE
nns	16	*NONE	*NONE	*STD
nss	17	*NONE	*STD	*STD
on	18	TRACE=ON - enables all FT traces		
The following entries store the traces in main memory:				
smnn	19	*STD	*NONE	*NONE
smsn	20	*STD	*STD	*NONE
smns	21	*STD	*NONE	*STD
smss	22	*STD	*STD	*STD
pmnn	23	*BY-PARTNER	*NONE	*NONE
pmsn	24	*BY-PARTNER	*STD	*NONE
pmns	25	*BY-PARTNER	*NONE	*STD
pmss	26	*BY-PARTNER	*STD	*STD
mmnn	27	*MINIMUM	*NONE	*NONE
mmsn	28	*MINIMUM	*STD	*NONE
mmns	29	*MINIMUM	*NONE	*STD
mmss	30	*MINIMUM	*STD	*STD

FT diagnosis control inputs

The most common inputs are “off” (1) and “sss” (8).
Please also read the section dealing with the MODIFY-FT-OPTIONS command in the *open*FT (BS2000) System Administrator Guide.

Partner information

MIB definition	Access	Meaning
ftPartnerName	read-only	Name of FT partner
ftPartnerType	read-only	FT protocol used by partner: openft (1), ftam (2)
ftPartnerState	read-write	Status of FT partner: act (1), inact (2), nocon (3), lunk (4), runk (5), adeact (6), ainact (7)
ftPartnerNetworkAddr	read-only	Layer 3 address of partner
ftPartnerTransportSel	read-only	Layer 4 address of partner
ftPartnerSessionSel	read-only	Layer 5 address of partner
ftPartnerPresentationSel	read-only	Layer 6 address of partner

Currently, a status change is only supported for one partner.

Traps

Object name	Trap no.	Meaning
Enterprise = sniFTTraps		
ftStopTrap	1	TRAP is sent if the file transfer was lost or was terminated with errors.
ftPartnerStateTrap	4	TRAP is sent if the status of the partner has changed.
ftPartnerUnreachableTrap	5	Partner is possibly unreachable.
ftStartTrap	6	TRAP is sent after *open*FT is started.
ftRequestQueueUpperLimitTrap	7	TRAP is sent if the queue of FT requests is filled to at least 85%.
ftRequestQueueLowerLimitTrap	8	TRAP is sent if at least 20% of the FT request queue is still free.
ftRequesrSuccessfulTrap	9	TRAP is sent if an FT request is executed successfully.
ftRequestErrorTrap	10	TRAP is sent if an FT request was terminated with errors.
ftSubsystemStartTrap	11	TRAP is sent if the FT subsystem was started.
ftSubsystemStopTrap	12	TRAP is sent if the FT subsystem was stopped.

*open*FT traps

Trap groups and trap control

The traps of the *open*FT subagent can be combined into groups that are represented by the following MIB objects. In this way, the transmission of traps can be permitted or suppressed for individual trap groups (trap state “on” or “off”):

– Specification of 2 (“on”): The traps of the respective group are sent.
– Specification of 1 (“off”): The traps of the respective group are not sent.

MIB definition	Access	Relevant traps
ftTrapsSubsystemState	read-write	– ftSubsystemStartTrap – ftSubsystemStopTrap
ftTrapsFTState	read-write	– ftStartTrap – ftStopTrap
ftTrapsPartState	read-write	– ftPartnerStateTrap
ftTrapsPartnerUnreachable	read-write	– ftPartnerUnreachableTrap
ftTrapsRequestQueueState	read-write	– ftRequestQueueLowerLimitTrap – ftRequestQueueUpperLimitTrap
ftTrapsTransSucc	read-write	– ftRequestSuccesfulTrap
ftTrapsTransFail	read-write	– ftRequestErrorTrap

Trap information

MIB objects that are sent with the traps are defined in the MIB of the *open*FT subagent.

MIB definition	Access	Meaning
ftRequestID	not-accessible	Transfer ID of the request
ftRequestInitiator	not-accessible	Initiator of the request: local (1), remote (2)
ftRequestPartnerName	not-accessible	Partner of the requestor
ftRequestUserID	not-accessible	User ID of the requester
ftRequestFileName	not-accessible	Name of the file to be transferred
ftRequestError	not-accessible	Error in the request

6.3 SNMP management for HIPLEX-AF

The HIPLEX-AF MIB contains information about the systems involved in a HIPLEX-AF network and the defined switchover units. Important events in the network, such as starting and stopping or switchover, are notified by traps.

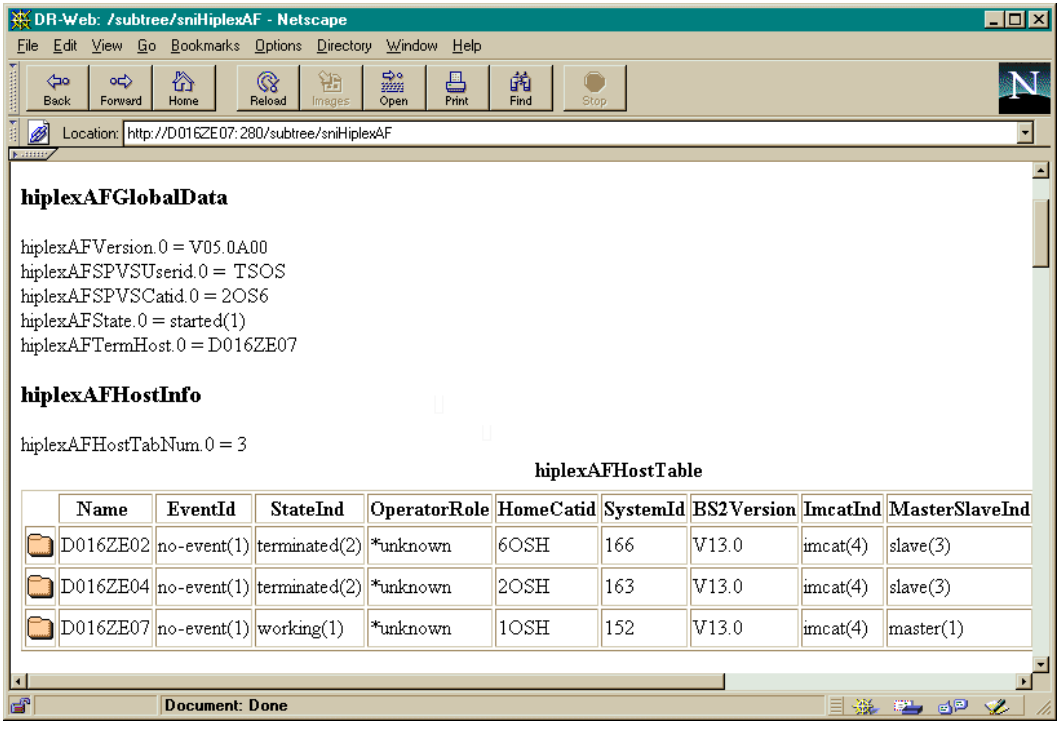


Figure 21: HIPLEX-AF: display of values for the individual object (GlobalData) and for the values of the host table

HIPLEX-AF single objects

Object name	Access	Meaning
hiplexAFVersion	read-only	Version of the HIPLEX-AF subagent
hiplexAFSPVSUserid	read-only	HIPLEX-A user ID (usually TSOS)
hiplexAFSPVSCatid	read-write	Catalog ID of the shared pubset that contains the job variables and files of HIPLEX-AF
hiplexAFStatus	read-write	Status of HIPLEX-AF: – *started*: At least one main procedure has been started on a host. – *stopped*: HIPLEX-AF terminated. No host is executing an HIPLEX-AF procedure. – *undefined*: The status of HIPLEX-AF is unknown. Unable to find a job variable with the correct values. It is possible to switch from status *started* to status *stopped*.
hiplexAFTermHost	read-only	BCAM name of the system on which the request to terminate HIPLEX-AF was initiated.

Host table

Object name	Access	Meaning
hiplexAFHostTabNum	read-only	Number of hosts in the availability cluster
Table:		
hiplexAFHostName	read-only	BCAM name of the system
hiplexAFHostEventId	read-only	Request indicator for the main procedure: – no-request – termination-requested – undefined
hiplexAFHostStateInd	read-write	Status of the main procedure that indicates the host involvement: – working – terminated – undefined Status *working* can only be reached from the status *terminated*.
hiplexAFHostOperatorRole	read-write	Status of the operator role parameter of the START-XAF commands
hiplexAFHostHomeCatid	read-only	Home Cat ID of the system
hiplexAFHostSystemId	read-only	System ID
hiplexAFHostBS2Version	read-only	BS2000/OSD version on the system
hiplexAFHostImcatInd	read-only	Status of the host: – check – crash – exact – imcat – mchange – readerr – shutdown – wrterr – undefined
hiplexAFHostMasterSlaveInd	read-only	Type host in the shared pubset network: – master – backup – slave – undefined
hiplexAFHostSnmpAgentStatusInd	read-only	Status of the SNMP subagent on the system: – working – not-working – undefined

Table for the switching units

Object name	Access	Meaning
hiplexAFSWUTabNum	read-only	Number of switching units in the availability cluster
Table:		
hiplexAFSWUName	read-only	Name of the switching unit
hiplexAFSWUCreaTime	read-only	Creation date of the switching unit
hiplexAFSWUWorkSystem	read-write	BCAM name of the system the is currently the switching unit for the work system
hiplexAFSWUVirtHost	read-only	BCAM name of the virtual host for the switching unit
hiplexAFSWUVirtHostAct	read-only	Name of the procedure for activating the virtual host
hiplexAFSWUVirtHostDeact	read-only	Name of the procedure for deactivating the virtual host
hiplexAFSWUFEPNumber	read-only	Number of front-end processors
hiplexAFSWUPubsetNumber	read-only	Number of data volume used by applications of the switching unit
hiplexAFSWUApplicationNumber	read-only	Number of applications contained in the switching unit

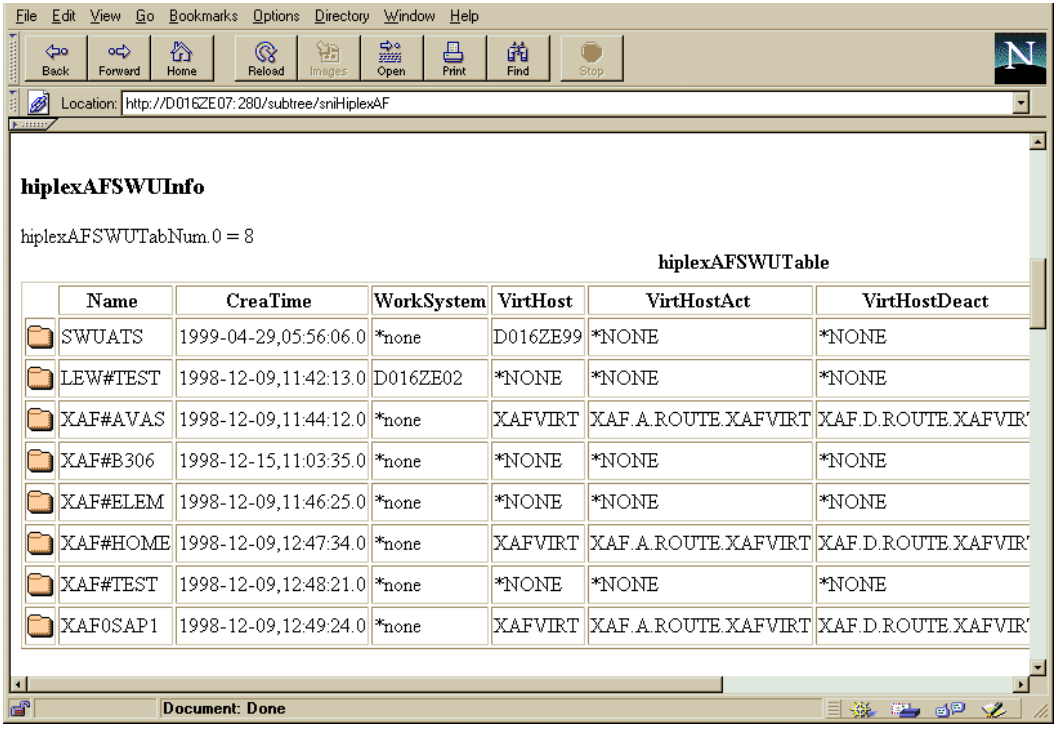


Figure 22: HIPLEX-AF: Information about the switching units

Table of host-specific parameters for the switching units

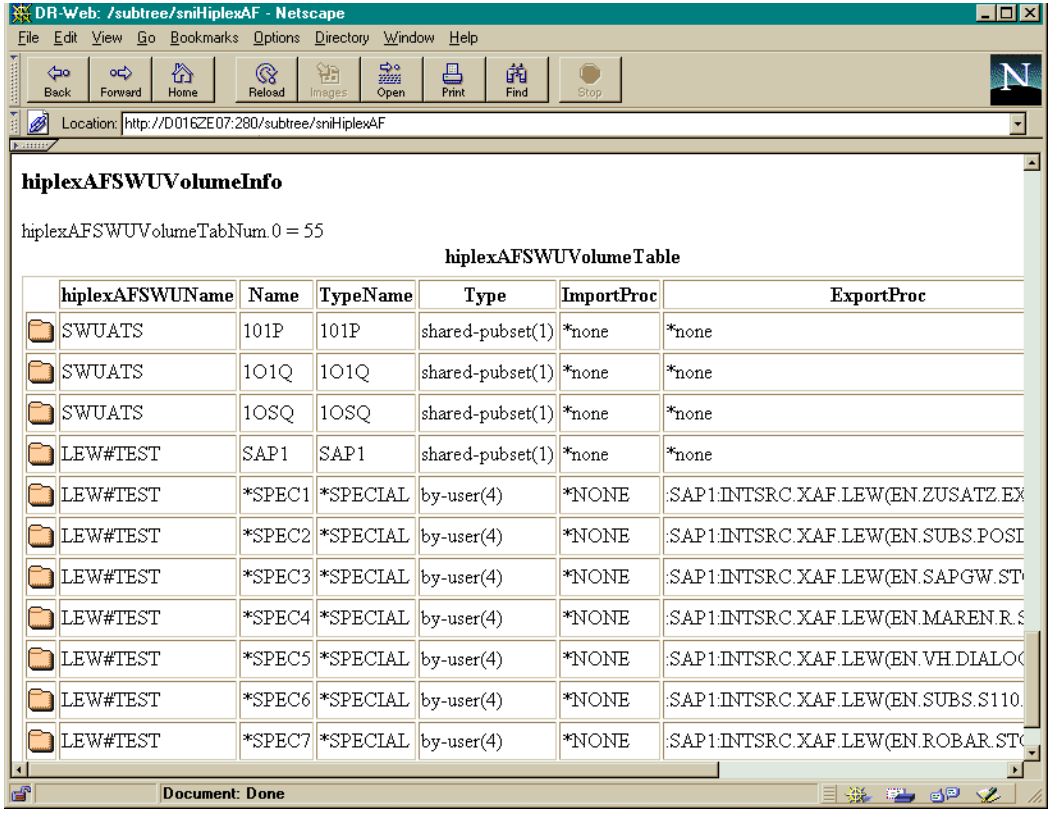
Object name	Access	Meaning
hiplexAFSWUHostParamEventId	read-only	Last action of the switching unit: – no-action – pass-over – take-over – terminate – undefined
hiplexAFSWUHost ParamStat Ind	read-write	Status of the switching unit: – work – standby – crashed – terminated – undefined
hiplexAFSWUHostParamPriority	read-write	Priority of the host on automatic switchover
hiplexAFSWUHostParamOperatorRole	read-write	Value specified for the “Operator-Role” parameter when the START-SWITCH-UNIT command is issued

Table of front-end processors

Object name	Access	Meaning
hiplexAFSWUHostFEPTabNum	read-only	Number of FEP entries
Table:		
hiplexAFSWUHostFEPIndex	read-only	Index of the front-end processor
hiplexAFSWUHostFEPName	read-write	BCAM name of the front-end processor
hiplexAFSWUHostFEPPortnumber	read-write	Number of ports used for the system

Data volume table

Object name	Access	Meaning
hiplexAFSWUVolumeTabNum	read-only	Number of data volume
Table:		
hiplexAFSWUVolumeName	read-only	Name of the data media
hiplexAFSWUVolumeType Name	read-only	Name of the data media type
hiplexAFSWUVolumeType	read-only	Data media type: – shared-pubset – pubset – private-disk – by-user – undefined
hiplexAFSWUVolumeImportProc	read-only	Name of the procedure for importing the data volume
hiplexAFSWUVolumeExportProc	read-only	Name of the procedure for exporting the data volume



hiplexAFSWUName	Name	TypeName	Type	ImportProc	ExportProc
SWUATS	101P	101P	shared-pubset(1)	*none	*none
SWUATS	1O1Q	1O1Q	shared-pubset(1)	*none	*none
SWUATS	1OSQ	1OSQ	shared-pubset(1)	*none	*none
LEW#TEST	SAP1	SAP1	shared-pubset(1)	*none	*none
LEW#TEST	*SPEC1	*SPECIAL	by-user(4)	*NONE	:SAP1:INTSRC.XAF.LEW(EN.ZUSATZ.EX
LEW#TEST	*SPEC2	*SPECIAL	by-user(4)	*NONE	:SAP1:INTSRC.XAF.LEW(EN.SUBS.POSI
LEW#TEST	*SPEC3	*SPECIAL	by-user(4)	*NONE	:SAP1:INTSRC.XAF.LEW(EN.SAPGW.ST
LEW#TEST	*SPEC4	*SPECIAL	by-user(4)	*NONE	:SAP1:INTSRC.XAF.LEW(EN.MAREN.R.S
LEW#TEST	*SPEC5	*SPECIAL	by-user(4)	*NONE	:SAP1:INTSRC.XAF.LEW(EN.VH.DIALO
LEW#TEST	*SPEC6	*SPECIAL	by-user(4)	*NONE	:SAP1:INTSRC.XAF.LEW(EN.SUBS.S110.
LEW#TEST	*SPEC7	*SPECIAL	by-user(4)	*NONE	:SAP1:INTSRC.XAF.LEW(EN.ROBAR.ST

Figure 23: HIPLEX-AF: Display of the data volume table

Table of applications

Object name	Access	Meaning
hiplexAFSWUApplicationTabNum	read-only	Number of applications
Table:		
hiplexAFSWUApplicationMonJVName	read-only	Name of the job variables for monitoring the application
hiplexAFSWUApplicationType	read-only	Type of application: – job – utm – bcam – by-user – undefined
hiplexAFSWUApplicationStartProc	read-only	Procedure for starting the application
hiplexAFSWUApplicationStopProc	read-only	Procedure for terminating the application

Table of trap filters

Object name	Access	Meaning
hiplexAFTrapFilterHost1Name	read-only	BCAM name of the system on which the HIPLEX-AF subagent is running
hiplexAFTrapFilterHost2Name	read-only	BCAM name of a monitored system in the HIPLEX-AF cluster
hiplexAFTrapSendInd	read-write	Send confirmation: – yes: Events on the monitored system are notified via a trap. – no : Events on the monitored system are not notified via traps. – undefined

Traps

Object name	Access	Meaning
hiplexAFStart	301	HIPLEX-AF main procedure started.
hiplexAFSWUStart	302	Procedure for switching unit started.
hiplexAFStop	303	Termination of HIPLEX-AF initiated.
hiplexAFCrash	304	System terminated abnormally.
hiplexAFSWUAppStop	305	Applications of a switching unit are stopped.
hiplexAFSWUAppStart	306	Applications of a switching unit are started.
hiplexAFSWUStop	307	Switching unit procedure has been stopped.

6.4 SNMP management for Host Resources

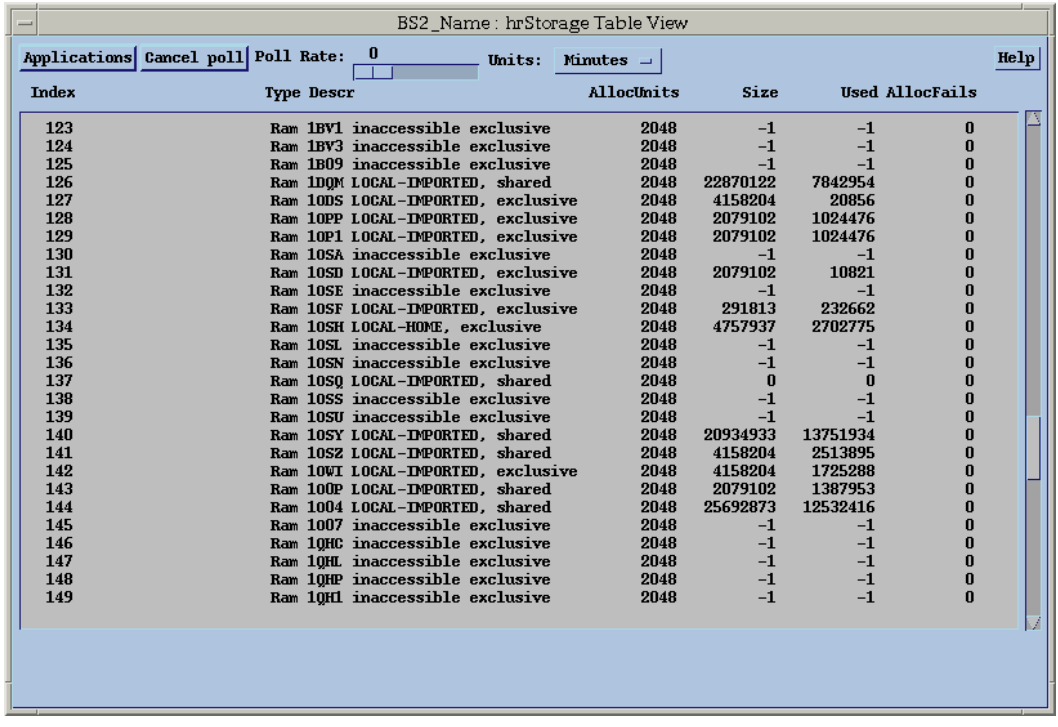
The Host Resources subagent supplies information about the system, devices, and file system, as well as about the installed software according to Standard RFC 1514**.**

Host resources system group

Object name	Access	Meaning
hrSystemUptime	read-only	The amount of time since this host was last initialized.
hrSystemDate	read-write	The host's notion of the local date and time of day.
hrSystemInitialLoadDevice	read-write	Display = -1
hrSystemInitialLoad Parameters	read-write	Display = '-'
hrSystemNumUsers	read-only	The number of user tasks for which this host is storing state information. A session is a collection of processes requiring a single act of user authentication and possibly subject to collective job control.
hrSystemProcesses	read-only	The number of task contexts currently loaded or running on this system.
hrSystemMaxProcesses	read-only	The maximum number of process contexts this system can support. If there is no fixed maximum, the value should be zero. On systems that have a fixed maximum, this object can help diagnose failures that occur when this maximum is reached. In BS2000/OSD currents 4096

Host Resources storage group

Object name	Access	Meaning
hrMemorySize	read-only	The amount of physical main memory contained by the host.
hrStorageIndex	read-only	A unique value for each logical storage area contained by the host. INTEGER (1..2147483647)
hrStorageType	read-only	The type of storage represented by this entry.
hrStorageDescr	read-only	A description of the type and instance of the storage described by this entry.
hrStorageAllocationUnits	read-only	The size, in bytes, of the data objects allocated from this pool. If this entry is monitoring sectors, blocks, buffers, or packets, for example, this number will commonly be greater than one. Otherwise this number will typically be one. For pubsets this value is 2048
hrStorageSize	read-write	The size of the storage represented by this entry, in units of *hrStorageAllocationUnits*. INTEGER (0..2147483647)
hrStorageUsed	read-only	The amount of the storage represented by this entry that is allocated, in units of *hrStorageAllocationUnits*. INTEGER (0..2147483647)
hrStorageAllocationFailures	read-only	Display = ‘0’

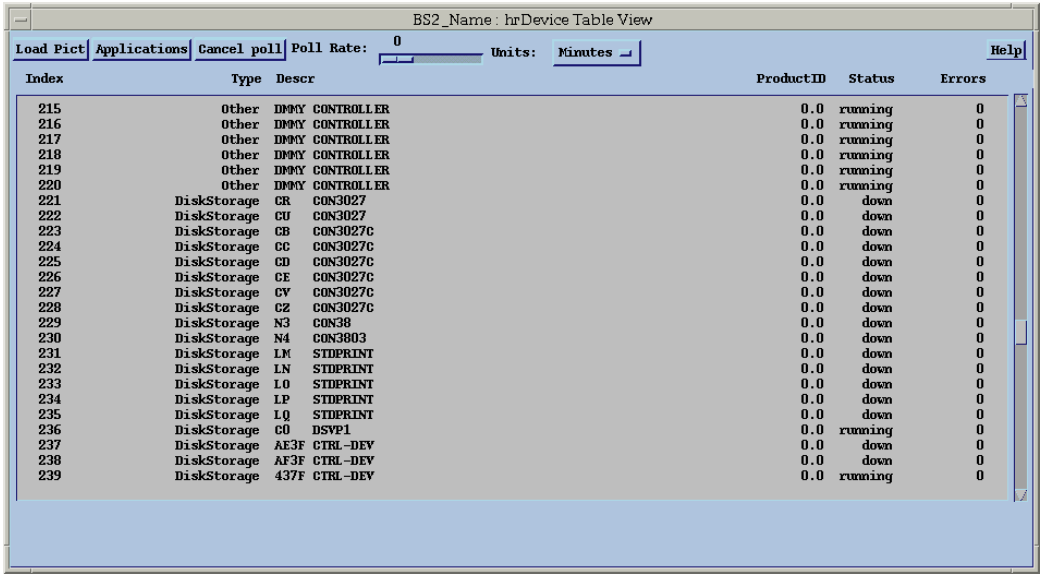


Index	Type Descr	AllocUnits	Size	Used	AllocFails
123	Ram 1BV1 inaccessible exclusive	2048	-1	-1	0
124	Ram 1BV3 inaccessible exclusive	2048	-1	-1	0
125	Ram 1BO9 inaccessible exclusive	2048	-1	-1	0
126	Ram 1DQM LOCAL-IMPORTED, shared	2048	22870122	7842954	0
127	Ram 1ODS LOCAL-IMPORTED, exclusive	2048	4158204	20856	0
128	Ram 1OPP LOCAL-IMPORTED, exclusive	2048	2079102	1024476	0
129	Ram 1OP1 LOCAL-IMPORTED, exclusive	2048	2079102	1024476	0
130	Ram 1OSA inaccessible exclusive	2048	-1	-1	0
131	Ram 1OSD LOCAL-IMPORTED, exclusive	2048	2079102	10821	0
132	Ram 1OSE inaccessible exclusive	2048	-1	-1	0
133	Ram 1OSF LOCAL-IMPORTED, exclusive	2048	291813	232662	0
134	Ram 1OSH LOCAL-HOME, exclusive	2048	4757937	2702775	0
135	Ram 1OSL inaccessible exclusive	2048	-1	-1	0
136	Ram 1OSN inaccessible exclusive	2048	-1	-1	0
137	Ram 1OSQ LOCAL-IMPORTED, shared	2048	0	0	0
138	Ram 1OSS inaccessible exclusive	2048	-1	-1	0
139	Ram 1OSU inaccessible exclusive	2048	-1	-1	0
140	Ram 1OSY LOCAL-IMPORTED, shared	2048	20934933	13751934	0
141	Ram 1OS2 LOCAL-IMPORTED, shared	2048	4158204	2513895	0
142	Ram 1OWI LOCAL-IMPORTED, exclusive	2048	4158204	1725288	0
143	Ram 1OOP LOCAL-IMPORTED, shared	2048	2079102	1387953	0
144	Ram 1OO4 LOCAL-IMPORTED, shared	2048	25692873	12532416	0
145	Ram 1OO7 inaccessible exclusive	2048	-1	-1	0
146	Ram 1QHG inaccessible exclusive	2048	-1	-1	0
147	Ram 1QHL inaccessible exclusive	2048	-1	-1	0
148	Ram 1QHP inaccessible exclusive	2048	-1	-1	0
149	Ram 1QH1 inaccessible exclusive	2048	-1	-1	0

Figure 24: Host resources subagent: host resources storage group

Host resources device group

Object name	Access	Meaning
hrDeviceIndex	read-only	A unique value for each device contained by the host. The value for each device must remain constant at least from one re-initialization of the agent to the next re-initialization. INTEGER (1..2147483647)
hrDeviceType	read-only	An indication of the type of device. If this value is hrDeviceProcessor {*hrDeviceTypes 3* } then an entry exists in the *hrProcessorTable* which corresponds to this device.
hrDeviceDescr	read-only	A textual description of this device, including the device's manufacturer and revision, and optionally, its serial number.
hrDeviceID	read-only	The product ID for this device.
hrDeviceStatus	read-only	The current operational state of the device described by this row of the table. A value unknown (1) indicates that the current state of the device is unknown. running (2) indicates that the device is active (attached) and that no unusual error conditions are known. The warning (3) state (detached-pending) The down (5) state (detached) is used only when the agent has been informed that the device is not available for any use.
hrDeviceErrors	read-only	The number of errors detected on this device.
hrProcessorFrwID	read-only	The product ID of the firmware associated with the processor.
hrProcessorLoad	read-only	Display = '0'



Index	Type	Descr	ProductID	Status	Errors
215	Other	DMMY CONTROLLER	0.0	running	0
216	Other	DMMY CONTROLLER	0.0	running	0
217	Other	DMMY CONTROLLER	0.0	running	0
218	Other	DMMY CONTROLLER	0.0	running	0
219	Other	DMMY CONTROLLER	0.0	running	0
220	Other	DMMY CONTROLLER	0.0	running	0
221	DiskStorage	CR CON3027	0.0	down	0
222	DiskStorage	CU CON3027	0.0	down	0
223	DiskStorage	CB CON3027C	0.0	down	0
224	DiskStorage	CC CON3027C	0.0	down	0
225	DiskStorage	CD CON3027C	0.0	down	0
226	DiskStorage	CE CON3027C	0.0	down	0
227	DiskStorage	CV CON3027C	0.0	down	0
228	DiskStorage	CZ CON3027C	0.0	down	0
229	DiskStorage	N3 CON38	0.0	down	0
230	DiskStorage	N4 CON3803	0.0	down	0
231	DiskStorage	LM STDPRINT	0.0	down	0
232	DiskStorage	LN STDPRINT	0.0	down	0
233	DiskStorage	LO STDPRINT	0.0	down	0
234	DiskStorage	LP STDPRINT	0.0	down	0
235	DiskStorage	LQ STDPRINT	0.0	down	0
236	DiskStorage	C0 DSVP1	0.0	running	0
237	DiskStorage	AE3F CTRL-DEV	0.0	down	0
238	DiskStorage	AF3F CTRL-DEV	0.0	down	0
239	DiskStorage	437F CTRL-DEV	0.0	running	0

Figure 25: Host resources subagent: host resources device group

Host Resources partition table

Object name	Access	Meaning
hrPartitionIndex	read-only	A unique value for each partition on this long-term storage device.The value for each long-term storage device must remain constant at least from one re-initialization of the agent to the next re-initialization. INTEGER (1..2147483647)
hrPartitionLabel	read-only	A textual description of this partition.
hrPartitionID	read-only	A descriptor which uniquely represents this partition to the operating system. On some systems, this might take on a binary representation.
hrPartitionSize	read-only	The size of this partition.
hrPartitionFSIndex	read-only	The index of the file system mounted on this partition. If no file system is mounted on this partition, then this value shall be zero. Note that multiple partitions may point to one file system, denoting that file system resides on those partitions. Multiple file systems may not reside on one partition. INTEGER (0..2147483647)

File system table

Object name	Access	Meaning
hrFSIndex	read-only	A unique value for each file system local to this host. The value for each file system must remain constant at least from one re-initialization of the agent to the next re-initialization. INTEGER (1..2147483647)
hrFSMountPoint	read-only	The path name of the root of this file system.
hrFSRemoteMountPoint	read-only	A description of the name and/or address of the server that this file system is mounted from. This may also include parameters such as the mount point on the remote file system. If this is not a remote file system, this string should have a length of zero.
hrFSType	read-only	The value of this object identifies the type of this file system.
hrFSAccess	read-only	An indication if this file system is logically configured by the operating system to be readable and writable or only readable. This does not represent any local access-control policy, except one that is applied to the file system as a whole. readWrite (1), readOnly (2)
hrFSBootable	read-only	A flag indicating whether this file system is bootable.
hrFSStorageIndex	read-only	The index of the *hrStorageEntry* that represents information about this file system. If there is no such information available, then this value shall be zero. The relevant storage entry will be useful in tracking the percent usage of this file system and diagnosing errors that may occur when it runs out of space. INTEGER (0..2147483647)
hrFSLastFullBackupDate	read-write	The last date at which this complete file system was copied to another storage device for backup. This information is useful for ensuring that backups are being performed regularly. If this information is not known, then this variable shall have the value corresponding to January 1, year 0000, 00:00:00.0, which is encoded as (hex) '00 00 01 01 00 00 00 00'.

Object name	Access	Meaning
hrFSLastPartialBackupDate	read-write	The last date at which a portion of this file system was copied to another storage device for backup. This information is useful for ensuring that backups are being performed regularly. If this information is not known, then this variable shall have the value corresponding to January 1, year 0000, 00:00:00.0, which is encoded as (hex) '00 00 01 01 00 00 00 00'.

Host Resources installed software table

Object name	Access	Meaning
hrSWInstalledLastChange	read-only	The value of sysUpTime when an entry in the hrSWInstalledTable was last added, renamed, or deleted. Because this table is likely to contain many entries, polling of this object allows a management station to determine when re-downloading of the table might be useful. The value is always the sysUpTime at the time the HR agent is initialized.
hrSWInstalledLastUpdateTime	read-only	The value of sysUpTime when the hrSWInstalledTable was last completely updated. Because caching of this data will be a popular implementation strategy, retrieval of this object allows a management station to obtain a guarantee that no data in this table is older than the indicated time. The value is always the sysUpTime at the time the HR agent is initialized
Table		
hrSWInstalledIndex	read-only	A unique value for each piece of software installed on the host. This value shall be in the range from 1 to the number of pieces of software installed on the host. INTEGER (1..2147483647)
hrSWInstalledName	read-only	A textual description of this installed piece of software, including the manufacturer, revision, the name by which it is commonly known, and optionally, its serial number.
hrSWInstalledID	read-only	The product ID of this installed piece of software.
hrSWInstalledType	read-only	The type of this software. unknown (1), operatingSystem (2), deviceDriver (3), application (4)
hrSWInstalledDate	read-only	The last-modification date of this application as it would appear in a directory listing. Always takes the value for unknown: 0000, 00:00:00.0, which stands for January 1 Year: 0000, time: 00:00:00.0

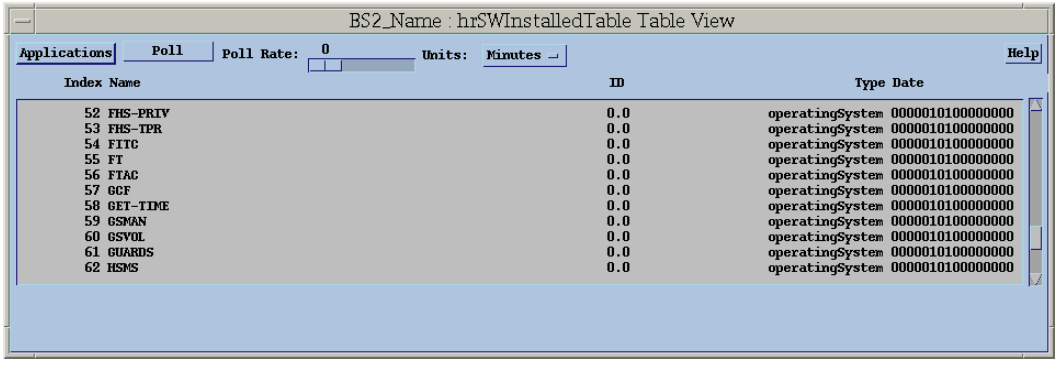


Figure 26: Host resources subagent: table of host resources installed software

6.5 SNMP management for HSMS

The HSMS subagent allows you to read and change global HSMS data. It also supplies detailed information on HSMS requests and their states. You can restrict the scope of the information displayed using the selection criteria “state” and “origin”. The HSMS subagent itself does not send any traps.

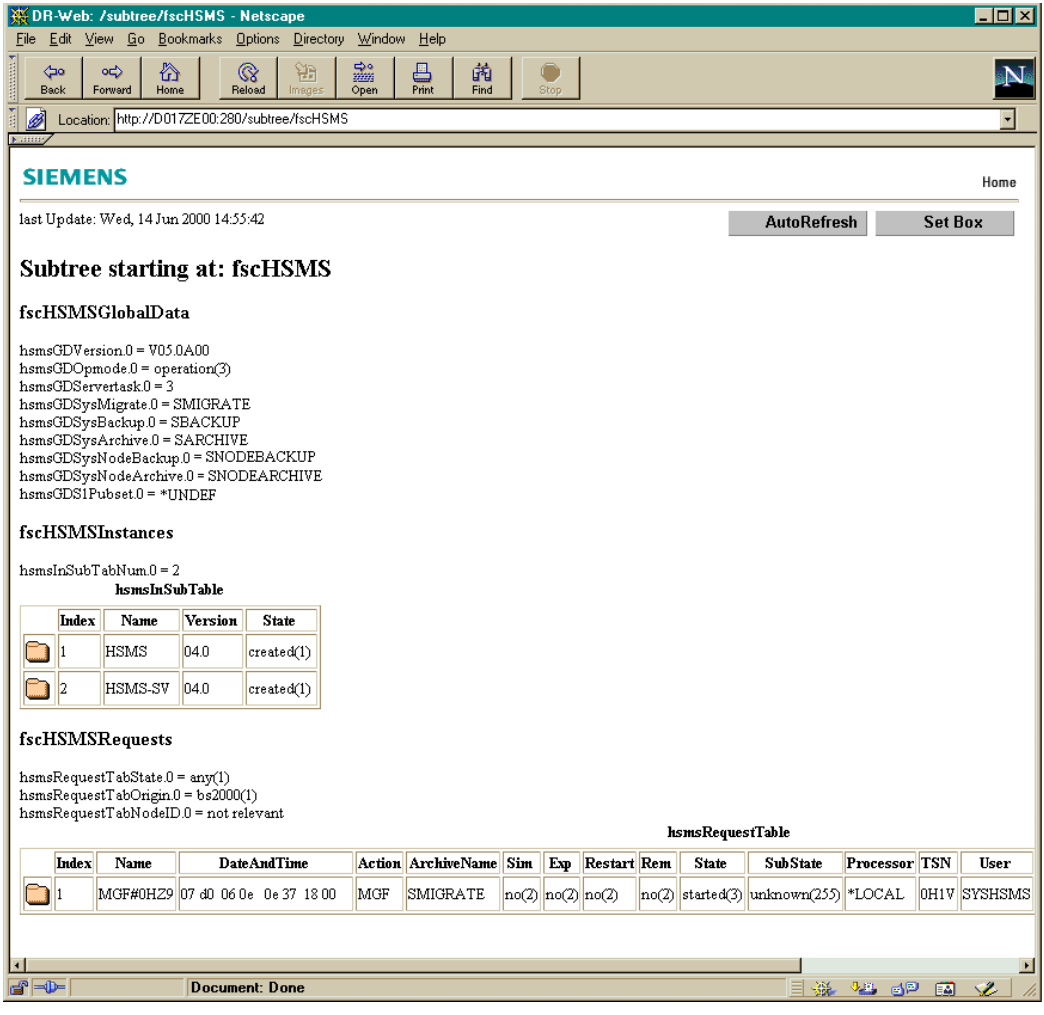


Figure 27: HSMS overview

Global data

Object name	Access	Meaning
hsmsGDVersion	read-only	Version of the subagent
hsmsGDOpmode	read-write	Operating mode of HSMS
hsmsGDServertask	read-write	Number of server tasks running
hsmsGDSysMigrate	read-only	Name of standard HSMS archive for migration
hsmsGDSysBackup	read-only	Name of standard HSMS archive for data backups
hsmsGDSysArchive	read-only	Name of standard HSMS archive for long-term archiving
hsmsGDSysNodeBackup	read-only	Name of standard HSMS archive for node backups
hsmsGDSysNodeArchive	read-only	Name of standard HSMS archive for node archiving
hsmsGDS1Pubset	read-only	ID of the pubset used as the default S1 pubset

Instances

The following values are supplied for the two subsystems HSMS and HSMS-SV:

Object name	Access	Meaning
hsmsInSubIndex	read-only	Index
hsmsInSubName	read-only	Name
hsmsInSubVersion	read-only	Version
hsmsInSubState	read-only	Status

HSMS requests

All HSMS requests that are processed by the respective BS2000/OSD host are displayed in a table. The HSMS subagent determines this information by evaluating an OPS variable.

The following information is supplied for each request:

Object name	Access	Meaning
hsmsRequestIndex	read-only	Index
hsmsRequestName	read-only	Name of request
hsmsRequestDateAndTime	read-only	Date and time the request was created
hsmsRequestAction	read-only	Action instruction*
hsmsRequestArchiveName	read-only	Archive name*
hsmsRequestSim	read-only	Simulated request (yes/no)
hsmsRequestExp	read-only	Express request (yes/no)
hsmsRequestRestart	read-only	Request following restart (yes/no)
hsmsRequestRem	read-only	Remote request (master processing for shared pubset)
hsmsRequestState	read-only	Status
hsmsRequestSubstate	read-only	Substatus
hsmsRequestProcessor	read-only	BCAM name of the host that processes the request (*LOCAL, if local).
hsmsRequestTSN	read-only	TSN of executing server task

*) as of HSMS V4.0 only

In addition, the following information is supplied depending on the particular host in question:

For BS2000/OSD only:

hsmsRequestUser	read-only	User ID under which the request was created

For workstations only:

hsmsRequestUserNo	read-only	UNIX user number
hsmsRequestNodeId	read-only	ID of the node
hsmsRequestIPAddr	read-only	IP address of the node
hsmsRequestIPPort	read-only	IP port number of the node
hsmsRequestBspId	read-only	Order ID on the client

To ensure that the subagent can continue displaying the requests even after they have terminated, the requests must not be deleted by command. Requests with the status *COMPLETED* are, however, deleted automatically by an implicit recovery at the start of each HSMS session.

The number of requests displayed can be restricted depending on the

– processing state of the requests
– host from which the request originates

Object name	**Access**	**Meaning**
hsmsRequestTabState	read-write	Restriction of the HSMS requests displayed, depending on the processing state of the individual requests. The following selection criteria are available: – ANY – COMPLETED – ACCEPTED – STARTED – INTERRUPTED
hsmsRequestTabOrigin	read-write	Restriction of the HSMS requests displayed, depending on the host from which the request originates. The host can be specified as follows: – *BS2000 (central BS2000/OSD host) – *NODE-CL (a client on which HSMS-CL is active)
hsmsRequestTabNodeID	read-write	Name of client. This object setting is rejected if *hsmsRequestTabOrigin* has the value *BS2000.

6.6 SNMP Management for OMNIS

The subagent for OMNIS enables the administration of OMNIS via SNMP. The OMNIS subagent allow the monitoring of data terminals, partners and applications. It is also possible to issue OMNIS commands. When critical events occur, the OMNIS subagent sends traps.

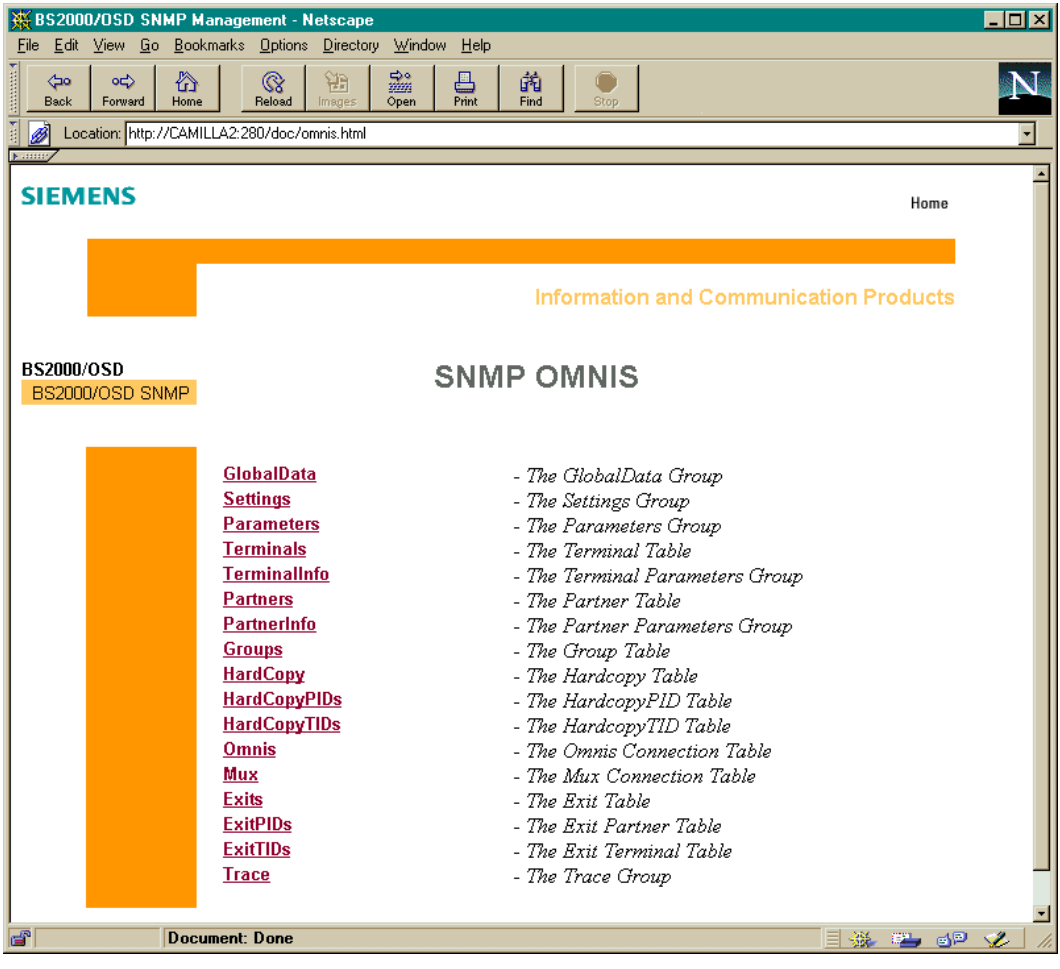


Figure 28: OMNIS overview

Global OMNIS information

Object name	Access	Meaning
omnisGlobalDataSubagent Version	read-only	Version of OMNIS Subagent
omnisGlobalDataTabNum	read-only	The total number of OMNISes
omnisGlobalDataActID	read-write	The ID of the current OMNIS. All other data refer to this OMNIS. Setting this ID results in the display of OMNIS with the specified ID.
omnisGlobalDataActName	read-write	The name of the current OMNIS. All other data refer to this OMNIS. Setting this name results in the display of OMNIS with the specified name.
Table of configured OMNIS:		
omnisGlobalDataOmnID	read-only	OMNIS index
omnisGlobalDataVersion	read-only	OMNIS version
omnisGlobalDataOmnName	read-only	OMNIS name
omnisGlobalDataState	read-write	OMNIS state Connection to OMNIS is set up / cleared down. open (1), close (2)

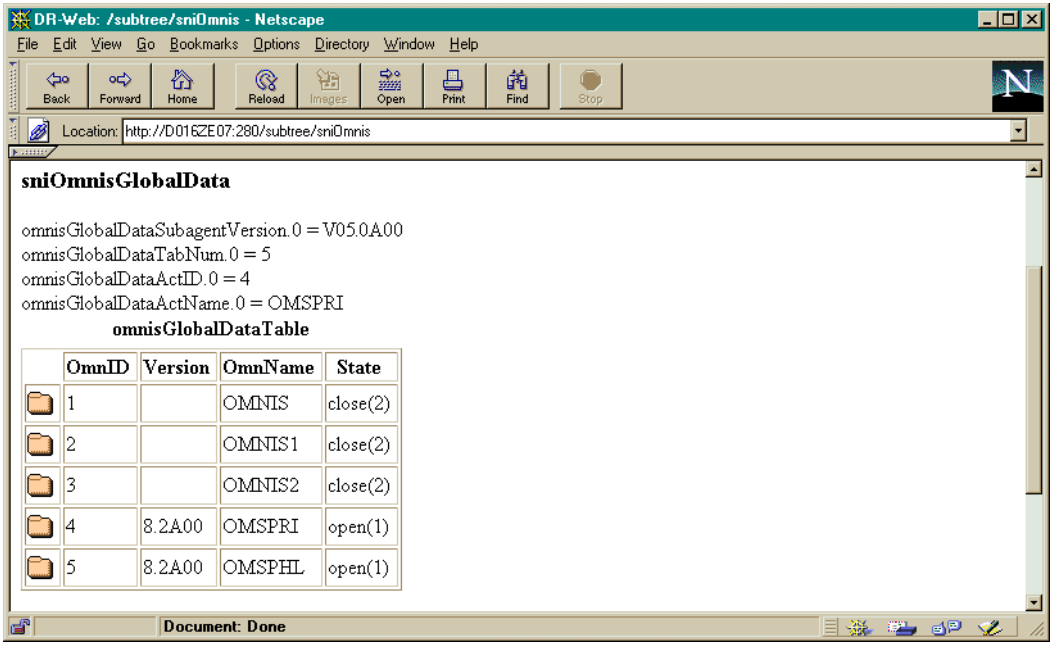


Figure 29: OMNIS subagent: display of global OMNIS information

OMNIS settings

Object name	Access	Meaning
omnisSettingsAppName	read-only	The NEA name of the OMNIS application
omnisSettingsAppNameISO	read-only	The ISO name of the OMNIS application
omnisSettingsNumPartners	read-only	The actual number of partners
omnisSettingsNumTerminals	read-only	The actual number of terminals
omnisSettingsDSTMax	read-write	The maximum number of DSTs
omnisSettingsPTNMax	read-write	The maximum number of partners
omnisSettingsPACMax	read-write	The maximum number of partners for one DST
omnisSettingsState	read-only	Subagent connected open (1), close (2), end (3), unknown (99)
omnisSettingsAPASS	read-write	Administrator password
omnisSettingsHOLD	read-write	Hold the connection yes (1), no (2), std (3), auto (4), unknown (99)
omnisSettingsHcyForm	read-write	Printer form
omnisSettingsHCopy	read-write	Route the output to a printer
omnisSettingsLogging	read-write	Protocolling in logging file yes (1), no (2), unknown (99)
omnisSettingsChangeLogging	read-write	Switch to a new logging file change (1), unknown (99)
omnisSettingsACK	read-write	Acknowledgment required yes (1), no (2), std (3), unknown (99)
omnisSettingsMTAB	read-write	The report table
omnisSettingsEXIT	read-write	The address code of EXIT

Object name	Access	Meaning
omnisSettingsOpncon	read-write	The connect permission std (1), free (2), dcl (3), unknown (99)
omnisSettingsBreakKey	read-write	The K-key to produce a break on a $DIALOG-Partner: k1 (1) ... k14 (14), no (98), unknown (99)
omnisSettingsCallKey	read-write	The K-key for calling k1 (1) ... k14 (14), no (98), unknown (99)
omnisSettingsCallInf	read-write	Notify the call-state-information yes (1), no (2), std (3), unknown (99)
omnisSettingsPac	read-write	The address code of the partner line (1), prefix (2), no (3), std (4), unknown (99)
omnisSettingsInputLog	read-write	The protocol for the terminal input rec (1), send (2), both (3), std (4), unknown (99)
omnisSettingsOutputLog	read-write	The protocol for the terminal output rec (1), send (2), both (3), std (4), unknown (99)
omnisSettingsLine25	read-write	Use the 25. display-line yes (1), no (2), std (3), unknown (99)

Object name	Access	Meaning
omnisSettingsDisMod	read-write	The display mode system (1), omnis (2), mixed (3), unknown (99)
omnisSettingsKPAC	read-write	K-key to switch in command mode/to change the partner k1 (1) ... k14 (14), no (98), unknown (99)
omnisSettingsExitPri	read-write	The priority for SET, PARAMETER and OPTION by EXIT set-opt (1), opt-set (2), std (3), unknown (99)
omnisSettingsReply	read-write	Route Data to UCON application restricted (1), all (2), unknown (99)
omnisSettingsExitAuth	read-write	The authorization for Exit command all (1), adm (2), unknown (99)
omnisSettingsLoggPri	read-write	The priority for SET, PARAMETER and OPTION by Logging set-opt (1), opt-set (2), unknown (99)
omnisSettingsAudit	read-write	OMNIS-Audit on (1), off (2), unknown (99)
omnisSettingsMDefAuth	read-write	The authorization for MDEF command all (1), adm (2), unknown (99)
omnisSettingsHoldPri	read-write	The priority for SET and OPTION by HOLD set-opt (1), opt-set (2), unknown (99)

Object name	Access	Meaning
omnisSettingsInsave	read-write	The saved key k1 (1) ... k14 (14), f1 (21) ... f24 (44), no (97), std (98), unknown (99)
omnisSettingsOpnStart	read-write	Start-sequence permitted yes (1), no (2), unknown (99)
omnisSettingsExclPartner	read-write	The exclusive partner yes (1), no (2), std (3), unknown (99)
omnisSettingsSave	read-write	Saved after disconnect pkey (1), screen (2), all (3), no (4), std (5), unknown (99)
omnisSettingsMessageALL	read-write	Message to all OMNIS users
omnisSettingsMessageADM	read-write	Message to OMNIS administrator

OMNIS parameters

Object name	Access	Meaning
omnisParametersAppName	read-only	The name of the OMNIS application
omnisParametersAppNameISO	read-only	The ISO name of the OMNIS application
omnisParametersPrefix	read-only	Prefix of standby application
omnisParametersProName	read-only	Processor name
omnisParametersVirtProName	read-only	The virtual processor name
omnisParametersLoggingFile	read-only	The name of logging file
omnisParametersStartupFile	read-only	The name of startup file
omnisParametersConfigFile	read-only	The name of configuration file
omnisParametersConfUpdate	write-only	The object for CONFUPDATE command (write only), start (1)
omnisParametersModulFile	read-only	The name of module file
omnisParametersBulletinFile	read-only	The name of bulletin file
omnisParametersTextFile	read-only	The name of text file
omnisParametersPagePool	read-only	The name of PagePool File
omnisParametersIOAreaLength	read-only	Length of IO Area
omnisParametersTWorkLength	read-only	Length of Terminal Work Area
omnisParametersPWorkLength	read-only	Length of Partner Work Area
omnisParametersTextKeLength	read-only	Length of text key
omnisParametersSecurity Level	read-only	The security level high (1), medium (2), low (3), unknown (99)
omnisParametersDCAMInt Vers	read-only	The version of DCAM Interface
omnisParametersVTSUBVers	read-only	The version of local VTSU-B
omnisParametersVTSUCB Vers	read-only	The version of local VTSUBC
omnisParametersCMD	read-write	BS2000 command for OMNIS task
omnisParametersDump	read-write	/DUMP command start (1), unknown (99)
omnisParametersDumpMsgNr	read-write	OMNIS message number
omnisParametersDumpInsert	read-write	/DUMP command
omnisParametersDumpInsertNr	read-write	Number of Insert

OMNIS terminal table

Object name	Access	Meaning
omnisTerminalsStatus	read-write	State-Flag for restriction in Show Table all (1), activ (2), -- default hold (3), inactive (4)
omnisTerminalsTabNum	read-only	The number of entries in the table *omnisTerminalsTable*
omnisTerminalsTID	read-write	Terminal ID
omnisTerminalsPtnName	read-only	Partner name
omnisTerminalsProName	read-only	Processor name
omnisTerminalsTyp	read-only	Terminal type term (1), appl (2), skp (3), cons (4), unknown (99)
omnisTerminalsState	read-write	Terminal state / value *cancel* only for *write* decl (1), opn (2), act (3), los (4), cls (5), hold (6), inact (7), cancel (8), unknown (99)
omnisTerminalsRoute	read-only	Route ind (1), dir (2), unknown (99)
omnisTerminalsKPAC	read-only	K-key to switch in command mode/to change the partner k1 (1) ... k14 (14), no (97), std (98), unknown (99)
omnisTerminalsUser	read-only	User
omnisTerminalsMessage	write-only	Message to one TID (only for write)

OMNIS terminal information

Object name	Access	Meaning
omnisTerminalInfoTID	read-write	Terminal ID
omnisTerminalInfoPtnName	read-only	Partner name
omnisTerminalInfoProName	read-only	Processor name
omnisTerminalInfoTyp	read-only	Terminal type term (1), appl (2), skp (3), cons (4), unknown (99)
omnisTerminalInfoState	read-only	Terminal state decl (1), opn (2), act (3), los (4), cls (5), hold (6), inact (7), unknown (99)
omnisTerminalInfoRoute	read-only	Route ind (1), dir (2), unknown (99)
omnisTerminalInfoKPAC	read-only	K-key to switch in command mode/to change the partner k1 (1) ... k14 (14), no (97), std (98), unknown (99)
omnisTerminalInfoUser	read-only	User
omnisTerminalInfoPAC	read-only	The type of display output for PAC no (1), line (2), prefix (3), std (4), unknown (99)
omnisTerminalInfoADM	read-only	Administration permitted no (1), yes (2), unknown (99)

Object name	Access	Meaning
omnisTerminalInfoOPass	read-only	The password for OCCUPY command no (1), yes (2), unknown (99)
omnisTerminalInfoMTAB	read-only	The report table
omnisTerminalInfoExit	read-only	The address code of EXIT
omnisTerminalInfoHold	read-only	Hold the connection yes (1), no (2), std (3), auto (4), unknown (99)
omnisTerminalInfoChange	read-only	CHANGELOG permitted yes (1), no (2), unknown (99)
omnisTerminalInfoHcopy	read-only	The hardcopy printer
omnisTerminalInfoAck	read-only	Acknowledgment required yes (1), no (2), std (3), unknown (99)
omnisTerminalInfoListening	read-only	Another listening terminal
omnisTerminalInfoColour	read-only	The display color blue (1), cyan (2), green (3), yellow (4), magenta (5), red (6), white (7), unknown (99)
omnisTerminalInfoLogging	read-only	Protocolling in logging file yes (1), no (2), std (3), unknown (99)
omnisTerminalInfoBerID	read-only	Password/permission in SKP generated yes (1), no (2), unknown (99)

Object name	Access	Meaning
omnisTerminalInfoDeclared	read-only	Declared yes (1), no (2), unknown (99)
omnisTerminalInfoBreakKey	read-only	The K-key to produce a break on a $DIALOG-Partner: k1 (1) ... k14 (14), no (97), std (98), unknown (99)
omnisTerminalInfoCallKey	read-only	The K-key for calling k1 (1) ... k14 (14), no (97), std (98), unknown (99)
omnisTerminalInfoCallInf	read-only	Notify the call-state-information yes (1), no (2), std (3), unknown (99)
omnisTerminalInfoDisMod	read-only	Display mode std (1), system (2), omnis (3), mixed (4), unknown (99)
omnisTerminalInfoConnect	read-only	The state of connection logon (1), start (2), unknown (99)
omnisTerminalInfoOpncon	read-only	The connect permission std (1), free (2), dcl (3), unknown (99)
omnisTerminalInfoPacAnz	read-only	Number of active partners

Object name	Access	Meaning
omnisTerminalInfoInput Logging	read-only	The protocol for the terminal input std (1), both (2), rec (3), send (4), unknown (99)
omnisTerminalInfoOutput Logging	read-only	The protocol for the terminal output std (1), both (2), rec (3), send (4), unknown (99)
omnisTerminalInfoAutoLogoff	read-only	Automatic logoff std (1), yes (2), no (3), unknown (99)
omnisTerminalInfoLine25	read-only	Use the 25. display-line yes (1), no (2), std (3), unknown (99)
omnisTerminalExclPartner	read-only	The exclusive partner yes (1), no (2), std (3), unknown (99)
omnisTerminalInfoSave	read-only	Saved after disconnect std (1), screen (2), pkey (3), all (4), no (5), unknown (99)
omnisTerminalInfoReply	read-only	Route Data to UCON-application restricted (1), all (2), std (3), unknown (99)

Object name	Access	Meaning
omnisTerminalInfoUserProt	read-only	The user protocol no (1), omnis (2), vtsucb (3), unknown (99)
omnisTerminalInfoTestmode	read-only	The test mode no (1), yes (2), unknown (99)
omnisTerminalInfoInsave	read-only	The saved key k1 (1) ... k14 (14), f1 (21) ... f24 (44), no (97), std (98), unknown (99)
omnisTerminalInfoSNMP	read-only	SNMP-terminal controlling no (1), yes (2), unknown (99)
omnisTerminalInfoTransProt	read-only	The transport protocol
omnisTerminalInfoHcyForm	read-only	Printer form

OMNIS Partner table

Object name	Access	Meaning
omnisPartnerStatus	read-write	State flag for restriction in Show Table all (1), active (2), -- default hold (3), inactive (4)
omnisPartnerTabNum	read-only	The number of entries in the table *omnisPartnerTable*
omnisPartnerSelectTID	read-write	The Terminal ID the following table is reduced to
OMNIS Partner-Table:		
omnisPartnerPID	read-write	Partner ID
omnisPartnerPAC	read-only	Partner address code
omnisPartnerPtnName	read-only	Partner name
omnisPartnerProName	read-only	Processor name
omnisPartnerTyp	read-only	Terminal type tiam (1), dcam (2), ucon (3), utm (4), svp (5), skp (6), unknown (99)
omnisPartnerState	read-write	Partner state opn (1), act (2), los (4), cls (5), hold (6), inact (7), cancel (8), unknown (99)
omnisPartnerRoute	read-only	Route ind (1), dir (2), mux (3), unknown (99)

Object name	Access	Meaning
omnisPartnerKPAC	read-only	K-key to change the partner k1 (1) ... k14 (14), no (97), std (98), unknown (99)
omnisPartnerTid	read-only	Terminal ID

OMNIS partner information

Object name	Access	Meaning
omnisPartnerInfoPID	read-write	Partner ID
omnisPartnerInfoPAC	read-only	Partner address code
omnisPartnerInfoPtnName	read-only	Partner name
omnisPartnerInfoProName	read-only	Processor name
omnisPartnerInfoTyp	read-only	Partner type tiam (1), dcam (2), ucon (3), utm (4), svp (5), skp (6), unknown (99)
omnisPartnerInfoState	read-only	Partner state opn (1), act (2), los (3), cls (4), hold (5), inact (6), unknown (99)
omnisPartnerInfoRoute	read-only	Route ind (1), dir (2), mux (3), unknown (99)
omnisPartnerInfoKPAC	read-only	K-key to change the partner k1 (1) ... k14 (14), no (97), std (98), unknown (99)
omnisPartnerInfoRepApp Name	read-only	The name of the representative application
omnisPartnerInfoOPass	read-only	The password for OCCUPY command no (1), yes (2), unknown (99)
omnisPartnerInfoMTAB	read-only	The report table
omnisPartnerInfoExit	read-only	The address code of EXIT

Object name	Access	Meaning
omnisPartnerInfoHold	read-only	Hold the connection yes (1), no (2), std (3), unknown (99)
omnisPartnerInfoChange	read-only	CHANGELOG permitted yes (1), no (2), unknown (99)
omnisPartnerInfoHcopy	read-only	The hardcopy printer
omnisPartnerInfoClass	read-only	The message class for the partner sav (1), out (2), del (3), unknown (99)
omnisPartnerInfoColour	read-only	The display color blue (1), cyan (2), green (3), yellow (4), magenta (5), red (6), white (7), unknown (99)
omnisPartnerInfoProtocol	read-only	The protocol type for a DCAM-connection dssim (1), omnis (2), unknown (99)
omnisPartnerInfoLogging	read-only	Protocolling in logging file yes (1), no (2), std (3), unknown (99)
omnisPartnerInfoLPass	read-only	Connection password required yes (1), no (2), unknown (99)
omnisPartnerInfoDeclared	read-only	Declared yes (1), no (2), unknown (99)

Object name	Access	Meaning
omnisPartnerInfoAutoLogoff	read-only	Automatic logoff std (1), yes (2), no (3), unknown (99)
omnisPartnerInfoLine25	read-only	Use the 25. display-line yes (1), no (2), std (3), unknown (99)
omnisPartnerStartSequ	read-only	Number of the start sequence
omnisPartnerInfoCMsg	read-only	Connection message yes (1), no (2), unknown (99)
omnisPartnerInfoLCase	read-only	Lower-case character to partner permitted yes (1), no (2), unknown (99)
omnisPartnerInfoSave	read-only	Saved after disconnect std (1), screen (2), pkey (3), all (4), no (5), unknown (99)
omnisPartnerInfoTid	read-only	Terminal ID
omnisPartnerInfoSNMP	read-only	SNMP terminal controlling no (1), yes (2), unknown (99)
omnisPartnerInfoPACPrefix	read-only	The type of display output for PAC no (1), std (2), line (3), prefix (4), unknown (99)
omnisPartnerInfoBerid	read-only	Password/permission in SKP generated no (1), yes (2), unknown (99)

Object name	Access	Meaning
omnisPartnerInfoConnect	read-only	The state of connection opncon (1), logon (2), start (3), unknown (99)

OMNIS group table

Object name	Access	Meaning
omnisGroupsTabNum	read-only	The number of entries in the table *omnisGroupsTable*
omnisGroupsSelectTID	read-write	The terminal ID for which the group table is to be displayed
OMNIS groups table:		
omnisGroupsGAC	read-only	The address code of the group
omnisGroupsPAC	read-only	The address code of the partner
omnisGroupsTid	read-only	The terminal ID

OMNIS hardcopy table

Object name	Access	Meaning
omnisHardCopyStatus	read-write	State flag for restriction in Show Table all (1), active (2), -- default inactive (4)
omnisHardCopyTabNum	read-only	The number of entries in the table *omnisHardCopyTable*
Hardcopy Table:		
omnisHardCopyHAC	read-write	The address code of the printer
omnisHardCopyHID	read-write	The ID of the printer
omnisHardCopyPtnName	read-write	Partner name
omnisHardCopyProName	read-write	Processor name
omnisHardCopyState	read-write	The state of the printer cls-p (1), opn (2), act (3), los (4), cls (5), inact (6), cancel (7), unknown (99)
omnisHardCopyINOP	read-only	The address code of the substituting printer
omnisHardCopyConnect	read-write	The start time for the connection to a printer s (1), u (2), unknown (99)
omnisHardCopyRestart	read-write	Restart hardcopy connection start (1), unknown (99)

OMNIS HAC/PID table

Object name	Access	Meaning
omnisHardCopyPIDsTabNum	read-only	The number of entries in *omnisHardCopyTable*
omnisHardCopyPIDsSelect Hid	read-write	Hardcopy ID for which HAC/PID is to be displayed
HAC/PID Table:		
omnisHardCopyPIDsHID	read-only	Identifier for printer (HID)
omnisHardCopyPIDsID	read-only	Identifier of partner (PID)

OMNIS HAC/TID table

Object name	Access	Meaning
omnisHardCopyTIDsTabNum	read-only	The number of entries in the *omnisHardCopyTable*
omnisHardCopyTIDsSelect Hid	read-write	Hardcopy ID for which HAC/PID is to be displayed
HAC/TID Table:		
omnisHardCopyTIDsHID	read-only	Identifier of printer (HID)
omnisHardCopyTIDsHAC	read-only	The address code of printer (HAC)
omnisHardCopyTIDsID	read-only	Identifier of terminal (TID)

OMNIS Hardcopy Create

Object name	Access	Meaning
omnisHardCopyCreateHAC	read-write	The address code for a new printer
omnisHardCopyCreateHID	read-only	The ID for a created printer
omnisHardCopyCreatePtn Name	read-write	The terminal name for a new printer
omnisHardCopyCreatePro Name	read-write	The processor name for a new printer
omnisHardCopyCreateInop	read-write	The address code of the substituting printer
omnisHardCopyCreate Connect	read-write	The start time for the connection to a new printer s (1), u (2), unknown (99)

OMNIS-OMNIS table

Object name	Access	Meaning
omnisOmnisStatus	read-write	State flag for restriction in Show Table all (1), active (2), -- default inactive (4)
omnisOmnisTabNum	read-only	The number of entries in *omnisOmnisTable*
OMNIS Table:		
omnisOmnisOAC	read-write	Omnis address code
omnisOmnisID	read-only	Identifier of OMNIS
omnisOmnisPtnName	read-only	Partner name
omnisOmnisProName	read-only	Processor name
omnisOmnisState	read-only	The OMNIS state cls-p (1), opn (2), act (3), los (4), cls (5), inact (6), cancel (7), unknown (99)
omnisOmnisConnect	read-only	The state of connection start (1), opncon (2), unknown (99)
omnisOmnisTime	read-only	The connect duration
omnisOmnisLPass	read-only	The connection password yes (1), no (2), unknown (99)
omnisOmnisOpncon	read-only	The connect permission dcl (1), free (2), unknown (99)
omnisOmnisRestart	write-only	Restart Omnis-Omnis connection start (1), unknown (99)

OMNIS Mux table

Object name	Access	Meaning
omnisMuxStatus	read-write	State flag for restriction in Show Table all (1), active (2), -- default inactive (4)
omnisMuxTabNum	read-only	The number of entries in *omnisMuxTable*
Mux table:		
omnisMuxID	read-only	Identifier of MUX
omnisMuxPtnName	read-only	Partner name
omnisMuxProName	read-only	Processor name
omnisMuxState	read-only	The state of the MUX cls-p (1), opn (2), act (3), los (4), cls (5), inact (6), cancel (7), unknown (99)
omnisMuxConnect	read-only	The state of connection start (1), opncon (2), unknown (99)
omnisMuxLPass	read-only	The connection password yes (1), no (2), unknown (99)
omnisMuxSessions	read-only	number of actual sessions
omnisMuxAvailability	read-only	The availability yes (1), no (2), unknown (99)

OMNIS EXIT table

Object name	Access	Meaning
omnisExitTabNum	read-only	The number of entries in *omnisExitTable*
EXIT table:		
omnisExitEAC	read-write	The address code of EXIT
omnisExitID	read-only	Identifier of Exit
omnisExitModul	read-write	The EXIT module

OMNIS EXIT/TID table

Object name	Access	Meaning
omnisExitTIDsTabNum	read-only	The number of entries in *omnisExitTable*
omnisExitTIDsSelectEac	read-write	The exit address code to be displayed for the EXIT/ID table
EXIT/TID Table:		
omnisExitTIDsEAC	read-only	The address code of EXIT
omnisExitTIDsID	read-only	Identifier of terminal (TID)

OMNIS EXIT/PID table

Object name	Access	Meaning
omnisExitPIDsTabNum	read-only	The number of entries in the *omnisExitTable*
omnisExitPIDsSelectEac	read-write	The terminal ID to be displayed for the EXIT/ID table
EXIT/PID Table:		
omnisExitPIDsEAC	read-only	The address code of EXIT
omnisExitPIDsID	read-only	Identifier of partner (PID)

OMNIS Exit Create

Object name	Access	Meaning
omnisExitCreateEAC	read-write	The address code for a new EXIT
omnisExitCreateModul1	read-write	The 1st module for a new EXIT
omnisExitCreateModul2	read-write	The 2nd module for a new EXIT
omnisExitCreateModul3	read-write	The 3rd module for a new EXIT
omnisExitCreateModul4	read-write	The 4th module for a new EXIT
omnisExitCreateModul5	read-write	The 5th module for a new EXIT
omnisExitCreateModul6	read-write	The 6th module for a new EXIT
omnisExitCreateModul7	read-write	The 7th module for a new EXIT
omnisExitCreateModul8	read-write	The 8th module for a new EXIT
omnisExitCreateModul8	read-write	The 9th module for a new EXIT
omnisExitCreateModul10	read-write	The 10th module for a new EXIT
omnisExitCreateModul11	read-write	The 11th module for a new EXIT
omnisExitCreateModul12	read-write	The 12th module for a new EXIT
omnisExitCreateModul13	read-write	The 13th module for a new EXIT
omnisExitCreateModul14	read-write	The 14th module for a new EXIT
omnisExitCreateOption	read-write	Defines the operation to be performed create (1), modify (2), delete (3), unknown (99)

OMNIS trace

Object name	Access	Meaning
omnisTraceConnection	read-write	DCAM connection trace yes (1), no (2), unknown (99)
omnisTraceExit	read-write	EXIT trace yes (1), no (2), unknown (99)
omnisTraceTransport	read-write	DCAM transport trace yes (1), no (2), select (3), unknown (99)
omnisTraceTransportTrm	read-write	DCAM transport trace for a selected terminal
omnisTraceTransporthcy	read-write	DCAM transport trace for a selected printer
omnisTraceTransportmux	read-write	DCAM transport trace for a selected MUX
omnisTraceTransportoms	read-write	DCAM transport trace for a selected OMNIS

Traps

Object name	Access	Meaning
Enterprise = 1.3.6.1.4.1.231.2.31.20		
omnisStopTrap	301	Sending a TRAP, if an OMNIS has been terminated
omnisStartTrap	302	Sending a TRAP, if an OMNIS has been activated
omnisConnStopTrap	303	Sending a TRAP, if a critical connection has been deactivated
omnisDstConnStopTrap	304	Sending a TRAP, if an OMNIS session has been normally terminated
omnisEventTrap	305	Sending a Trap, if a critical Omnis message arrived
omnisDstLevelTrap	306	Sending a TRAP, if DSTMAX has been reached
omnisPacLevelTrap	307	Sending a TRAP, if PACMAX has been reached
omnisPtnLevelTrap	308	Sending a TRAP, if PTNMAX has been reached
omnisMuxConnStopTrap	309	Sending a TRAP, if a critical Mux connections has been deactivated.
omnisOmnConnStopTrap	310	Sending a TRAP, if a critical OMNIS-OMNIS connections has been deactivated.
omnisHcConnStopTrap	311	DCAM- connection trace
omnisDumpWriteTrap	312	Sending a TRAP, if a critical hardcopy connections has been deactivated.
omnisDumpEndTrap	313	Sending a TRAP, if OMNIS has terminated the dump.
omnisEndTrap	314	Sending a TRAP, is OMNIS has terminated normally.

OMNIS Trap group

User-defined trap objectt

Object name	Access	Meaning
Enterprise = 1.3.6.1.4.1.231.2.31.21		
omnisTrapMsgText	read-only	OMNIS message text generates a trap.

User-defined trap

Object name	Access	Meaning
Enterprise = 1.3.6.1.4.1.231.2.31.21		
omnisGeneralTrap	320	TRAP is sent in all user-defined cases.

6.7 SNMP management for basic performance monitoring with SM2

The subagent for basic performance monitoring with SM2 supplies average values for monitoring the CPU utilization and I/O rates.

Object name	Access	Meaning
Group SM2 Params		
sm2Status	read-only	Status of the measurement subsystem SM2
sm2Interval	read-write	Online cycle of the measurement subsystem SM2 in units of a second, also called measurement interval (default value: 120)
Group SM2 Basic		
sm2BasicStatus	read-only	Status assigned to the BASIC buffer
sm2BasicTime	read-only	Time at the end of the last measurement interval DateAndTime in accordance with RFC1514
sm2BasicTimeString	read-only	Time at the end of the last measurement interval. Date and time in a directly displayable format: YYYY-MM-DD,hh:mm:ss.d[,shh:mm] Where YYYY-MM-DD is the local data in the order year-month-day, hh:mm:ss.d stands for the local time in hours minutes, seconds and tenths of seconds, shh:mm indicates the deviation from UTC in (-|+) hours:minutes
Group of I/O values		
sm2TimeIOStatus	read-only	Status associated with TIME IO buffer (data-supported, data-invalid, prg-inactive, no-subsystem, sm2-not-running, unknown)
sm2TimeIOMachTabNumber	read-only	Number of entries in following machine table
sm2TimeIOMachTabIdleTime	read-only	Time in which the logical machine was inactive (not stopped) in o/oo (share of time in parts per thousand)
sm2TimeIOMachTabTUTime	read-only	TU time of the logical machine in o/oo (share of time in parts per thousand)
sm2TimeIOMachTabTPRTime	read-only	TPR time of the logical machine in o/oo (share of time in parts per thousand)

Object name	Access	Meaning
sm2TimeIOMachTabSIHTime	read-only	SIH time of the logical machine in o/oo (share of time in parts per thousand)
sm2TimeIOMachTabStopTime	read-only	Stop time of the logical machine in o/oo (share of time in parts per thousand)
sm2TimeIOMachTabPagingIO	read-only	Rate of paging IO concerning the logical machine in (number of paging IOs per second) * 10
sm2TimeIOMachTabDiskIO	read-only	Rate of disk IO concerning the logical machine in (number of disk IOs per second) * 10
sm2TimeIOMachTabTapeIO	read-only	Rate of tape IO concerning the logical machine in (number of tape IOs per second) * 10
sm2TimeIOMachTabPrinterIO	read-only	Rate of printer IO concerning the logical machine in (number of printer IOs per second) * 10
sm2TimeIOMachTabOtherIO	read-only	Rate of other IO concerning the logical machine in (number of other IOs per second) * 10

6.8 SNMP management for SESAM databases

The subagent for managing SESAM databases supplies information about SESAM databases and SESAM DBHs, which are used to processes these databases. It supports the RDBMS-MIB according to RFC 1697.

Table of installed databases

Object name	Access	Meaning
rdbmsDbVendorName	read-only	The name of the vendor whose RDBMS manages this database, for informational purposes.
rdbmsDbName	read-only	The name of this database, in a product specific format. The product may need to qualify the name in some way to resolve conflicts if it is possible for a database name to be duplicated on a host. It might be necessary to construct a hierarchical name embedding the RDBMS instance/installation on the host, and/or the owner of the database. For example: "/test-installation/database-owner/database-name".
rdbmsDbContact	read-write	The textual identification of the contact person for this managed database, together with information on how to contact this person.

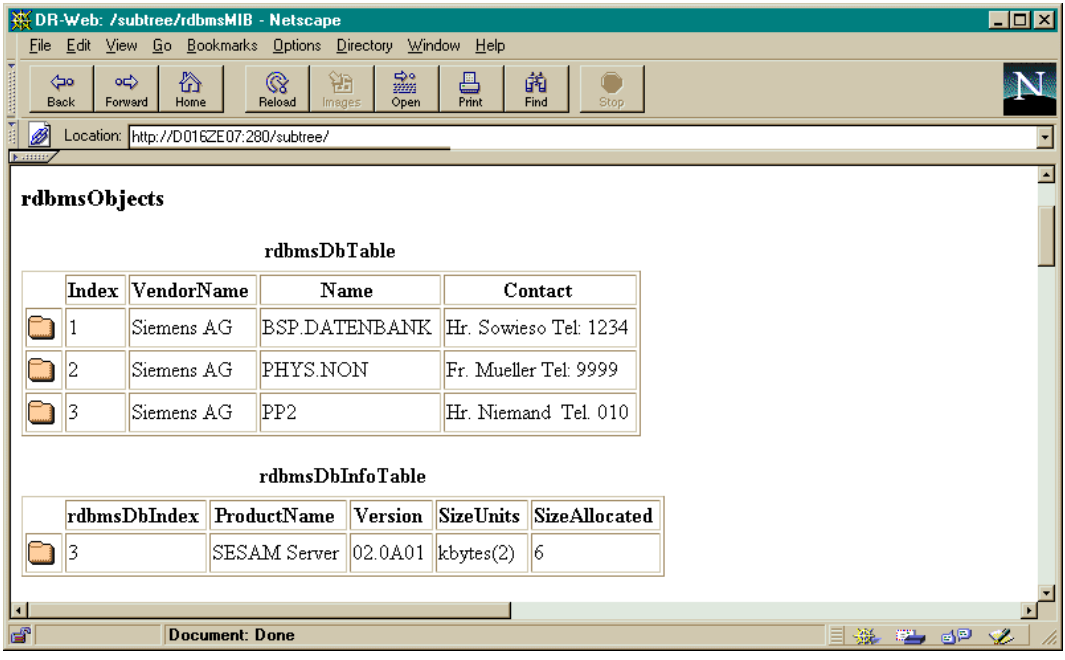


Figure 30: Overview of installed SESAM databases

Additional information about databases currently processed at a server

Object name	Access	Meaning
rdbmsDbInfoProductName	read-only	The textual product name of the server that created or last restructured this database. The format is product specific.
rdbmsDbInfoVersion	read-only	The version number of the server that created or last restructured this database. The format is product specific.
rdbmsDbInfoSizeUnits	read-only	INTEGER bytes(1), kbytes(2), mbytes(3), gbytes(4), tbytes(5) Identification of the units used to measure the size of this database in rdbmsDbInfoSizeAllocated and rdbmsDbInfoSizeUsed. bytes(1) indicates individual bytes, kbytes(2) indicates units of kilobytes, mbytes(3) indicates units of megabytes, gbytes(4) indicates units of gigabytes, and tbytes(5) indicates units of terabytes. All are binary multiples -- 1K = 1024. If writable, changes here are reflected in the get values of the associated objects.
rdbmsDbInfoSizeAllocated	read-write	The estimated size of this database (in rdbmsDbInfoSizeUnits), which is the disk space that has been allocated to it and is no longer available to users on this host. rdbmsDbInfoSize does not necessarily indicate the amount of space actually in use for database data. Some databases may support extending allocated size, and others may not. Note that the SESAM subagent agent does not need to allow write access to this object.

Table of installed servers (SESAM-DBH)

Object name	Access	Meaning
rdbmsSrvVendorName	read-only	The name of the vendor whose RDBMS manages this database, for informational purposes.
rdbmsSrvProductName	read-only	The product name of this server. This is normally the vendor‘s formal name for the product, in product specific format.
rdbmsSrvContact	read-write	The textual identification of the contact person for this managed server, together with information on how to contact this person.

Additional information on currently active servers

Object name	Access	Meaning
rdbmsSrvInfoStartupTime	read-only	Date and time the server was last started.
rdbmsSrvInfoFinishedTransactions	read-only	The number of transactions visible for this server and which were completed either with commit or abor.t Some database cooperations, e.g. read-only queries, may cause a failure to create the transaction.
rdbmsSrvInfoDiskReads	read-only	The total number of reads for database files since this server was started by the operation system. The numbers are not comparable across product boundaries. The calculation and definition of reads is product-specific.
rdbmsSrvInfoLogicalReads	read-only	The total number of logical read of database files made internally since this server was started. The values of this object and of rdbmsSrvInfoDiskReads underline the effect of caching on read operations. The numbers are not comparable across product boundaries. They are definitive only when calculated for all servers that use the cache commonly.
rdbmsSrvInfoDiskWrites	read-only	The total number of writes for database files since this server was started by the operation system. The numbers are not comparable across product boundaries.

Object name	Access	Meaning
rdbmsSrvInfoLogicalWrites	read-only	Indicates how often parts of the database files are marked “dirty” and the necessity recognized to write them to disk. This value and the value of rdbmsSrvInfoDiskWrites indicate how effective “write-behind” strategies are in reducing the frequency of writ-to-disk operations (compared to database operations). Since the writes can come from different servers than the ones that marked parts of the database files as “dirty”, these values are definitive only when calculated for all servers that use the cache commonly The numbers are not comparable across product boundaries.
rdbmsSrvInfoHandledRequests	read-only	The total number of requests made to the server on inbound associations. The meaning of “requests” is product specific, and is not comparable between products. This is intended to encapsulate high level semantic operations between clients and servers, or between peers. For instance, one request might correspond to a “select” or an “insert” statement. It is not intended to capture disk i/o described in rdbmsSrvInfoDiskReads and rdbmsSrvInfoDiskWrites.

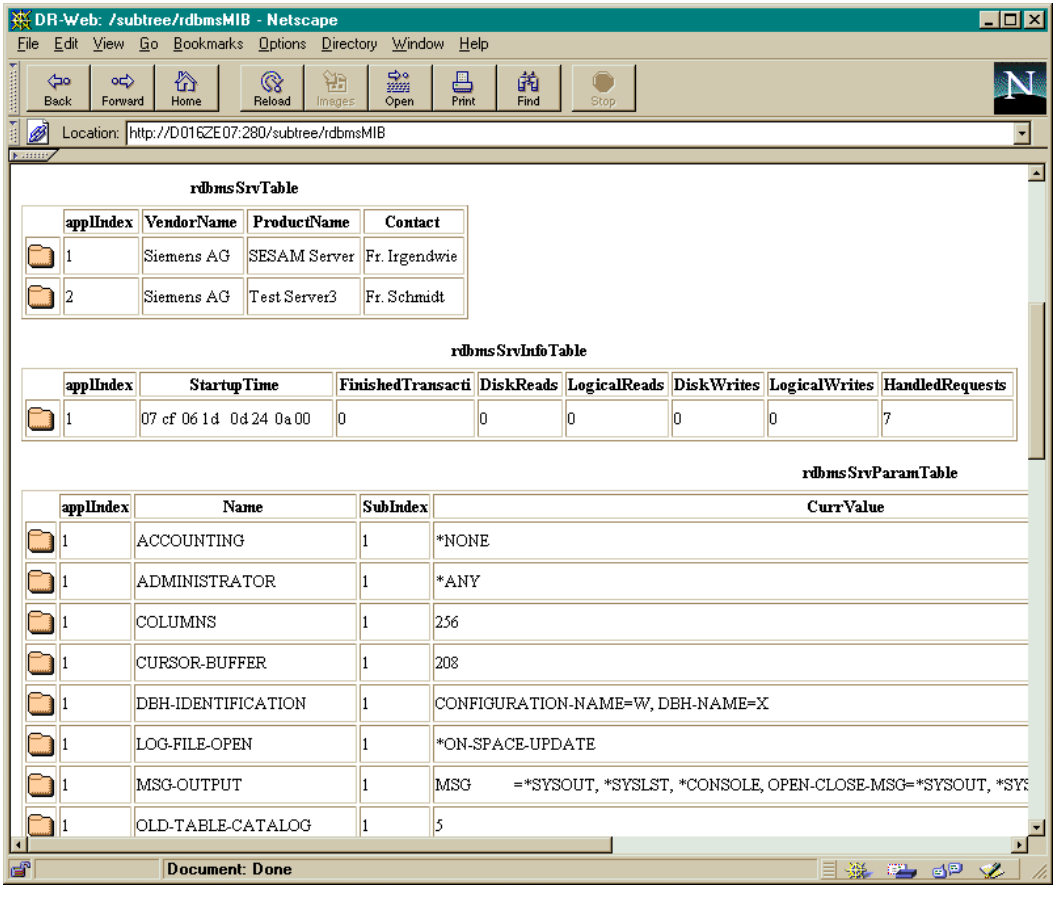


Figure 31: Display of currently installed servers with additional information on the currently active servers

Limits

Object name	Access	Meaning
rdbmsSrvLimitedResourceLimit	read-write	The maximum value the resource use may attain. The SESAM subagent does not permit access to this object.
rdbmsSrvLimitedResourceCurrent	read-only	The current value for the resource.
rdbmsSrvLimitedResourceHighwater	read-only	The maximum value of the resource seen since applUpTime was reset.

Limits

ResourceName	ResourceLimit	ResourceCurrent	ResourceHighwater
CURSORS	X	X	
PLANS	X	X	
SERVICE-TASKS	X	X	
SPACES		X	
SUBORDERS	X	X	X
THREADS		X	
USERS		X	

The limits described by the DBH load parameters listed above are supported for each active server (SESAM-DBH) entered in the configuration file.

Displaying the current DBH load parameters for active servers

Object name	Access	Meaning
rdbmsSrvParamCurrValue	read-write	The value for a configuration parameter now in effect, the actual setting for the server. While there may multiple values in the temporal domain of interest (for instance, the value to take effect at the next restart), this is the current setting. The SESAM subagent does not permit write access to this object.
rdbmsSrvParamComment	read-write	Annotation which describes the purpose of a configuration parameter or the reason for a particular parameter‘s setting. The SESAM subagent does not permit write access to this object.

Display of current DBH load parameters for active servers

Relation between databases and servers

Object name	Access	Meaning
rdbmsRelState	read-only	INTEGER other(1), active(2), available(3), restricted(4), unavailable(5) The state of this server‘s access to this database. Active(2) means the server is actively using the database. Available(3) means the server could use the database if necessary. Restricted(4) means the database is in some administratively determined state of less-than-complete availability. Unavailable(5) means the database is not available through this server. Other(1) means the database/server is in some other condition, possibly described in the vendor private MIB.

Traps

Object name	Trap No.	Meaning
Enterprise = rdbmsMIB.2		
rdbmsStateChange	1	An rdbmsStateChange trap means that one of the database servers/databases managed by this agent has a changed rdbmsRelState, which restricts its availability. In these cases, both active(2) and available(3) are deemed to have unrestricted access. The status sent with the trap is the new status with lower availability.

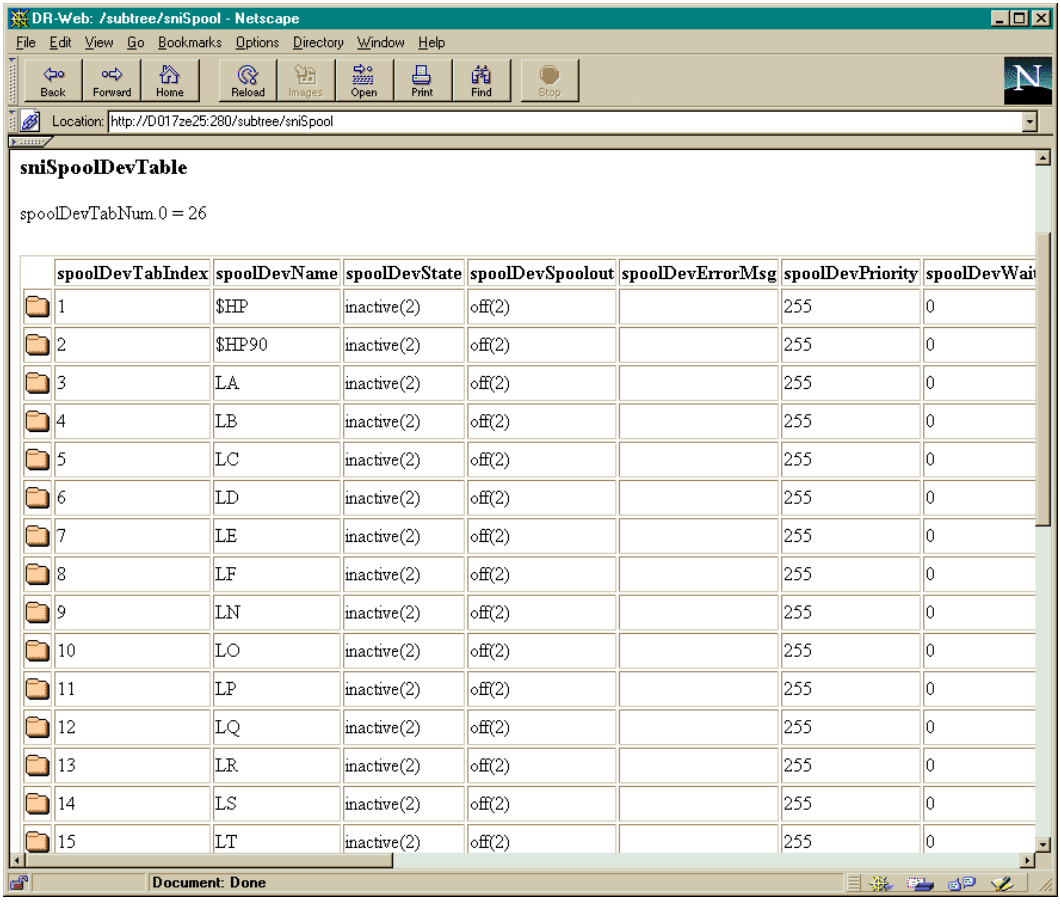
6.9 SNMP management for Spool & Print Service

The SPOOL and RSO devices are monitored by the spool and print services subagent, which supplies information about devices and print jobs.The print service agent is supplied with a proprietary MIB that is identical to the SINIX spool MIB. The device and job groups for BS2000/OSD are provided by this MIB.

Print device management

MIB definition	Access	Meaning
spoolDevTabNum	read-only	Number of table elements
spoolDevTabIndex	read-only	Index
spoolDevName	read-only	Name
spoolDevState	read-only	Status
spoolDevSpoolout	read-only	Spoolout
spoolDevErrorMsg	read-only	Error message
spoolDevPriority	read-only	Priority
spoolDevWaitingJobs	read-only	Pending jobs
spoolDevCurForm	read-only	Form
spoolDevActJid	read-only	Print job
spoolDevHost	read-only	Host
spoolDevAdmin	read-only	Manager
spoolDevDftForm	read-only	Default form
spoolDevAdmComment	read-only	Remarks
spoolDevEnablePoll	read-only	Poll option

Device group



spoolDevTabIndex	spoolDevName	spoolDevState	spoolDevSpoolout	spoolDevErrorMsg	spoolDevPriority	spoolDevWait
1	$HP	inactive(2)	off(2)		255	0
2	$HP90	inactive(2)	off(2)		255	0
3	LA	inactive(2)	off(2)		255	0
4	LB	inactive(2)	off(2)		255	0
5	LC	inactive(2)	off(2)		255	0
6	LD	inactive(2)	off(2)		255	0
7	LE	inactive(2)	off(2)		255	0
8	LF	inactive(2)	off(2)		255	0
9	LN	inactive(2)	off(2)		255	0
10	LO	inactive(2)	off(2)		255	0
11	LP	inactive(2)	off(2)		255	0
12	LQ	inactive(2)	off(2)		255	0
13	LR	inactive(2)	off(2)		255	0
14	LS	inactive(2)	off(2)		255	0
15	LT	inactive(2)	off(2)		255	0

Figure 32: Print device management table

Print job management

MIB definition	Access	Meaning
spoolJobTabNum	read-only	Number of table elements
spoolJobTabIndex	read-only	Index
spoolJobGlobalJid	read-only	Global job ID
spoolJobComment	read-only	Comment
spoolJobOriginator	read-only	Originator
spoolJobOrigHost	read-only	Origination host
spoolJobDestination	read-only	Target printer
spoolJobFileList	read-only	File list
spoolJobPriority	read-only	Priority
spoolJobTotalSize	read-only	Size
spoolJobRawMode	read-only	Raw mode
spoolJobDevName	read-only	Printer
spoolJobState	read-only	Status
spoolJobErrorMsg	read-only	Error message
spoolJobRqCopies	read-only	Copies requested
spoolJobPrCopies	read-only	Copies printed
spoolJobPrPercent	read-only	printed (in percent)

Job group

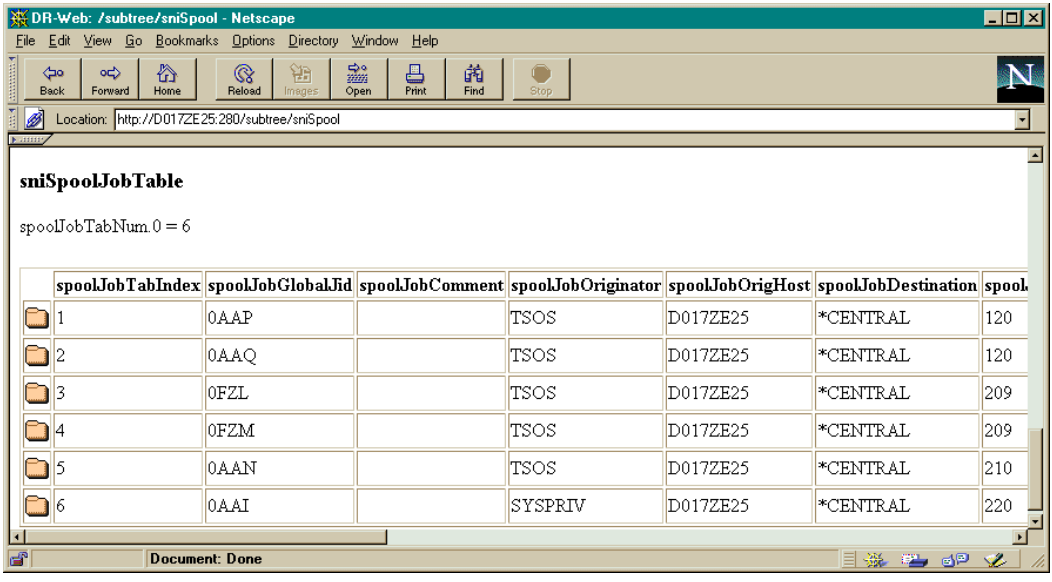


Figure 33: Print job management table

6.10 SNMP management for storage management

The subagent for storage management supplies information about pubsets and disks as well as on the availability of the storage management products HSMS, MAREN, TLS and ROBAR. Correspondingly, a proprietary MIB is supplied with the subagent, which contains the following information in addition to the global data of the storage management subagent:

– general information about HSMS, MAREN, ROBAR and TLS,
– resource information,
– display of all pubsets in a table
– display of all disks in a table

Global data of the storage management subagent

MIB definition	Access	Meaning
storMgmtGlobalDataVersion	read-only	Version of the subagent
storMgmtGlobalDataInputFile	read-write	Name of the input file

General information on HSMS, MAREN, ROBAR and TLS

MIB definition	Access	Meaning
storMgmtProductTabNum	read-only	Number of table elements
storMgmtProductIndex	read-only	Index
storMgmtProductName	read-only	Name
storMgmtProductVersion	read-only	Version
storMgmtProductState	read-only	Subsystem status

Product group

The product information is displayed in a table containing index, name, version and state information for the HSMS, MAREN, TLS and ROBAR products. The name of the robot archive (storage site) is also output for ROBAR.

A distinction is made between the following subsystem states values (see page 185):

- created
- not-created
- in-delete
- in-create
- in-resume
- in-hold
- not-resumed
- locked
- not-installed

The highest version is displayed for each of the MAREN, HSMS and TLS subsystems whose state is not equal to not-created. An entry without a version is returned if all versions of a subsystem are in the not-created state.

The failure of one of the subsystems HSMS, MAREN or TLS cannot currently be passed via traps to the management station.

If the status of a monitored pubset or a monitored disk changes, a trap is sent to the management station.

Example:

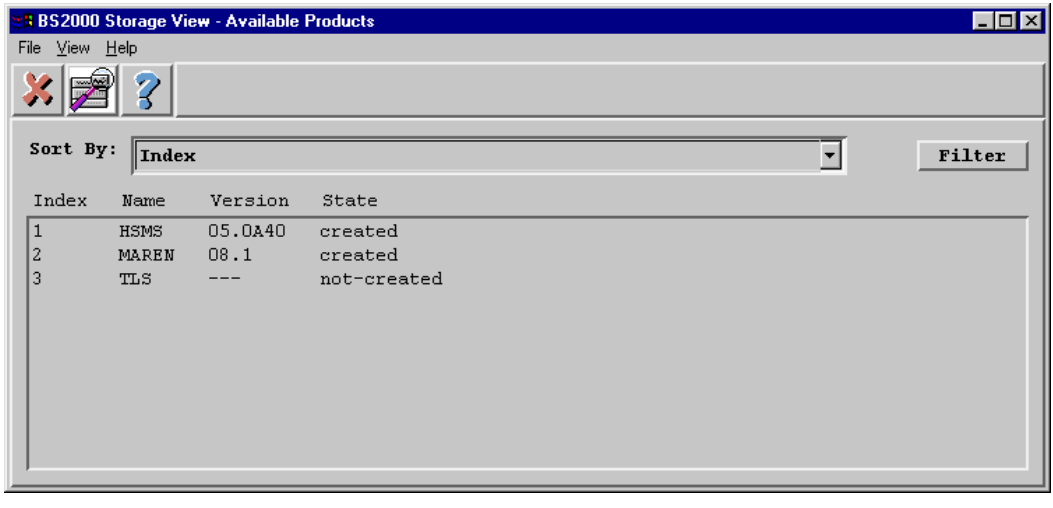


Figure 34: Overview of available products

Resource information

MIB definition	Access	Meaning
storMgmtResourcePubset	read-write	Pubset (Cat-ID)
storMgmtResourceSaturation	read-only	Current saturation level level-0 to level-5 or unknown-level*
storMgmtResourceCapacity	read-only	Capacity (in half pages (HP)
storMgmtResourceSpaceAllocated	read-only	Allocated memory (number of allocated HPs)
storMgmtResourceFragment	read-only	Degree of fragmentation
storMgmtResourceReusableS1	read-only	Number of backup files; Number of PAM pages used by these backup files; Number of un-used PAM pages
storMgmtResourceSecureQueue	read-only	Number of waiting tasks in the SECURE queue

Resource group
*The value “unknown-level” is output if no information can be obtained for the specified pubset. *storMgmtResourceCapacity.0* and *storMgmtResourceSpaceAllocated.0* have the value 1 in this case.

Specific information can be requested for a selected pubset. A pubset is selected by entry of the catalog ID in uppercase for the *storMgmtResourcePubset.0* object. The pubset information is displayed in a table.

Display of all pubsets in a table

The storage management subagent enables all pubsets to be displayed in a table. In addition, it is also possible to monitor the saturation level of individual pubsets. To do this, these pubsets must be defined in the relevant input file when configuring storage management subagent (see page 85). When the saturation level changes, the storage management subagent sends a trap with the specified community string (see page 297).

MIB definition	Access	Meaning
storMgmtPubsetTabNum	read-only	Number of table entries
storMgmtPubsetTabState	read-write	Status of the pubsets in the pubset table: – all – paging – local – remote – accessible – local-accessible – shared – exclusive – remote-accessible – local-accessible-speedcat – xcs-pubset – hsms-supported – single-feature – system-managed – volume-sets – unused-volsets – master-change-error These values can be set to modify the table output.
Table:		
storMgmtPubsetIndex	read-only	Unique value for each table entry (CatID of the associated pubset)
storMgmtPubsetTyp	read-only	Type of pubsets: – single-featured – system-managed – volumeset – unknown
storMgmtPubsetLocal	read-only	Indicates whether pubset is *local* or *remote*
storMgmtPubsetHome	read-only	Indicates whether pubset is *home* or *imported*
storMgmtPubsetShared	read-only	Indicates whether pubset is *shared* or *exclusive*

MIB definition	Access	Meaning
storMgmtPubsetMaster	read-only	Indicates whether pubset is master or slave
storMgmtPubsetAccessible	read-only	Indicates whether pubset is accessible or not
storMgmtPubsetQuiet	read-only	Indicates whether pubset is a “Quiet Pubset“
storMgmtPubsetPaging	read-only	Indicates whether pubset is a “Paging Pubset“
storMgmtPubsetSize	read-only	Size of pubset
storMgmtPubsetUsedSize	read-only	Space occupied by pubset
storMgmtPubsetSaturationLevel	read-write	Saturation level for pubset

Pubset table

Example:

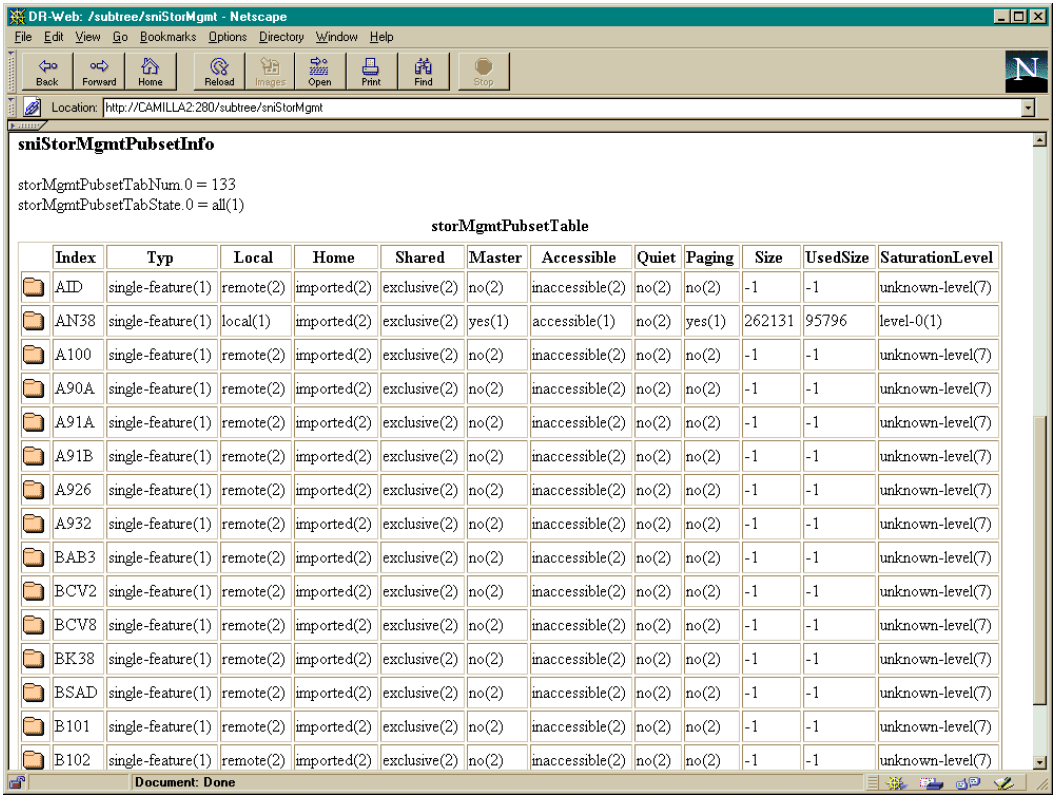


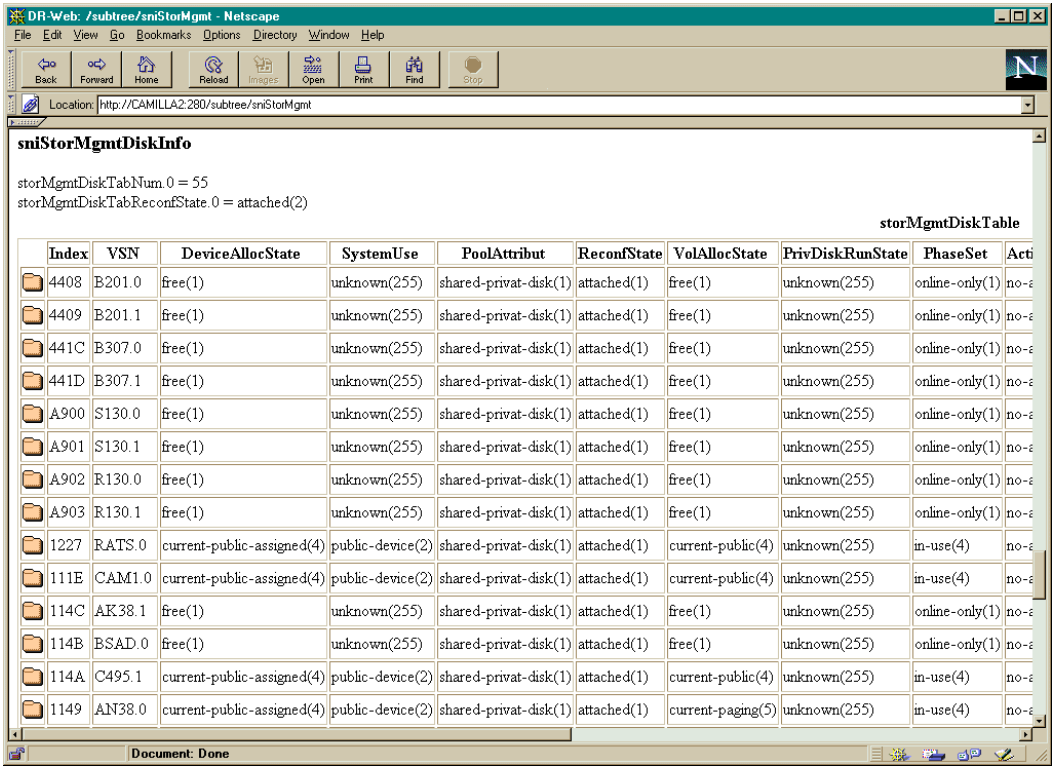
Figure 35: Display of all pubsets

Display of all disks in a table

The storage management subagent enables all disks to be displayed in a table. In addition, it is also possible to monitor the reconfirmation state of individual disks. To do this, these disks must be defined in the relevant input file when configuring storage management subagent (see page 85). When the reconfiguration state changes, the storage management subagent sends a trap with the specified community string (see page 297).

MIB definition	Access	Meaning
storMgmtDiskTabNum	read-only	Number of table entries
storMgmtDiskTabReconfState	read-write	Reconfiguration state of the disk displayed: all attached detached other These values can be set to modify the table output.
storMgmtDiskIndex	read-only	Unique value for each table entry (mnemonic name of the disk)
storMgmtDiskVSN	read-only	Volume Serial Number (VSN) of disk
storMgmtDiskDeviceAllocState	read-only	Device allocation state of disk
storMgmtDiskSystemUse	read-only	Type of disk
storMgmtDiskPoolAttribut	read-only	Indicated whether the disk is *local* or *remote*
storMgmtDiskReconfState	read-only	Reconfiguration state of disk
storMgmtDiskVolAllocState	read-only	Volume allocation state of disk
storMgmtDiskPrivDiskRunState	read-only	Private disk run state of disk
storMgmtDiskPhaseSet	read-only	Phase set of disk
storMgmtDiskActionState	read-only	Action state of disk
storMgmtDiskUse	read-only	Disk use
storMgmtDiskAssignTime	read-only	Assign time of disk
storMgmtDiskUserAllocation	read-only	User allocation of disk
storMgmtDiskOperatorControl	read-only	Operator control of disk
storMgmtDiskSystemAllocation	read-only	System allocation of disk
storMgmtDiskAccess	read-only	Disk access of disk
storMgmtDiskRecordingMode	read-only	Disk recording mode of disk

Example:



Index	VSN	DeviceAllocState	SystemUse	PoolAttribut	ReconfState	VolAllocState	PrivDiskRunState	PhaseSet	Acti
4408	B201.0	free(1)	unknown(255)	shared-privat-disk(1)	attached(1)	free(1)	unknown(255)	online-only(1)	no-a
4409	B201.1	free(1)	unknown(255)	shared-privat-disk(1)	attached(1)	free(1)	unknown(255)	online-only(1)	no-a
441C	B307.0	free(1)	unknown(255)	shared-privat-disk(1)	attached(1)	free(1)	unknown(255)	online-only(1)	no-a
441D	B307.1	free(1)	unknown(255)	shared-privat-disk(1)	attached(1)	free(1)	unknown(255)	online-only(1)	no-a
A900	S130.0	free(1)	unknown(255)	shared-privat-disk(1)	attached(1)	free(1)	unknown(255)	online-only(1)	no-a
A901	S130.1	free(1)	unknown(255)	shared-privat-disk(1)	attached(1)	free(1)	unknown(255)	online-only(1)	no-a
A902	R130.0	free(1)	unknown(255)	shared-privat-disk(1)	attached(1)	free(1)	unknown(255)	online-only(1)	no-a
A903	R130.1	free(1)	unknown(255)	shared-privat-disk(1)	attached(1)	free(1)	unknown(255)	online-only(1)	no-a
1227	RATS.0	current-public-assigned(4)	public-device(2)	shared-privat-disk(1)	attached(1)	current-public(4)	unknown(255)	in-use(4)	no-a
111E	CAM1.0	current-public-assigned(4)	public-device(2)	shared-privat-disk(1)	attached(1)	current-public(4)	unknown(255)	in-use(4)	no-a
114C	AK38.1	free(1)	unknown(255)	shared-privat-disk(1)	attached(1)	free(1)	unknown(255)	online-only(1)	no-a
114B	BSAD.0	free(1)	unknown(255)	shared-privat-disk(1)	attached(1)	free(1)	unknown(255)	online-only(1)	no-a
114A	C495.1	current-public-assigned(4)	public-device(2)	shared-privat-disk(1)	attached(1)	current-public(4)	unknown(255)	in-use(4)	no-a
1149	AN38.0	current-public-assigned(4)	public-device(2)	shared-privat-disk(1)	attached(1)	current-paging(5)	unknown(255)	in-use(4)	no-a

Figure 36: Display of all disks

Traps

There is a trap for

– saturation level (pubset monitoring)
– reconfiguration state (disk monitoring)

Trap for saturation level (pubset monitoring)

Object name	Trap No.	Meaning
sniStorMgmtPubsetTraps (Enterprise=1.3.6.1.4.1.231.1.20.20)		
storMgmtPubsetSatLevTrap	301	Saturation level *x* reached

Trap for reconfiguration state (disk monitoring)

Object name	Trap No.	Meaning
sniStorMgmtDiskTraps (Enterprise=1.3.6.1.4.1.231.1.20.21)		
storMgmtDiskReconfStateTrap	301	Disk reconfiguration state *x* reached

7 SNMP management for extended performance monitoring with SM2

The SM2-based performance subagent SSA-SM2-BS2 is available for BS2000 systems as of BS2000/OSD V2.0. SSA-SM2-BS2 supplies basic information on SM2 itself, i.e. on subsystem status, version, measurement interval size and sampling cycle. The actual measurement values correspond to the 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 of *open*UTM applications
- resource usage of separate tasks

The display of returned measurement values on the management station can be supported by the management applications PMBS2 (Reliant UNIX, Windows NT) and SMAWpmbs2 (Solaris), which also enable the simultaneous monitoring of several BS2000/OSD systems. The management applications PMBS2 and SMAWpmbs2 are included on the CD-ROM enclosed with the SBA-BS2 product. Please refer to page 379 for a detailed description.

SM2 parameters and basic values

Object	Access	Meaning
Group SM2 Params		
sm2Status	read-only	Status of the measurement subsystem SM2
sm2Version	read-only	Version of the measurement subsystem SM2: format Vnn.nAnn
sm2Interval	read-write	Online cycle of the measurement subsystem SM2 in units of a second, also called measurement interval Range: 10 - 3600 (default value: 120)
sm2SamplingCycle	read-write	sampling cycle of the measurement subsystem SM2 in milliseconds Range: 200 - 10000 (default value: 800)
Group SM2 Basic		
sm2BasicStatus	read-only	Status associated with BASIC buffer
sm2BasicTime	read-only	Time at the end of the last measurement interval DateAndTime in accordance with RFC1514
sm2BasicTimeString	read-only	Time at the end of the last measurement interval. Date and Time in a direct displayable format: YYYY-MM-DD,hh:mm:ss.d[,shh:mm] where YYYY-MM-DD is local date in the order year-month-day, hh:mm:ss.d is local time with hour-minutes-seconds.deciseconds, shh:mm is (-|+)hours:minutes from UTC
sm2BasicSamples	read-only	Number of samples taken place within last measurement interval
sm2BasicMaxLogMach	read-only	Number of logical machines
sm2BasicVM2000	read-only	Info about VM2000 activity (no-data, inactive, active)

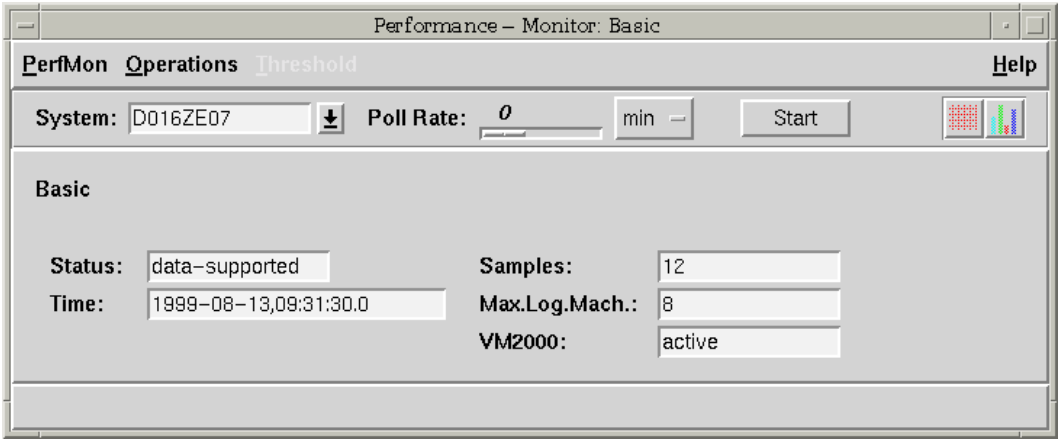
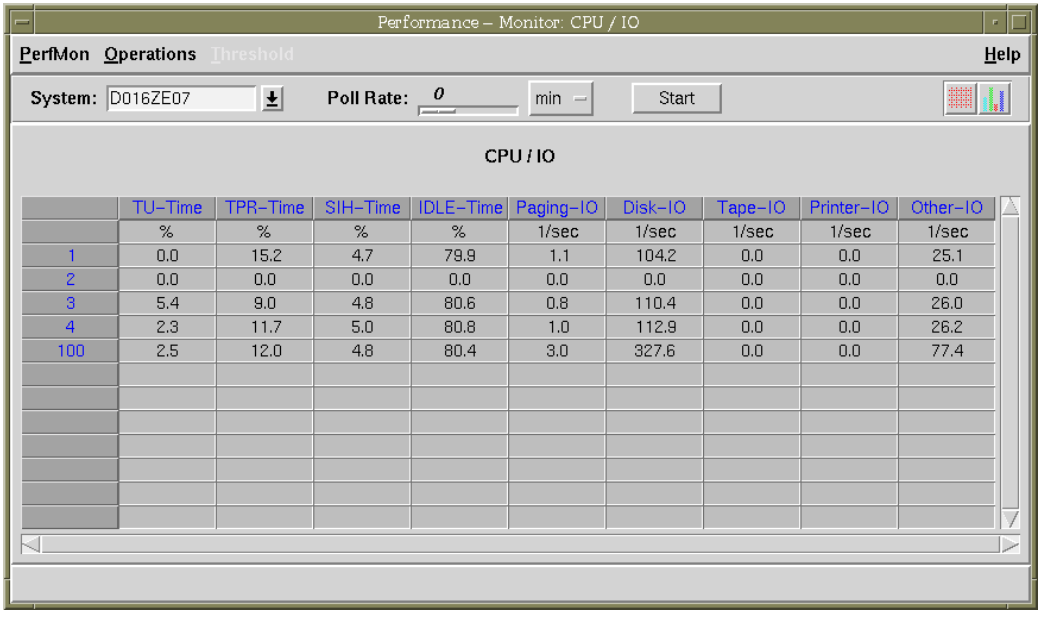


Figure 37: Display of SM2 parameters by the Performance Monitor

I/O values

Object	Access	Meaning
sm2TimeIOStatus	read-only	Status associated with TIME IO buffer (data-supported, data-invalid, prg-inactive, no-subsystem, sm2-not-running, unknown)
sm2TimeIOActMach	read-only	Number of active logical machines
sm2TimeIOMachTabNumber	read-only	Number of entries in following machine table
sm2TimeIOMachTabIndex	read-only	The index starts with 1 and uniquely identifies each entry. Entry “average” (index 100) will contain average values for all provided measurement times and sums for the remaining IO counters over all logical machines.
sm2TimeIOMachTabIdleTime	read-only	Time during which the logical machine was not active (not halted) (share of time in parts per thousand)
sm2TimeIOMachTabTUTime	read-only	TPR time of the logical machine (share of time in parts per thousand)
sm2TimeIOMachTabTPRTime	read-only	SIH time of the logical machine (share of time in parts per thousand)
sm2TimeIOMachTabSIHTime	read-only	Stop time of the logical machine (share of time in parts per thousand)
sm2TimeIOMachTabStopTime	read-only	Rate of paging IO concerning the logical machine in (number of paging IOs per second) * 10
sm2TimeIOMachTabPagingIO	read-only	Rate of disk IO concerning the logical machine in (number of disk IOs per second) * 10
sm2TimeIOMachTabDiskIO	read-only	Rate of tape IO concerning the logical machine in (number of tape IOs per second) * 10
sm2TimeIOMachTabTapeIO	read-only	Rate of printer IO concerning the logical machine in (number of printer IOs per second) * 10
sm2TimeIOMachTabPrinterIO	read-only	Rate of other IO concerning the logical machine in (number of other IOs per second) * 10
sm2TimeIOMachTabOtherIO	read-only	Rate of the other IOs concerning the logical machine in (number of other IOs per second) * 10



	TU–Time	TPR–Time	SIH–Time	IDLE–Time	Paging–IO	Disk–IO	Tape–IO	Printer–IO	Other–IO
	%	%	%	%	1/sec	1/sec	1/sec	1/sec	1/sec
1	0.0	15.2	4.7	79.9	1.1	104.2	0.0	0.0	25.1
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	5.4	9.0	4.8	80.6	0.8	110.4	0.0	0.0	26.0
4	2.3	11.7	5.0	80.8	1.0	112.9	0.0	0.0	26.2
100	2.5	12.0	4.8	80.4	3.0	327.6	0.0	0.0	77.4

Figure 38: Display of CPU load and IO rate

Main memory load and virtual address space

Object	Access	Meaning
sm2MemoryStatus	read-only	Status associated with memory buffer (data-supported, data-invalid, prg-inactive, no-subsystem, sm2-not-running, unknown)
sm2MemorySize	read-only	Total size of main memory in kilo bytes (kB)
sm2MemoryPageableSize	read-only	Size of pageable memory in kilo bytes (kB)
sm2MemoryFreeReadSize	read-only	Size of free read-only memory (pageable) in kilo bytes (kB)
sm2MemoryFreeReadWriteSize	read-only	Size of free read/write memory (pageable) in kilo bytes (kB)
sm2MemoryPagingAreaTotal	read-only	Total size of paging area (incl. ES/GS) in kilo bytes (kB)
sm2MemoryPagingAreaESGS	read-only	Size of ES/GS paging area in kilo bytes (kB)
sm2MemoryPagingAreaFree	read-only	Size of free paging area in kilo bytes (kB)
sm2MemoryPageFaults	read-only	Total number of page faults interrupts per second * 10
sm2MemoryPage1stFaults	read-only	Number of page faults interrupts for the first access to a page per second * 10
sm2MemoryPageReclaims	read-only	Number of page faults interrupts for which the addressed page is still in memory per second * 10
sm2MemoryPageReads	read-only	Number of pages read from background storage per second * 10 (PAGE READS)
sm2MemoryPageWrites	read-only	Number of pages written to background storage per second * 10 (PAGE WRITES)
sm2MemoryPageReadESGS	read-only	Number of pages read from expanded (ES) or global storage (GS) per second * 10
sm2MemoryPageWriteESGS	read-only	Number of pages written to expanded (ES) or global storage (GS) per second * 10

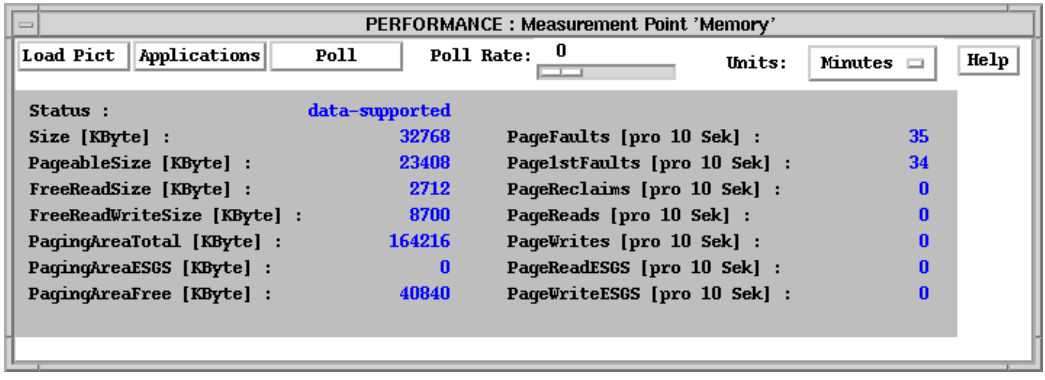


Figure 39: Performance Monitor: measuring range “Memory“

Main memory allocation by standard categories of tasks

Object	Access	Meaning
sm2CategoryStatus	read-only	Status associated with Category buffer (data-supported, data-invalid, prg-inactive, no-subsystem, sm2-not-running, unknown)
sm2CategorySystemTasks	read-only	Average number of system tasks *10
sm2CategoryDialogTasks	read-only	Average number of dialog tasks *10
sm2CategoryBatchTasks	read-only	Average number of batch tasks *10
sm2CategoryTPTasks	read-only	Average number of tp tasks *10

Device-specific values

Object	Access	Meaning
sm2SDeviceStatus	read-only	Status assigned to the device buffer.
sm2SDeviceRealNumber	read-only	Actual number of device available
sm2SDeviceTabNumber	read-only	Number of entries in following table; default: 10 entries
sm2SDeviceTabIndex	read-only	Uniquely identifies each entry of this table
sm2SDeviceTabVSN	read-only	Volume sequence number VSN: (volume name) The entities of the DEVICES object group are sorted according to the umber of I/O operations for each measuring time. The first entity corresponds to the device with the most I/O operations in the measuring period.
sm2SDeviceTabMnemonic	read-only	Mnemonic device name
sm2SDeviceTabType	read-only	Device type gained using macro NKGTYPE
sm2SDeviceTabIO	read-only	Number of IO operations performed physically per second * 10
sm2SDeviceTabBusyDms	read-only	Device activity (excluding paging) in o/oo (share of time in parts per thousand)
sm2SDeviceTabBusyPaging	read-only	Device activity due to paging in o/oo (share of time in parts per thousand)

Application-specific data of *open*UTM applications

Object	Access	Meaning
sm2UTMStatus	read-only	Status associated with UTM buffer
sm2UTMTabNumber	read-only	Number of entries in following table
sm2UTMTabIndex	read-only	Uniquely identifies the individual entries in this table The entities of the UTM object group are sorted according to the number of dialog steps for each measuring time. The first entity corresponds to the device with the most dialog steps in the measuring period.
sm2UTMTabApplName	read-only	Name of UTM application
sm2UTMTabUTMVersion	read-only	Version of subsystem. Format: Vnn.nAnn
sm2UTMTabApplMode	read-only	UTM application mode
sm2UTMTabTasksRunning	read-only	Number of tasks running for this UTM application
sm2UTMTabMaxAsyncTasks	read-only	Maximal number of tasks for asynchronous processing
sm2UTMTabConnectedUsers	read-only	Number of currently connected users
sm2UTMTabCurrConvDial	read-only	Number of active dialog conversations
sm2UTMTabCurrConvAsync	read-only	Number of active asynchronous conversations
sm2UTMTabWaitingATACS	read-only	Number of waiting asynchronous transactions, buffered but not yet processed
sm2UTMTabCacheHitRate	read-only	UTM cache hit rate (parts per thousand)
sm2UTMTabFreePagePool	read-only	Percentage of free pages in UTM pagepool *10d (in parts per thousand)
sm2UTMTabDialTACS	read-only	Number of (finished) dialog transactions per second *10
sm2UTMTabAsyncTACS	read-only	Number of (finished) asynchronous transactions (ATACS) per second *10
sm2UTMTabDialTotalTime	read-only	Average time period of a dialog step (UTM response time) in seconds *10
sm2UTMTabDialTotalTimeDB	read-only	Average time period of a dialog step (UTM response time) in seconds *10 Only dialog steps with data base accesses are considered

Object	Access	Meaning
sm2UTMTabDialDBTime	read-only	Average time per dialog step UTM is waiting for data base responses in seconds *10. Only dialog steps with data base accesses are considered
sm2UTMTabDialDBCall	read-only	Average number of data base calls per dialog step. Only dialog steps with data base accesses are considered
sm2UTMTabDialDBCpuTime	read-only	Average CPU time per dialog step data base/s is/are processing in seconds *10. Only dialog steps with data base accesses are considered
sm2UTMTabDialDBIO	read-only	Average number of data base IOs per dialog step. Only dialog steps with data base accesses are considered
sm2UTMTabDialUTMCpuTime	read-only	Average CPU time per dialog step UTM tasks are consuming in seconds *10
sm2UTMTabDialUTMIO	read-only	Average number of IOs done by UTM tasks per dialog step
sm2UTMTabAsyncTotalTime	read-only	Average Time period of a asynchronous conversation (UTM response time) in seconds *10
sm2UTMTabAsyncTotalTimeDB	read-only	Average time period of a asynchronous conversation in seconds *10. Only asynchronous conversations with data base accesses are considered.
sm2UTMTabAsyncDBTime	read-only	Average time per asynchronous conversation UTM is waiting for data base responses in seconds *10. Only asynchronous conversations with data base accesses are considered.
sm2UTMTabAsyncDBCall	read-only	Average number of data base calls per asynchronous conversation. Only asynchronous conversations with data base accesses are considered.
sm2UTMTabAsyncDBCpuTime	read-only	Average CPU time per asynchronous conversation data base/s is/are processing in seconds *10. Only asynchronous conversations with data base accesses are considered.

Object	Access	Meaning
sm2UTMTabAsyncDBIO	read-only	Average number of data base IOs per asynchronous conversation. Only asynchronous conversations with data base accesses are considered.
sm2UTMTabAsyncUTMCpuTime	read-only	Average CPU time per asynchronous conversation UTM tasks are consuming in seconds *10
sm2UTMTabAsyncUTMIO	read-only	Average number of IOs done by UTM tasks per asynchronous conversation

Capacity values for individual tasks

Object	Access	Meaning
sm2PerTaskStatus	read-only	Status associated with Periodic Task buffer (data-supported, data-invalid, prg-inactive, no-subsystem, sm2-not-running, unknown)
sm2PerTaskRealNumber	read-only	Real number of available tasks
sm2PerTaskTabNumber	read-only	Number of entries in following table; current maximum value: 10 entries
sm2PerTaskTabIndex *	read-only	Uniquely identifies each entry of this table. The entities of the PERTEASK object group are sorted according to the CPU capacity for each measuring time. The first entity corresponds to the device with the highest CPU capacity in the measuring period
sm2PerTaskTabTSN	read-only	Task sequence number
sm2PerTaskTabUserID	read-only	User ID under which the task is running
sm2PerTaskTabJobName	read-only	Job name
sm2PerTaskTabType	read-only	Task type (no-data, system, dialog, batch, tp)
sm2PerTaskTabCPU	read-only	CPU utilization (share of time in parts per thousand)
sm2PerTaskTabIO	read-only	Number of IO operations per second * 10
sm2PerTaskTabUPG	read-only	Average number of Used Pages
sm2PerTaskTabServiceUnits	read-only	Number of service units utilized per second * 10
sm2PerTaskTabPageRead	read-only	Number of pages read per second * 10

8 SNMP management for monitoring *open*UTM and *open*UTM applications

The *open*UTM subagent SSA-OUTM-BS2 provides the following services:

- monitoring and controlling 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/unlocking UTM data terminals
- terminating an *open*UTM application

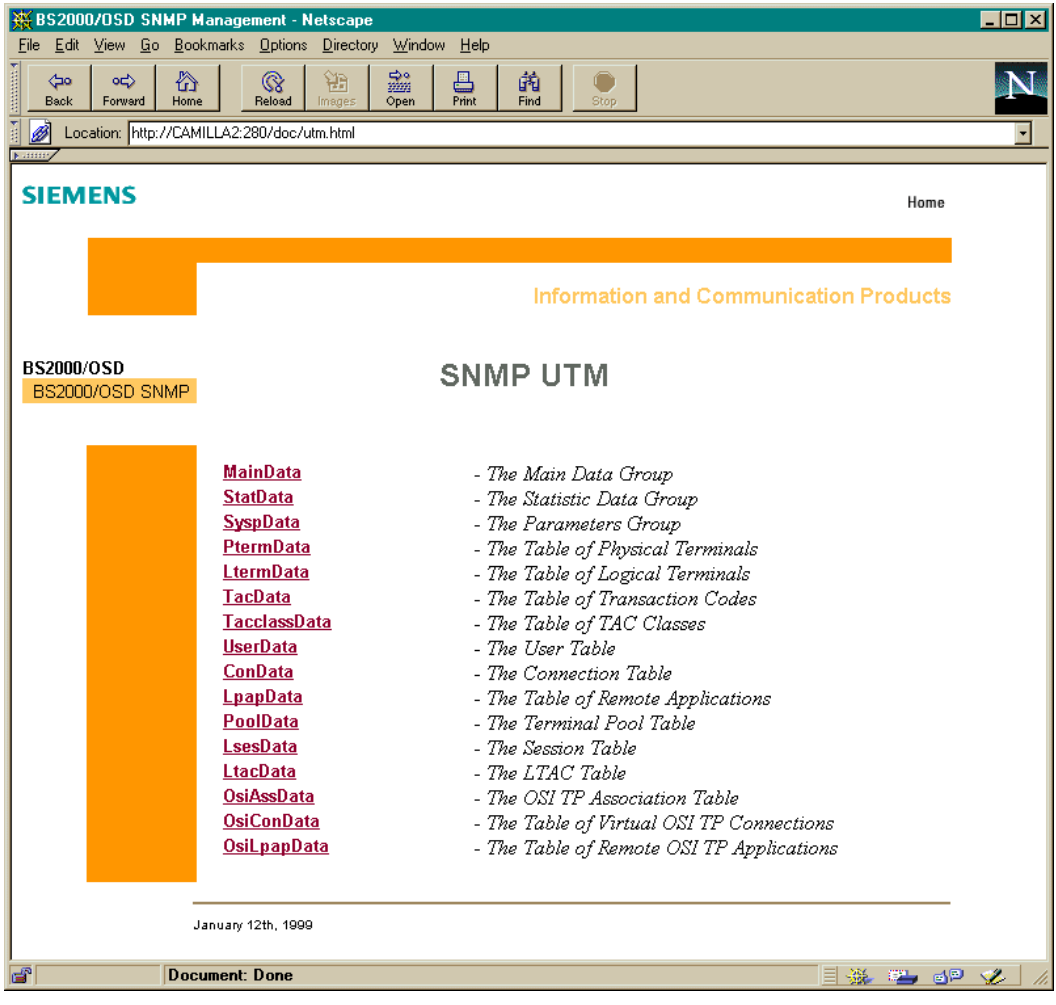


Figure 40: *open*UTM subagent: overview

Global data of the *open*UTM subagent

Object name	Access	Meaning
utmMainApplName	read-write	Name of the selected UTM application
utmMainBCAMAppl	read-write	BCAM application name of the selected UTM application
utmMainUTMversion	read-only	UTM version
utmMainApplStartStop	read-write	Reading: status of selected UTM application; writing: start (START) or terminate (STOP) selected UTM application (start, stop, undefined)
utmMainSubagentVersion	read-only	Version number of the SNMP subagent and the type of operating system

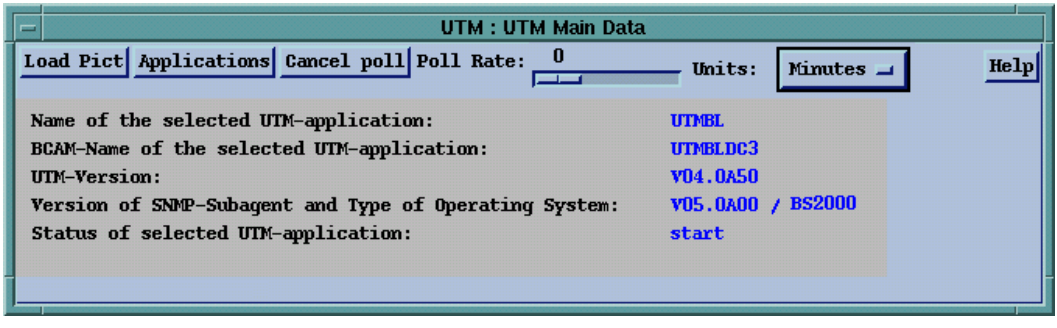


Figure 41: *open*UTM main parameters

General statistical information

Object name	Access	Meaning
utmStatStartDateAndTime	read-only	Date and time of the last cold start of the application (in DateAndTime format)
utmStatStartDateAndTimeString	read-only	Date and time of the last cold start of the application (printable)
utmStatTermInMsgs	read-only	Number of messages entered on all terminals since the completion of the last full hour
utmStatTermOutMsgs	read-only	Number of messages issued on all terminals since the completion of the last full hour
utmStatCurrTasks	read-only	Number of current tasks in this application
utmStatConnUsers	read-only	Number of connected users
utmStatOpenDialConv	read-only	Number of active dialog conversations
utmStatOpenAsynConv	read-only	Number of active asynchronous conversations
utmStatDialTAperSec	read-only	Number of dialog transactions per second
utmStatAsynTAperSec	read-only	Number of asynchronous transactions per second

Object name	Access	Meaning
utmStatDialStepSec	read-only	Number of dialog steps per second
utmStatMaxPoolSize	read-only	Maximum occupancy of the pagepool in percent
utmStatActPoolSize	read-only	Current occupancy of the pagepool in percent
utmStatAvgPoolSize	read-only	Average occupancy of the pagepool in percent
utmStatCacheHitRate	read-only	Hit rate in percent when searching a cache page
utmStatCacheWaits	read-only	Percentage of cache buffer requests leading to a waiting time
utmStatUnprocAtacs	read-only	Number of asynchronous transaction jobs, which are not yet processed
utmStatUnprocPrints	read-only	Number of waiting print jobs
utmStatWaitDPUTs	read-only	Number of pending time driven jobs
utmStatAbTermConv	read-only	Number of abnormally terminated conversations
utmStatResourcWaits	read-only	Relation between the number of resources requests with waiting and the total number of resources requests (in parts per thousand)
utmStatDeadlocks	read-only	Number of recognized and removed deadlocks
utmStatPeriodWrites	read-only	Number of periodic writes
utmStatPagesPWrite	read-only	Number of 2 KB pages being averagely saved by a periodic write
utmStatLogWrites	read-only	Number of write jobs to the user log file since the completion of the last hour
utmStatActJR	read-only	Current number of job receiving conversations, which are addressed at the same time
utmStatMaxJR	read-only	Maximum number of job receiving conversations, which have been addressed at the same time since KDCDEF generation

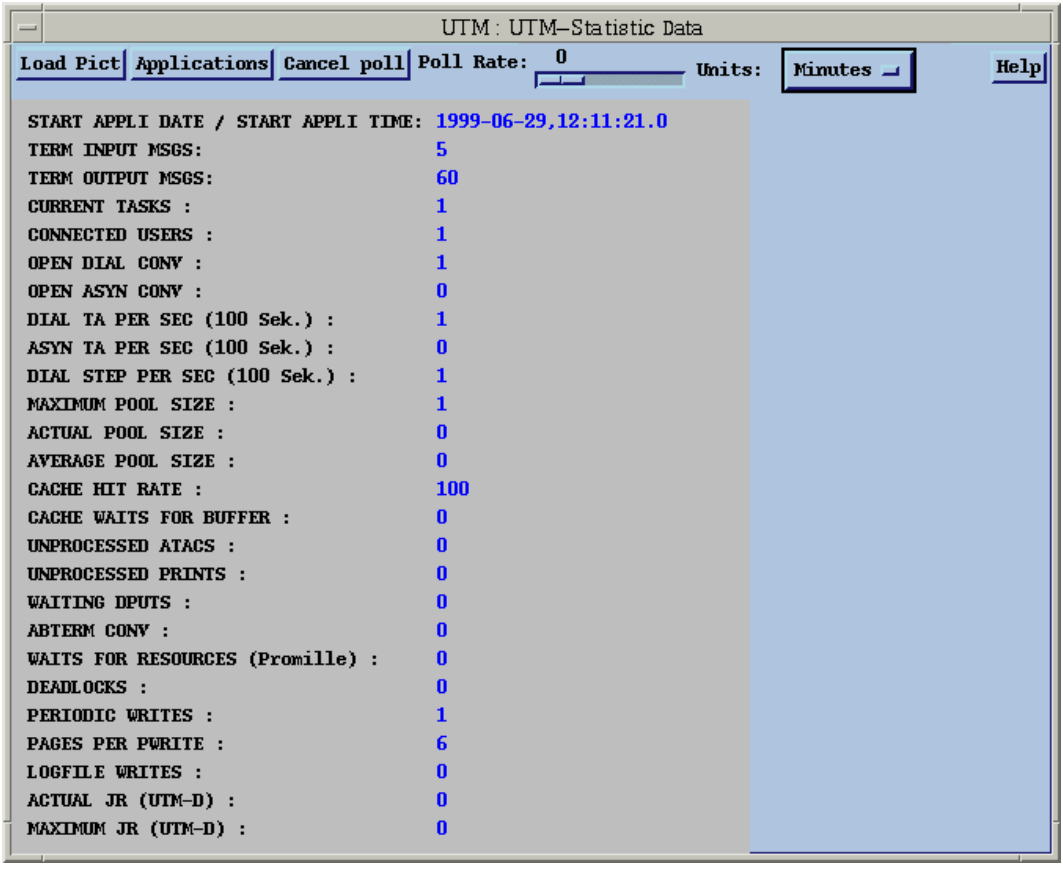


Figure 42: *open*UTM statistics

System parameters

Object name	Access	Meaning
utmSyspAccount	read-write	UTM accounting is either activated (ON) or not activated (OFF) (ON, OFF)
utmSyspCalcAccount	read-write	The calculation phase of UTM accounting is either activated (ON) or not activated (OFF) (ON, OFF)
utmSyspSM2	read-write	The delivery of data to SM2 is either activated (ON) or not activated (OFF) (ON, OFF)
utmSyspKDCMON	read-write	KDCMON is either activated (ON) or not activated (OFF) (ON, OFF)
utmSyspTestmode	read-write	Test mode is either activated (ON) or not activated (OFF) (ON, OFF)
utmSyspMaxPagRate	read-write	Percentage of cache pages which are to be written to KDCFILE in case of a bottleneck
utmSyspProgFGG	read-write	read: Number of the current file generation of the program write: -1 => old (load lower program generation) +1 => new (load higher program generation)
utmSyspTermWait	read-write	Maximum elapse time (seconds) between terminal output and the following input of the terminal user during a multi step conversation
utmSyspUsLogFGG	read-write	Number of the current file generation of the user log file
utmSyspResWaitTA	read-write	Maximum waiting time in seconds for a resource locked by another transaction
utmSyspMaxTasks	read-write	Maximum number of tasks which are allowed in this application
utmSyspResWaitPr	read-write	Maximum waiting time in seconds for a resource locked by another process
utmSyspCurrTasks	read-only	Number of current work processes of the application
utmSyspConRTime	read-write	Cycle in minutes, in which *open*UTM retries to create a logical connection
utmSyspMaxAsynTasks	read-write	Maximum number of tasks for asynchronous program
utmSyspLogAckwait	read-only	Maximum waiting time in seconds for a print or transport acknowledgment

Object name	Access	Meaning
utmSyspPTCTime	read-write	Maximum waiting time in seconds of a job receiving conversation in PTC for acknowledgment
utmSyspConcTime	read-write	Time in seconds permitted for creation of a session or an association
utmSyspPGWTTime	read-write	Maximum time in seconds permitted for the KDCS call PGWT
utmSyspTasksWaitPGWT	read-only	Current number of tasks being in a wait state by a PGWT call
utmSyspTasksinPGWT	read-write	Maximum number of task for PGWT calls

Table of physical data stations

Object name	Access	Meaning
utmPtermTabNum	read-only	The number of entries in the table of physical data terminals
utmPtermIndex	read-only	A unique value for each entry, its value ranges between 1 and the value of utmPtermTabNum
utmPtermName	read-only	Name of the chosen physical terminal
utmPtermProname	read-write	Processor name of the physical terminal (or blank for local devices)
utmPtermLterm	read-write	Name of the logical UTM terminal that belongs to the physical terminal
utmPtermBCAMAppl	read-write	BCAM name of the UTM application
utmPtermPtyp	read-only	Partner type of the UTM terminal
utmPtermStatus	read-write	The UTM terminal is either locked (OFF) or unlocked (ON) (ON, OFF)
utmPtermConnected	read-write	The UTM terminal is either connected (Y) not connected (N), or waiting for a connection (yes, no, waiting)
utmPtermConnectStatus	read-write	Status = “A” means automatic connection at application-start, status = “P” means a terminal pool connection (automaticCon, terminalPool, na)
utmPtermConnectForced	read-only	Like automatic connection but also forces a existing connection to the terminal to be cut (yes, no)
utmPtermConnectMultiplexed	read-only	The terminal uses either a multiplex connection or not (yes, no)
utmPtermConTime	read-only	Duration of existing connection in minutes
utmPtermLett	read-only	Number of messages input and output at the terminal since the start of the application
utmPtermConb	read-only	Number of breakdowns of the physical or virtual connection between this terminal and the application since the start of the application

Table of logical data stations

Object name	Access	Meaning
utmLtermTabNum	read-only	The number of entries in the table of logical data terminals
utmLtermIndex	read-only	A unique value for each entry, its value ranges between 1 and the value of utmLtermTabNum
utmLtermName	read-only	Name of the chosen logical UTM terminal
utmLtermPterm	read-write	Name of the physical terminal that belongs to the logical UTM terminal
utmLtermUser	read-only	Login name of the user currently connected to the logical UTM terminal
utmLtermKset	read-only	Key set of the UTM terminal
utmLtermLock	read-only	Lock of the UTM terminal
utmLtermUsageType	read-only	Type is “D” (dialog terminal) or “O” (output terminal) (dialog, output)
utmLtermUsageBundle	read-only	The UTM terminal is a pool (“bundle”) if “B” is set (yes, no)
utmLtermUsageTermPool	read-only	The UTM terminal is generated for a terminal pool if “P” is set (yes, no)
utmLtermStatus	read-write	The status of the UTM terminal is either “ON” or “OFF” (ON, OFF)
utmLtermOutq	read-only	Number of messages that still have to be output to this terminal
utmLtermInCnt	read-only	Number of messages input at this terminal since the start of the application; for printers, it is the number of print acknowledgments
utmLtermSecCnt	read-only	Number of security violations at this logical terminal since the start of the application

Table of transaction codes

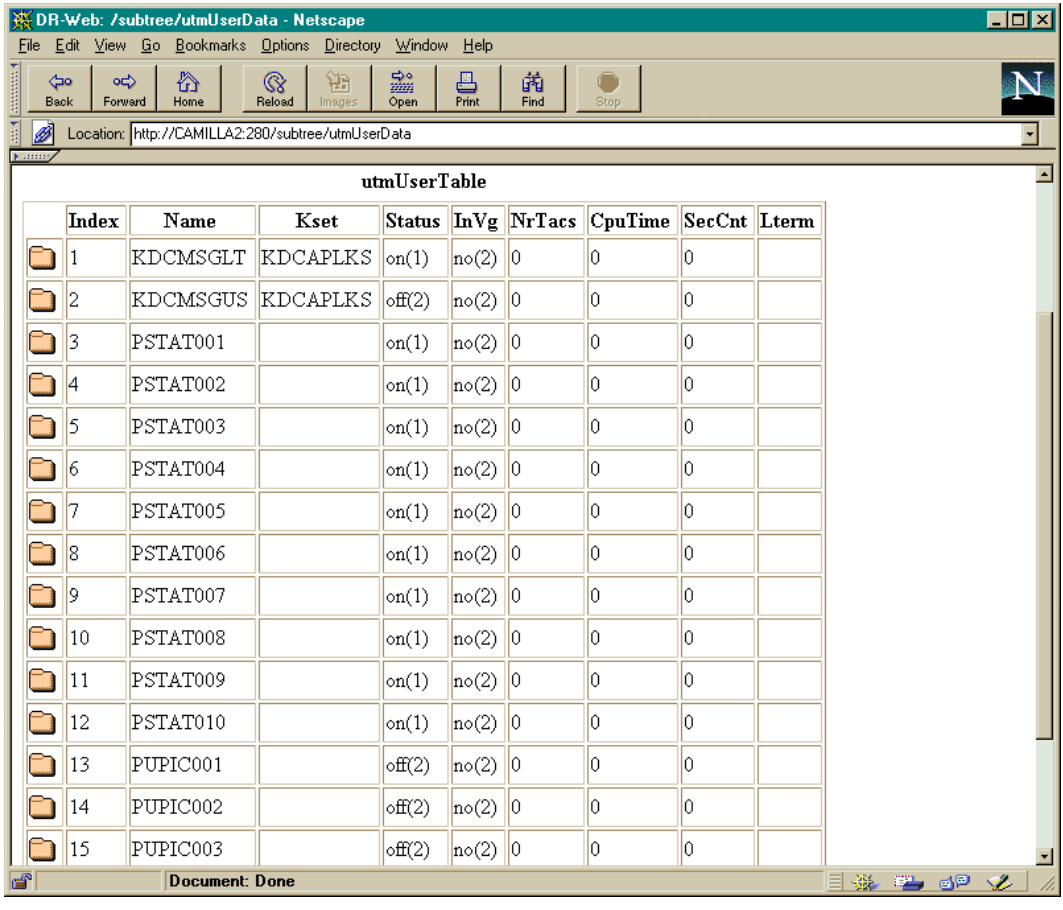
Object name	Access	Meaning
utmTacTabNum	read-only	The number of entries in the table of transaction codes
utmTacIndex	read-only	A unique value for each entry, its value ranges between 1 and the value of utmTacTabNum
utmTacName	read-only	Name of the chosen transaction code
utmTacLock	read-only	Lock of the transaction code (0 through 255)
utmTacStatus	read-write	The status of the transaction code is either “ON”, “OFF” or “HALT” (ON, OFF, HALT)
utmTacTcl	read-only	TAC class of this transaction code
utmTacInq	read-only	Number of messages still to be processed by the program unit run designated by transaction code
utmTacUsed	read-only	Number of program unit runs with this transaction code processed since the initial start of the application (only for asynchronous TACs)
utmTacError	read-only	Number of program unit runs with this transaction code terminated with errors since the initial start of the application
utmTacDbcnt	read-only	Mean number of database calls in the associated subprogram runs. Always “0” if XA interface is used
utmTacElap	read-only	Average runtime in milliseconds of program units with this transaction code
utmTacDbElap	read-only	Average time in milliseconds spent processing the database calls in the program unit runs with this transaction code
utmTacCpu	read-only	Average CPU time in milliseconds spent in the UTM program run for processing this transaction code

Table of TAC classes

Object name	Access	Meaning
utmTacclassNumber	read-only	TAC class number
utmTacclassTasks	read-write	Maximum number of tasks that can currently work for a specific TAC class
utmTacclassWtMesg	read-only	Number of messages for a specific TAC class currently buffered and not yet processed
utmTacclassAvgWtTime	read-only	Average wait times in milliseconds for all dialog TAC classes (1 through 8)
utmTacclassPGWT	read-only	Indicates whether program units containing a PGWT call can run in a specific TAC class (yes, no)

User table

Object name	Access	Meaning
utmUserTabNum	read-only	The number of entries in the table of *open*UTM users
utmUserIndex	read-only	A unique value for each entry, its value ranges between 1 and the value of utmUserTabNum
utmUserName	read-only	Name of the user
utmUserKset	read-only	Key set assigned to users login name
utmUserStatus	read-write	Login name is either locked (OFF) or unlocked (ON) (ON, OFF)
utmUserInVg	read-only	Determines if the user is currently processing a conversation (yes, no)
utmUserNrTacs	read-only	Number of transaction jobs entered by the user since the initial start of the application
utmUserCpuTime	read-only	Number of seconds spent in processing transaction jobs for the user (excluding database calls)
utmUserSecCnt	read-only	Number of security violations for the login name since the start of the application
utmUserLterm	read-only	Name of the logical terminal through witch the user of the UTM application logged on



utmUserTable

Index	Name	Kset	Status	InVg	NrTacs	CpuTime	SecCnt	Lterm
1	KDCMSGLT	KDCAPLKS	on(1)	no(2)	0	0	0	
2	KDCMSGUS	KDCAPLKS	off(2)	no(2)	0	0	0	
3	PSTAT001		on(1)	no(2)	0	0	0	
4	PSTAT002		on(1)	no(2)	0	0	0	
5	PSTAT003		on(1)	no(2)	0	0	0	
6	PSTAT004		on(1)	no(2)	0	0	0	
7	PSTAT005		on(1)	no(2)	0	0	0	
8	PSTAT006		on(1)	no(2)	0	0	0	
9	PSTAT007		on(1)	no(2)	0	0	0	
10	PSTAT008		on(1)	no(2)	0	0	0	
11	PSTAT009		on(1)	no(2)	0	0	0	
12	PSTAT010		on(1)	no(2)	0	0	0	
13	PUPIC001		off(2)	no(2)	0	0	0	
14	PUPIC002		off(2)	no(2)	0	0	0	
15	PUPIC003		off(2)	no(2)	0	0	0	

Figure 43: Table of *open*UTM users

Table of logical connections for distributed processing via LU6.1

Object name	Access	Meaning
utmConTabNum	read-only	The number of entries in the table of connections
utmConIndex	read-only	A unique value for each entry, its value ranges between 1 and the value of utmConTabNum
utmConName	read-only	Name of the connection
utmConProname	read-only	Processor name
utmConLpap	read-only	Name of the remote application in the local application
utmConBcamAppl	read-only	BCAM name of the *open*UTM application
utmConStatus	read-write	A connection to the remote application exists or an be set up (ON), or cannot be set up (OFF) (ON, OFF)
utmConConnected	read-write	A connection is either established (yes) or not (no), or UTM is trying to set up a connection (waiting) (yes, no, waiting)
utmConConnectStatus	read-write	n “A” means automatic connection setup at start of application (automaticCon, noAutomaticCon)
utmConConTime	read-only	Duration of connection in minutes
utmConLett	read-only	Number of messages input and output through the connection
utmConConb	read-only	Number of breakdowns of the connection since the start of the application

Table of remote applications for which the LU6.1 protocol is used for communication

Communication with the remote applications is carried out via the LU6.1 protocol.

Object name	Access	Meaning
utmLpapTabNum	read-only	The number of entries in the table of applications that communicate via LU6.1
utmLpapIndex	read-only	A unique value for each entry, its value ranges between 1 and the value of utmLpapTabNum
utmLpapName	read-only	Name of the LPAP
utmLpapKset	read-only	Key set of the remote application
utmLpapStatus	read-write	A connection to the remote application exists or can be set up (ON), or cannot be set up (OFF) (ON, OFF)
utmLpapQuiet	read-write	A “Q” means “quiet”, i.e. no more dialog jobs for the remote application are accepted (yes, no)
utmLpapOutq	read-only	Number of messages that still have to be sent to this remote application
utmLpapIdleTime	read-write	Time for monitoring the idle state of a session in seconds

Table of terminal pools

Object name	Access	Meaning
utmPoolTabNum	read-only	The number of entries in the table of terminal pools
utmPoolIndex	read-only	A unique value for each entry, its value ranges between 1 and the value of utmPoolTabNum
utmPoolProname	read-only	Processor name of the terminal pool
utmPoolBcamAppl	read-only	Name of the access point to the transport system which was generated for this terminal pool
utmPoolPtype	read-only	Physical terminal type of the terminals that can be connected to the application via this pool
utmPoolStations	read-only	Generated maximum number of terminals that can be connected to the application via this pool
utmPoolStatusOn	read-write	Maximum number of terminals with STATUS=ON
utmPoolActCon	read-only	Number of terminals connected to the application via this pool
utmPoolMaxCon	read-only	Maximum number of terminals that have been simultaneously connected to the application via this terminal pool
utmPoolKset	read-only	Key set of the terminals of this terminal pool
utmPoolLock	read-only	Lock of the terminals of this terminal pool. Number between 1 and the maximum number (255) permitted in the application Default: 0 (no lock)

Information about local sessions (only for VTV via the LU6.1 protocol)

Object name	Access	Meaning
utmLsesTabNum	read-only	The number of entries in the table of LU6.1 sessions
utmLsesIndex	read-only	A unique value for each entry, its value ranges between 1 and the value of utmLsesTabNum
utmLsesName	read-only	Name of the session in the local application
utmLsesRses	read-only	Name of the session in the remote application
utmLsesLpap	read-only	Name of the remote application for which the session is generated
utmLsesCon	read-only	Denotes the transport connection that is set up for the session
utmLsesProname	read-only	Processor name
utmLsesBcamAppl	read-only	Denotes the transport connection that is set up for the session
utmLsesAgUser	read-only	Name of the job-submitting partner for which the session has been reserved

Table of transaction codes for remote applications

Object name	Access	Meaning
utmLtacTabNum	read-only	The number of entries in the table of LTACs
utmLtacIndex	read-only	A unique value for each entry, its value ranges between 1 and the value of utmLtacTabNum
utmLtacName	read-only	Name of the LTAC
utmLtacLock	read-only	Lock of the remote conversation
utmLtacStatus	read-write	The LTAC transaction code is either locked (OFF) or unlocked (ON) (ON, OFF)
utmLtacRtac	read-only	Name of the transaction code in a remote application
utmLtacLpap	read-only	Name of the remote application in the local application
utmLtacAccessWait	read-write	Time in seconds spent waiting for a session or association to be reserved
utmLtacReplyWait	read-write	Time in seconds spent waiting for response from receiving partner
utmLtacUsed	read-only	Number of jobs issued to this LTAC since the start of the application

Table of OSI-TP associations

Object name	Access	Meaning
utmOsiAssTabNum	read-only	The number of entries in the table of OSI-TP associations
utmOsiAssIndex	read-only	A unique value for each entry, its value ranges between 1 and the value of utmOsiAssTabNum
utmOsiAssName	read-only	Name of the OSI association
utmOsiAssOsiLpap	read-only	Name of the remote application in the local application for which the association is generated
utmOsiAssOsiCon	read-only	Name of the connection set up to the remote application for the association
utmOsiAssAgUser	read-only	Name of the job submitter for which the association is reserved. As of *open*UTM V4.0, no valid entry is supplied for this object, but *not supported* is output.
utmOsiAssConTime	read-only	Duration of the connection in minutes
utmOsiAssLetters	read-only	Number of messages input and output since the start of the application. As of *open*UTM V4.0, no valid entry is supplied for this object, but *not supported* is output.

Information about the logical connections for distributed processing via the OSI-TP protocol

Object name	Access	Meaning
utmOsiConTabNum	read-only	The number of entries in the table of OSI connections
utmOsiConIndex	read-only	A unique value for each entry, its value ranges between 1 and the value of utmOsiConTabNum
utmOsiConName	read-only	Name of the OSI connection
utmOsiConOsiLpap	read-only	Name of the remote application in the local application
utmOsiConTsel	read-only	BCAM application name of the remote OSI-TP partner (transport selector)
utmOsiConNsel	read-only	Name of the processor on which the OSI-TP partner is located (network selector)
utmOsiConAccPnt	read-only	Local name of an access point through which communication with OSI-TP partners takes place
utmOsiConActive	read-write	The transport connection can either be used (YES) or is reserved as a substitute connection (NO) (yes, no)

Information about remote partner applications

Object name	Access	Meaning
utmOsiLpapTabNum	read-only	The number of entries in the table of the OSI-LPAPs
utmOsiLpapIndex	read-only	A unique value for each entry, its value ranges between 1 and the value of utmOsiLpapTabNum
utmOsiLpapName	read-only	Name of the OSI LPAP
utmOsiLpapKset	read-only	Key set of the remote application
utmOsiLpapStatus	read-write	A connection to the remote application exists or can be set up (ON), or cannot be set up (OFF) (ON, OFF)
utmOsiLpapQuiet	read-write	A “Q” means “quiet”, i.e. no more dialog jobs for the remote application are accepted (yes, no)
utmOsiLpapOutq	read-only	Number of messages that still have to be sent to this remote application
utmOsiLpapIdleTime	read-write	Time for monitoring the idle state of a session
utmOsiLpapOsiCon	read-only	Name of the transport connection used to communicate with the OSI-TP partner
utmOsiLpapAssoc	read-only	Number of parallel connections generated for the OSI-TP partner
utmOsiLpapConnect	read-only	Number of connections set up
utmOsiLpapAutoCon	read-only	Number of connections to be set up to the partner when the application is started

9 Operating the management station

The integration packages SMBS2 (for Reliant UNIX and Windows NT) and SMAWsmbs2 (for Solaris) contain the following additional components for integrating system management for BS2000/OSD into the following management platforms:

– Unicenter TNG
– TransView SNMP
– OpenView NNM (Network Node Manager)

The integration packages are described in section “User interface for the SNMP management of BS2000/OSD” (see page 18).

9.1 Integration in the user interface

The installation of SMBS2 or SMAWsmbs2 requires one of the above-mentioned management platforms.

9.1.1 Integration in the user interface of Unicenter TNG

BS2000/OSD is integrated as a separate object class in the World View repository of Unicenter TNG. The BS2000/OSD objects can be managed with the same functions as all other objects in the repository, i.e. icons for BS2000/OSD systems can be added to the 2D and 3D display of the network map (see pages 92 and 100). figure 44 on the following page shows the 2D representation of a network map with BS2000/OSD systems.

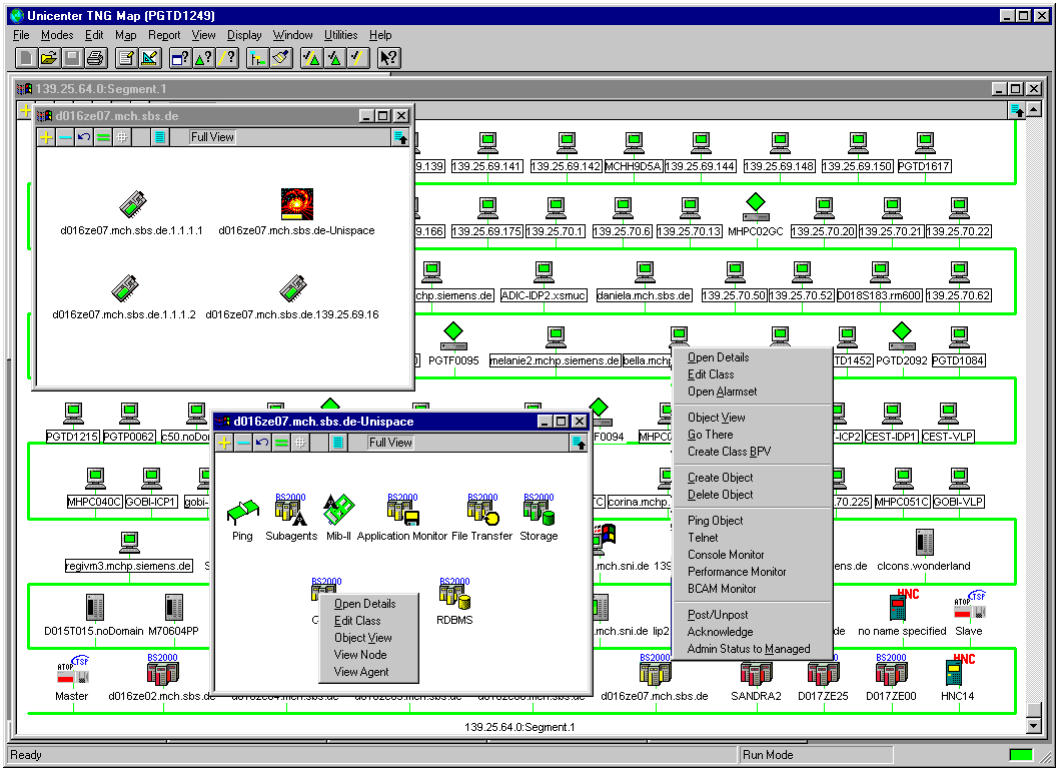


Figure 44: Representation of the monitored BS2000/OSD system in a network map

Each icon in the network map is linked to a pop-up menu that opens when you click the right mouse button.

The menu of the “Siemens_BS2000” class is divided into five subgroups:

– Open Details
– Edit Class
– Open Alarmset

– Object View
– Go There
– Create Class BPV

– Create Object
– Delete Object

– Ping Object
– Telnet
– Console Monitor
– Performance Monitor
– BCAM Monitor

– Post/Unpost
– Acknowledge
– Admin Status to Managed

The commands “Console Monitor”, “Performance Monitor”, and “BCAM Monitor” are used to call the management applications Console Monitor, Performance Monitor and BCAM-Monitor respectively. When these applications are called, certain presettings – in particular the system name – are taken from the network map data.

The remaining commands correspond to the commands in the standard menu for objects of the “Host” class. The functions of these commands are described in the Unicenter TNG documentation.

A double-click on the BS2000/OSD icon opens a further subnet map. As of Version 2.2 of Unicenter TNG, this window can contain icons for the individual interfaces. If you are implementing a full version of Unicenter TNG with agent technology, a Unispace icon is entered in the network map.

A double-click on the Unispace icon opens a further subnet map that contains icons for all BS2000/OSD agent classes found. These may be icons for the classes “Ping“, “Mib2“, “Supervisor“, “Application Monitor”, “AVAS”, “HSMS”, “Storage”, “RDBMS”, and “OMNIS”.

The standard pop-up menu for agent classes is linked to the icons with the following entries:

– Open Details
– Edit Class
– Object View
– View Node
– View Agent

This menu corresponds to the standard menu for objects of the “Agent” class. The functions of the menu commands are described in the Unicenter TNG manuals.

9.1.1.1 NodeView display

Choose the “View Node” command to open a “Node View” window displaying the status of the MIB-II interface and of the individual subagents (see figure 45 on the next page).

The following ten DSM objects can be created for each monitored BS2000/OSD system:

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

Status changes are triggered by polls and by the receipt of traps.

The NodeView contains the following displays for BS2000/OSD systems:

- Application Monitor:

  Status display of all monitored subsystems and of all BCAM, user, and DCAM applications.

- AVAS:

  Overall status of AVAS

- HSMS:

  Displays availability of the HSMS subagent

- Omnis:

  – Displays the status of all monitored OMNIS systems
  – Displays the receipt of important traps relating to an OMNIS system

- RDBMS:

  Displays the availability of the database server for the databases

- Storage:

  Displays the saturation levels of pubsets and the availability of private disks

- Supervisor:

  Status display for the subagents

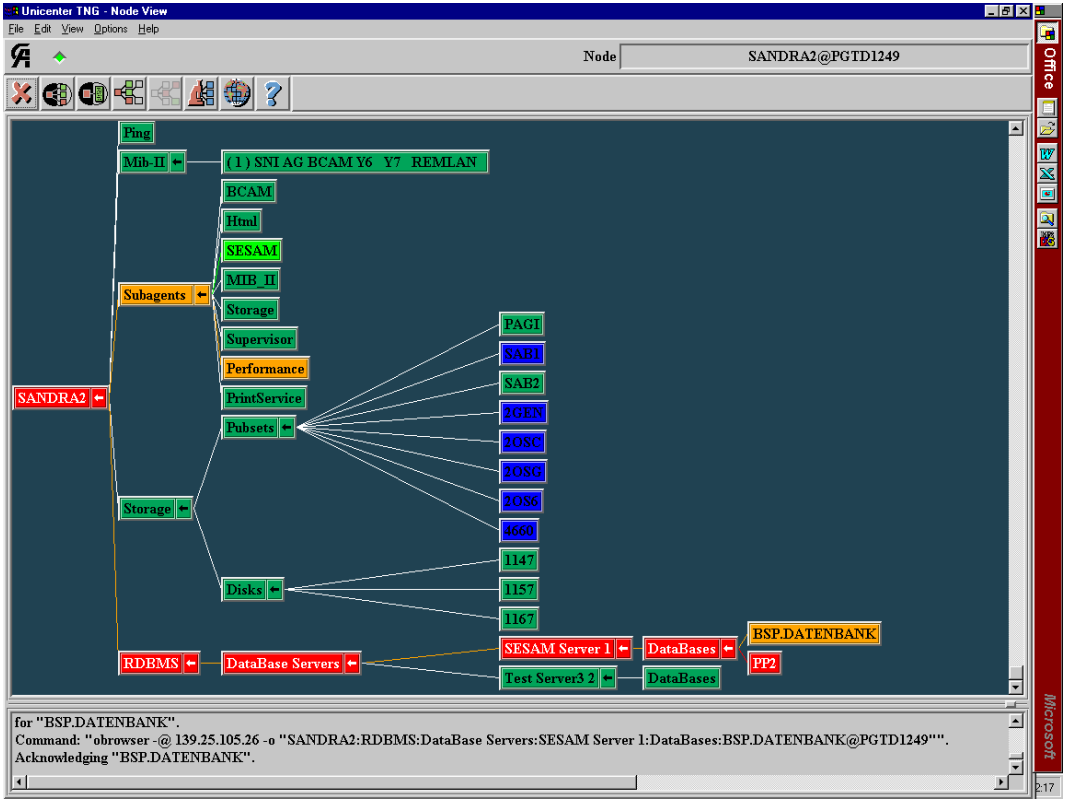


Figure 45: Node view

9.1.2 Integration in the user interface of TransView SNMP

The management platform TransView SNMP supports user-friendly monitoring of BS2000/OSD computers and applications in a heterogeneous IT landscape.

9.1.2.1 Monitoring the BS2000/OSD computer

A double-click on the BS2000/OSD computer icon opens the object view for this system. The functionality of the SNMP basic agent and the host resources subagents is hidden behind this icon.

The window working area includes the following displays (see figure 46 on the following page):

– system name
– system Internet address
– community string
– characteristics group

Part of the information from the MIB-II system group is displayed in the *sysDescr*, *sysObjectID* and *sysUpTime* fields. The *sysUpTime* describes the time that the master agent has been operational in this system. The poll cycle shows the time after which the information in the window is updated by a renewed request. A prerequisite for this is that the polling was enabled by activation of the *poll* action button. The menus and window elements correspond to all other device views under TransView SNMP.

The *objects* menu is specific:

– The menu entries *MIB II* and *MIB II Sx* output the MIB-II values that are supplied by the MIB-II subagents (see figure 46).

– The menu entries *RFC1514-HOST-RSC*, *BS2000-APPMON*, *Console Monitoring* and *Subagent Monitoring* show the information about the values of the associated MIBs that are supplied by the relevant subagents.

9.1.2.2 Monitoring of BS2000/OSD components

Part of the information from the MIB-II system group is displayed in the *sysDescr*, *sysObjectID* and *sysUpTime* fields. This information is identical to that of the BS2000/OSD system. The menus and window elements correspond to all other device views under TransView SNMP.

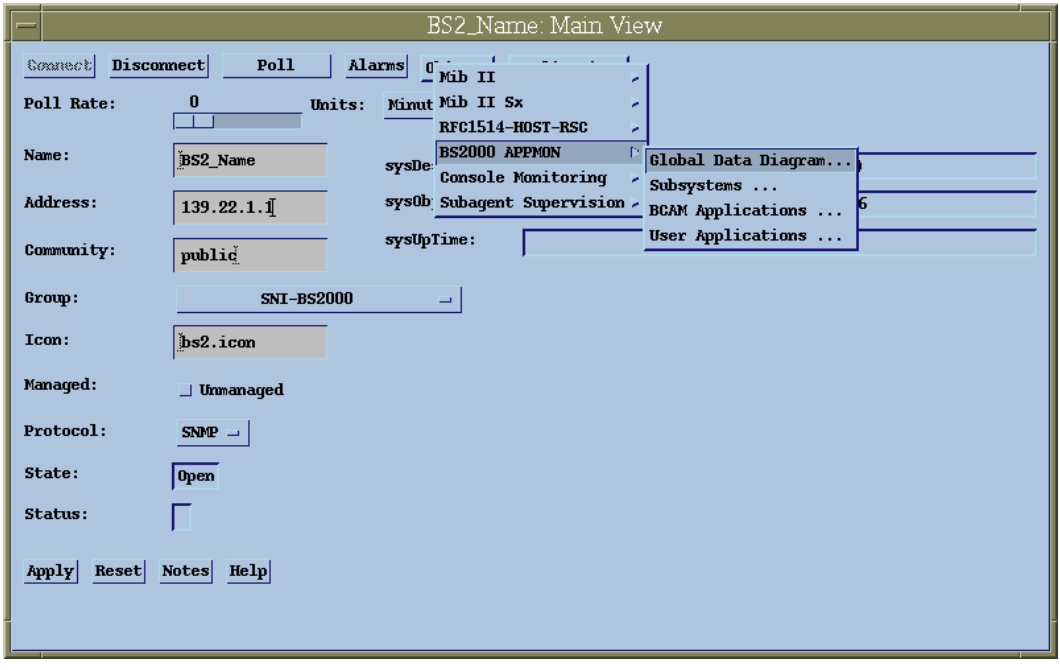


Figure 46: BS2000/OSD System overview in TransView SNMP

The *objects* menu is specific. This menu contains an entry with the name of the standardized or private MIBs for this component. This entry allows access to the values of the MIB objects that are supplied or changed by the corresponding subagent.

Displaying object values

The *objects* menu provides you with an overview with either a table window or a simple listing of the objects with their values, dependent on whether the object group allows more than one instance.

Table window

The work area of the window forms a table, where the columns stand for the attributes of the object types depicted in the window and the lines describe the object entities. The attribute values are in a single field. The object entities are addressed uniquely via the indices indicated in the first column of the table.

Object window

The object window is used for depicting the attribute values of object types that can only have one instance. The information is shown as a pair, made up of the attribute name and value.

The menus and window elements correspond to all other windows of this type under TransView SNMP.

In particular, the poll cycle also shows the time after which the information in the window is updated by a renewed request. A single poll is triggered when the window is opened. Activating the *Poll* button without setting a poll cycle also triggers a single poll cycle that results in a display update.

Individual columns in the tables for the objects of the SESAM subagent remain blank because the SESAM subagent does not fully support the RDBMS-MIB. However, the tables always contain columns for all the objects defined in the RDBMS-MIB. This allows for the utilization of the same formats when supporting other database systems.

Setting the object values

Some subagents allow object values to be set. You can change attribute values with the *set objects* application. The attribute must be defined as write and the Community String in the overview window must possess write authorization to the agent system.

1. Select *applications* in the overview display or in a table or form window.
2. Select *set object* from the displayed list.
3. Select the basic object and then the attribute in the dialog box.

Figure 47: Setting the object values

4. Write the desired value into the *value* field.
5. Click on the *add* push button.
6. If necessary, enter an instance for which the value is to be set.
7. Click on the *set object* push button.

TransView SNMP displays a message showing whether the value could be set. A description for this can be found in the relevant manual.

9.1.2.3 Alarms

A range of alarms are defined for the automatic monitoring of BS2000/OSD systems.

RDBMS_relState

This alarm is used to display the access options for the database server to the databases. The states of the alarms correspond to the values of the *rdbmsRelState* object. If access is lost or restricted, the SESAM subagent sends a trap. No trap is sent in the opposite direction to indicate a return to the normal state. The complete control of the alarms based only on traps for displaying the database availability is therefore not possible. Polls must be used. There is another reason, however: the alarm does not relate to the entire application monitored by the SESAM subagent, but only to one of several similar object entities, namely the relation of a server to a database. Alarms for object entities can only be controlled by polls in TransView SNMP. For the alarm, this means that the SESAM subagent trap can only be used to activate the four polls required. The polls are also activated after installation of SMBS2. All four polls must be activated for the alarm to be fully operable. The polls access the values *rdbmsRelState* objects every three minutes. This is the same interval at which the SESAM subagent updates these indicators. It can therefore happen that the display of the change in the access mode to a database is delayed. You can reduce this interval by dropping the poll frequency.

AVASState

The alarm indicates receipt of individual traps from the AVAS MIB. It has the state *Normal*, *Missing*, *Ready*, *Running*, *ErrorSystem*, *ErrorNet* and *ErrorSignon*. The first four states have the weight *normal*. The other three indicate problems with the weights *Serious* (or *Major*), *Light* (or *Minor*) or *Information*. The alarm is controlled exclusively by traps. The states *Normal* and *Missing* can only be assumed at the start. During later operation, the states can only change between the other values. The alarm states correspond to the meaning of the traps displayed.

SupervisBasic, SubagentStatus, SubagentStatus_a

These alarms generate a display for the states of the subagents. Whereas the status of the MIB-II, Application Monitor, Console Monitor and Host Resources subagent is indicated by a change in color of the BS2000/OSD icon, changes of state at the subagents cause a change in color of the line icon between BS2000/OSD icons and the associated application icons. The *active* state is displayed for the *Normal* state. This state applies as long as the subagent is unknown to the master and thus to the management station. Prior to the first logon, there is no entry in the subagent table of the supervisor MIB. The object entities for the alarm do not yet exist. Logging off a subagent causes the color to change to blue, i.e. to a state that has the weight *Information*. If a subagent does not appear to answer and is therefore set to the state *undefined*, the state changes to one that is weighted *Light* (or

Minor). A change of state can take place from any state to one of the other two states.The alarms are controlled by the traps of the supervisor MIB and by polls. After receipt of a trap, the polls are deactivated briefly.

OmnisMsg

This alarm indicates that OMNIS traps have been received. If an OMNIS trap is received, the *Normal* state is exited and one of the states *Inform* or *Minor* assumed, which are assigned the weights *Information* and *Light* (or *Minor*). The alarm only has the function of indicating trap receipt to the user. The weights correspond to the meaning of the traps. The alarm states cannot be exited automatically. The alarm must be reset manually.

9.1.3 Integration in TransView Control Center

Integration in TransView Control Center allows you to monitor Symmetrix devices. The alarms described below are defined for this purpose.

The following four alarms are used to display Symmetrix events. The state changes are only controlled by the TransView Control Center.

sym-sp

This alarm concerns problems with the Symmetrix service processor notified with the *NJD0010* message. The alarm diagram consists of two states: *Normal* (weight: *Normal*) and *SP-Down* (weight: *light*). The reference codes *x476* and *x477*, which indicate that there is no connection to the EMC Service Support Center (EMC CSC) via the service processor, cause a transition to the *SP-Down* state. A message with reference code *x47F* signals a successful autocall to the EMC Service Support Center. After this message, you can return to the *Normal* state. This event type can only be registered if Symmetrix has been configured appropriately. If the event type is not registered, this alarm must be reset manually.

sym-partner

The availability of the SRDF connections (SRDF=Symmetrix Remote Data Facility) to the partner devices is also indicated by the states of the alarms:

all-connected (weight: *Normal*) means that all SRDF connections are working.

part.-connec. (weight: *Warning*) means that the SRDF connections are only partially available. Individual connections have failed.

all-disconn. (weight: *Light*) means that all SRDF connections have failed and that there is no contact to the partner devices.

sym-disk

The diagram of this alarm has two states: *Normal* and *Disk_Problem* (weight: *Serious*) and only one change of state from *Normal* to *Disk_Problem*. If affects all reference codes that signal a disk problem. The alarm must be reset manually.

sym-error

This alarm concerns problems that are displayed with the message *NJD0011* and make up the majority of hardware problems in the Symmetrix system (e.g. overheating or problems with the power supply). As for the alarm that displays disk problems, there are only two states for this alarm: *Normal* and *Error* (weight: *Serious*). The change of state from *Normal* to *Error* occurs after a problem has been notified. In the *Normal* state, the alarm can only be reset manually.

9.1.4 Integration in the user interface of HP OpenView

After installation of SMBS2, the following extensions are available at the OpenView interface:

– The MIB structure is known in the MIB browser.

– The icons for BS2000/OSD systems can be added to the network map.

– The menu for the BS2000/OSD application has been added to the menu bar of the network representation. The menu can open windows to MIB objects.

The BS2000/OSD menu is called as follows:

1. A single click with the left mouse button marks the BS2000 icons.
2. Then you can open the menu for the BS2000/OSD application. This menu is available both in the menu bar and in the po-up window in the icon.

The structure of the menus and windows added by SMBS2 corresponds to the menus and windows of TransView-SNMP extended by the SMBS2 package in the main. The BS2000/OSD menu reflects the structure of the BS2000/OSD MIBs. The entries in the first level correspond to the areas represented by the various subagent icons in the subnet maps for BS2000/OSD systems in TransView-SNMP. The remaining structure of the submenus and the window design is similar to the menus of the BS2000/OSD interface for TransView-SNMP as far as possible.

The submenus of the BS2000/OSD menu include the windows for objects from the following MIBs.

System: Application Monitoring MIB, Console Monitoring MIB, Supervisor MIB

AVAS: AVAS MIB

File Transfer: File Transfer MIB

Host Resources: Host Resources MIB..

HIPLEX-AF: HIPLEX-AF MIB

OMNIS: OMNIS MIB

Performance: SNIPERF MIB

Printservice: Spool MIB

SESAM: RDBMS MIB

Storage: Storage Management MIB

UTM: UTM MIB

The *Performance* submenu has a special item *graphics*. This submenu allows you to call the graphic monitoring of performance values, such as CPU times and the I/O frequency. Regular polling of the current values produces a curve which changes in real-time.

The SESAM subagent does not support the full scope of the RDBMS-MIB. For this reason, only those columns are defined in the tables for part of the MIB objects which are supported by the subagent.

Status events are defined for the OMNIS, Supervisor and RDBMS traps. Receipt of these traps is indicated by a message.

Following installation, the BCAM-MIB is also loaded. Forms and tables for this MIB are not supported by SMBS2 for OpenView NNM.

General information about the operating status of a monitored BS2000/OSD system are available in the context of the *All Event Browsers*. If a system failure is notified, the icon frame turns red; under normal operating conditions, it is green.

9.2 Management applications CMBS2 and PMBS2

Separate management applications, which are customized to the special properties and tasks of the subagents concerned, are available for special subagents such as the Console Monitor subagent and the Performance Monitor subagent:

– CMBS2 for the Console Monitor subagent
– PMBS2 for the Performance Monitor subagent

These management applications can be integrated into the management platforms described above. The supplement and enhance the display and handling of the existing management platform and offers a network-wide overview of all BS2000/OSD systems.

9.2.1 CMBS2 application for the Console Monitor subagent

CMBS2 supports the Console Monitor subagent in recording, processing and filtering console messages and offers a remote access to the consoles of all BS2000/OSD systems in the network.

Functionality

- displaying console messages with the option to set filters both at the agent and at the management station,
- setting filters on the agent system to reduce network load and provide a more legible display
- replying to queries which appear on the console
- issuing commands
- automatic reactions to traps.

Starting and stopping CMBS2

CMBS2 is started from the shell level on UNIX systems by calling the *ConsMon* procedure. The application main window then appears.

On WindowsNT, the application is started from the *SNMP Management Applications* program group. Alternatively, you can call it by double-clicking on the *Console Monitor* in the *<tcldir>\appl\Cmbs2*.

CMBS2 can be stopped from any window by selecting *Terminate ConsMon* from the *ConsMon* menu. Unintentional termination is prevented by means of a security query. Parameter changes are either saved, canceled or queried, depending on the save mode.

9.2.1.1 Setting the user interface

CMBS2 has various display modes for concentrating the attention of the user on the most important objects of his current activity.

Trap window or command window

You can switch between a trap window display and that of a command window in the CMBS2 main window via the menu or toolbar. Refer to pages 350 or 358 for a description of the windows.

Displaying the SNMP parameters / hiding the community

The can hide the community display via the menu and the toolbar.

SNMP parameters

In order to sent a message to a system correctly via SNMP, it is necessary to specify the IP address of the target system, a port and a community string. The community string defines the access rights and the authorization scope. The Console Monitor generates a list of possible target systems with their names and SNMP parameters when it is started. You can edit this list of systems can under the menu bar
Options → *Settings* → *Systems*.

The IP address in specified in the *Address* input field. If the system name is modified in the SNMP parameter range, the IP address is read from the system list. If the system is not included in the system list, the IP address remains unchanged.

The *Port* input field defines the port. If the system name is modified in the SNMP parameter range, the port is read from the system list. If the system is not included in the system list, the port remains unchanged.

The *Community* input field contains data relating to the community.

The community defines the access authorization and its extent for SNMP requests. If the system name changes, the community is adapted according to the same rules as for ports.

Displaying the filters in the trap window

The trap window can be made more legible by deleting the local filters from the display.

9.2.1.2 Trap window

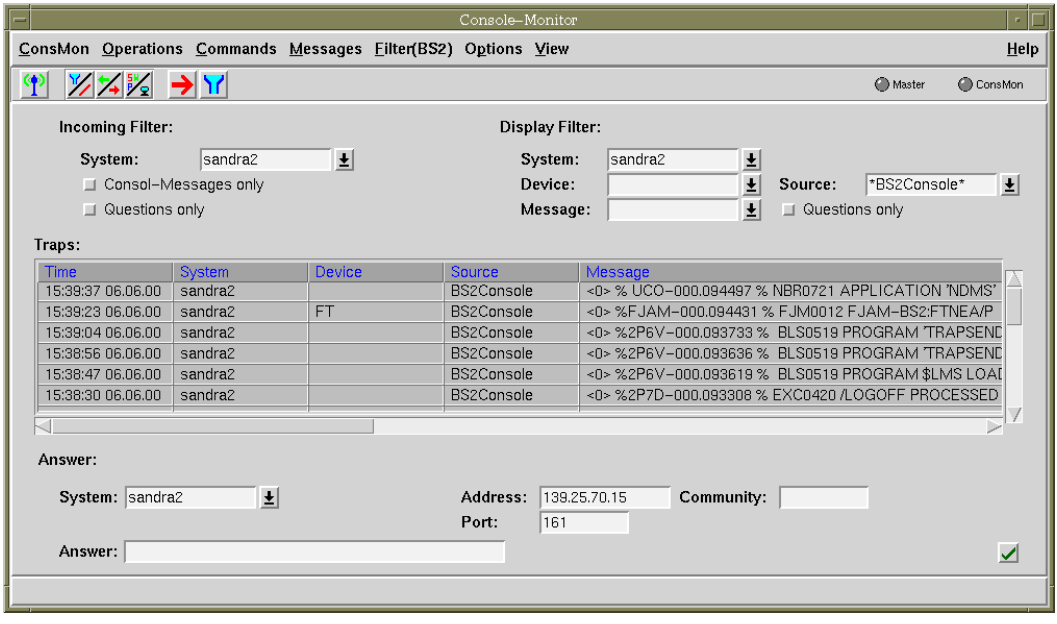


Figure 48: Trap window

The trap window is also the CMBS2 main window, from which you can open other windows for further actions. Closing this window terminates the application. You can change into command window mode via the menu or toolbar in this window (see page 360).

The window is divided into the following four areas:

1. menu bar
2. toolbar
3. work area
4. help line

Trap window menu bar

The menu bar contains the *ConsMon*, *Operations*, *Commands*, *Messages*, *Filters(BS2)*, *Options*, *View* and *Help* menus.

The *ConsMon* menu contains the item *Terminate ConsMon*.

The *Operations* menu contains the “Test” function, which is used to check the following:

– accessibility of the specified system via ICMP

– status of the SNMP master agent and of the Console Monitor subagent with the following result display:

  “Master” displayed in the toolbar:

  – red: no connection via ICMP
  – yellow: *noResponse* via SNMP
  – blue: community read-only
  – green: community read-write

  “ConsMon” displayed in the toolbar:

  – yellow: *noResponse* via SNMP
  – green: the Console Monitor subagent has a connection to the master agent

In secondary windows, the menu also contains the *Close* command for closing the window.

The *Commands* menu contains the item *New window* for opening a separate window to input BS2000 commands. All further items are for use in command mode and are deactivated in the trap window.

The menu entries

– *Delete messages* → *all*,
– *Delete messages* → *displayed* and
– *Delete messages* → *selected*

can be used to delete all messages displayed or selected in the trap output area. A message is selected by either double-clicking on it or moving the mouse over the message text.

In *Filter(BS2)*, the menu entry *Edit* opens a further window via which you can display and modify the message filter for the Console Monitor in BS2000.

Options contains the entries *Settings*, *Reactions* and *Save*:

– *Settings* opens an options dialog box, in which you can edit and save the settings for the areas *Systems*, *Groups*, *Ping*, *SNMP*, *Messages*, *Commands* and *Save*.

– *Reactions* in which you can define the reactions to incoming messages.

– *Save* → *Settings + Reactions* saves all values of the setting areas in the configuration file immediately.

– *Save* → *Start Configuration* saves all the values of the setting areas and the start configuration in the configuration file immediately. After a restart of the Console Monitor, all the settings and values for the trap window are in the state that applied when the menu entry was activated.

  Note: the menu entry *Save* is only active when all dialog boxes are closed.

The *View* menu contains the following entries:

– *Display Filter* allows you to show and hide the filter.

– *Commands* switches between the trap window and the command window.

– *Display Community* allows you to show and hide the community string.

Help provides information about the version, to the new functions in Version V5.0, the active window and the index.

Trap window toolbar

The toolbar contains a selection of the functions available in the menu. It provides fast access to frequently used operations. A brief description of the functionality behind the icon is displayed if the mouse cursor remains on the icon for more than one second.

The toolbar comprises the following three groups:

The first group contains the *Test* function, which can be used to check the accessibility of the specified system via ICMP as well as the status of the SNMP master agent and of the Console Monitor subagent.

The results are displayed as follows:

“Master” displayed in the toolbar:

– red: no connection via ICMP
– yellow: *noResponse* via SNMP
– blue: community read-only
– green: community read-write

“ConsMon” displayed in the toolbar:

– yellow: *noResponse* via SNMP
– green: the Console Monitor subagent has a connection to the master agent

The second group allows you to:

– hide the filter in the trap window
– switch the current work area between the work area of a trap window and the work area of a command window
– hide the community string

The third group contains the following functions for opening

– another command window and
– a window for displaying and setting filters

Work area of a trap window

The work area of a trap window is divided into the following three parts:

1. area for setting the local filter (input filter and display filter)
2. display area for incoming traps
3. area for replying to a console question

Setting a local filter

The local filters are split into incoming and display filters.

Incoming filter

The display of incoming traps can be restricted by the setting of the incoming filter. Previous traps are not removed from the display. Filtered messages are rejected. The filter is formed with a logical AND of the following three parameters.

System Combobox	Specifying a system or a system group allows the display to be limited to traps coming from a specific system or group of systems. Groups are represented by [groupname]. The square brackets need not be specified when entering group names. To edit groups, use *Options* → *Settings* → *Groups*. The corresponding configuration is displayed in the help line by double-clicking on the group name. If the field is left blank, the display is not limited. The Combobox list is preset by the system list and the system groups that have already been defined.
Checkbutton *only console messages*	Activating this button limits the display to Console Monitor traps.
Checkbutton *only queries*	Activating this button limits the display to queries from the console.

Display filter

Setting the display filter allows restriction of the display of all non-rejected traps. Filtering is always applied to the complete set of incoming traps, allowing return to the complete set with "*" or "". The filter is formed with a logical AND of the following four parameters.

System Combobox	Specifying a system or a system group allows the display to be limited to traps coming from a specific system or group of systems. Groups are represented by [groupname]. The square brackets need not be entered when specifying the group name. Groups are edited in the menu bar via *Options* → *Settings* → *Groups*. The configuration of a group is displayed in the help line by double-clicking on the group name. An empty field does not limit the display. The Combobox list is preset by the system list and the system groups that have already been defined.
Source Combobox	The source filter can be entered directly in the text field. The display is limited to the sources concerned. The Combobox list is preset with the filters defined in *Edit source filters* (see menu bar *Options* → *Settings* → *Messages*).
Object Combobox	The object filter can be entered directly in the text field. The display is limited to the sources concerned. The Combobox list is preset with the filters defined in *Edit object filter* (see menu bar *Options* → *Settings* → *Messages*).
Message Combobox	The message filter can be entered directly in the text field. This filter is evaluated by comparing the string with the net message according to the “global style rules”. This also allows limiting to the weight of a message. The Combobox list is preset with the entries defined in *Edit message filter* (see menu bar *Options* → *Settings* → *Messages*).
Checkbutton *only queries*	Activating this button limits the display to queries from the console.

Display area of incoming traps

The incoming traps are display in tabular form in this area. The last incoming trap is displayed in the first line.The oldest traps are deleted if the number exceeds the number defined in *Options → Settings → Messages*.

Output	**Meaning**
Time	Time and date of trap reception
System	Trap sender. If the system is entered in the system list, the name belonging to the address is displayed, otherwise the IP address.
Object	The text of the net message between the code words *$DEVC$:* and *MSG:* is evaluated for this field. This is taken from the DEVICE code of the BS2000 message filter file.
Source	Note as to whether the trap came from a BS2000/OSD console monitor subagent. This is normally *BS2Console*, unless a different source was negotiated as described on page 72. The text evaluated for this field is the net message between the keywords *$SOURCE$:* and *$DEVC$:*.
Message	Net message of the traps

The message can be marked by double-clicking on a message line or by moving the mouse cursor over an area of the message. A marked message can be deleted via the *Traps → Delete messages → Selected*. Double-clicking on the message line of a query transfers the entry of the sender into the *System* field and the TSN into the input field of the *Reply* area.

A double-click on the middle mouse button activates the trap confirmation function (see section “Trap security” on page 203); a further double-click deactivates it again. Red table headings indicate that the trap confirmation function is active.

Area for replying to a console query

You can enter a reply to a query from the console in this area.

The target system to which the reply is to be sent is defined in the *System* Combobox. The Combobox list is preset with the entries in the system list. The name of the system can be taken over from the list by the Combobox mechanism or input directly. It can also be taken over from the trap output area by double-clicking on a message line.

The SNMP parameters are displayed in the second part.

The reply text is input in the *Reply* input field. The required leader for the TSN can also be taken from the trap output area by double-clicking on the message line.

Action buttons:

The reply is sent to the specified system when this button is activated. The result of the SNMP operation is displayed between the input field and the button. A console message is displayed in the trap output window if the reply cannot be correctly processed by the BS2000 system.

Trap window help line

A brief description of the window element function is displayed in the help line if the mouse cursor remains on the element for more than one second.

If you double-click the group selected under *Input or output filter*, the systems belonging to this group are output in the help line.

9.2.1.3 Command window

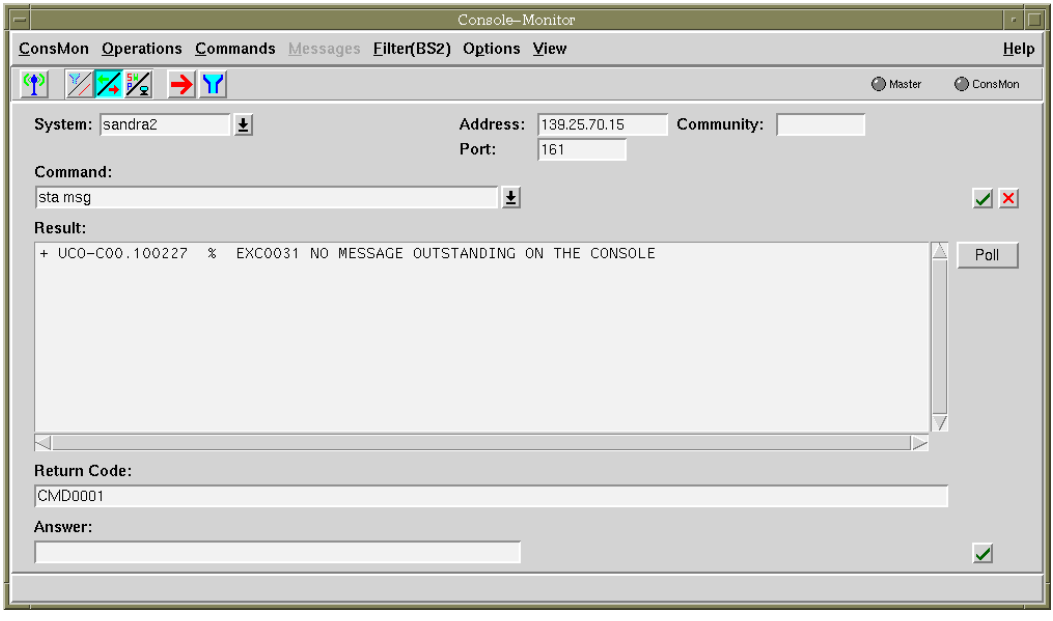


Figure 49: Command window

The command window can be either a main or subsidiary window. If it is a main window, further command windows can be opened from it. If it is closed as a main window, the application is terminated. If CMBS2 is started in trap mode and the command window is a main window, you can change into the trap window mode via the menu or toolbar (see page 351).

As with the trap window, the command window is also divided into four areas:

1. menu bar
2. toolbar
3. work area
4. help line

Command window menu bar

The menu bar contains the menus *ConsMon*, *Operations*, *Commands*, *Filter(BS2)*, *Options*, *View* and *Help*.

The *ConsMon* menu contains the *Close* entry.

The *Operations* menu contains the *Test* function, which is used to check the accessibility of the specified system via ICMP, as well as the status of the SNMP master agent and of the Console Monitor subagent.

Please refer to page 351 for the relevant descriptions.

The *Commands* menu contains the following entries:

– *New window* item opens a separate window to input BS2000 commands. This window is only available if the command window is the main window.

– The *Execute* menu sends the command specified in the *Command* input field to the Console Monitor for execution. Command output is in the result area. The SNMP message is output behind the input field if an error occurs.

– The process which sends the command to the subagent and fetches the command results is terminated with *Stop*. Depending on the state, it is not possible to be certain that the command is not executed. A command that has already been sent by the Console Monitor is never aborted!

– The *Command* input field list, which is reachable via the Combobox, is sorted in the *Sort list* menu.

Via the *Edit* item in the *Filter(BS2)* menu, a further window is opened in which the message filter for the Console Monitor in BS2000 can be displayed and edited.

Options contains the entries *Settings*, *Reactions* and *Save*:

– *Settings* opens an options dialog box, in which you edit and save the settings for the areas *Systems*, *Groups*, *Ping*, *SNMP*, *Messages*, *Commands* and *Save*.

– *Reactions* in which you define the reactions to incoming messages.

– *Save* → *Settings* + *Reactions* saves all values of the setting areas in the configuration file immediately.

– *Save* → *Start Configuration* saves all the values of the setting areas and the start configuration in the configuration file immediately.

The *View* menu contains the following entries:

– *Commands* switches between the display of a trap window and a command window.

– The *Display Community* function allows you to show and hide the community string.

Help provides information about the version, to the new functions in Version V5.0, the active window and the index.

Command window toolbar

The toolbar contains a selection of the functions offered in the menu. It allows fast access to frequently used operations. A brief description of the functionality behind the icon is displayed if the mouse cursor remains on the icon for more than one second.

The toolbar comprises the groups described on page 351:

– The first group contains the *Test* function, which is used to check the accessibility of the specified system via ICMP, as well as the status of the SNMP master agent and of the Console Monitor subagent.

– The second group allows you to:

  – switch the current work area between the work area of a trap window and the work area of a command window

  – hide the community string

The third group contains functions for opening

– another command window and
– a window for displaying and setting filters

Command window work area

The command window work area comprises the following three parts:

1. area for setting the system and the SNMP parameters
2. command area
3. area for replying to a console query

System settings and status values

The parameters for SNMP are set in the command window work area. If the command window was opened from another window, the parameters from the calling window are taken over, otherwise the default values are set.

With the *System* Combobox you define the target system to which the command or reply is to be sent. The Combobox list is preset to the entries in the */etc/hosts* system file. The SNMP parameters are displayed in the second part.

Command area

A command can be input in this area. The result and main return code are displayed.

- *Command* Combobox

  Command input. Commands can be entered directly in the text filed and taken over by the Combobox mechanism as *Temporary Commands* in the Combobox list. The Combobox list is preset with the *Standard Commands* defined in the menu bar *Options* → *Settings* → *Commands*. The commands can also be edited here and *Temporary Commands* converted to *Standard Commands*.

Output	**Meaning**
Result	Displays the command result
Return code	Displays the command main return code

Action buttons:

The command is sent to the Console Monitor subagent for execution when this button is activated. The command output is shown in the result area. The SNMP message is output behind the input field if an error occurs.

When this button is activated, the process which supplies the command to the subagent and fetches the command result is terminated. Depending on the state, it is not possible to be certain that the command is not executed. A command that has already been sent by the Console Monitor is never aborted.

poll Redisplays the result of the last command.

Area for replying to a console query

The reply to a query received from the console can be input in the area. The values specified in the first area under system settings are used for the target system and SNMP parameters.

Additional parameters:

The *Reply* input field is used to input the reply text.

Action button:

When this button is activated, the reply is sent to the specified system. The result of the SNMP operation is displayed between the input field and the button. If the reply cannot be correctly processed by the BS2000 system, a console message is output in the trap window.

Command window help line

The help line functionality corresponds to that of the trap window (cf. page 349).

9.2.1.4 Filter window

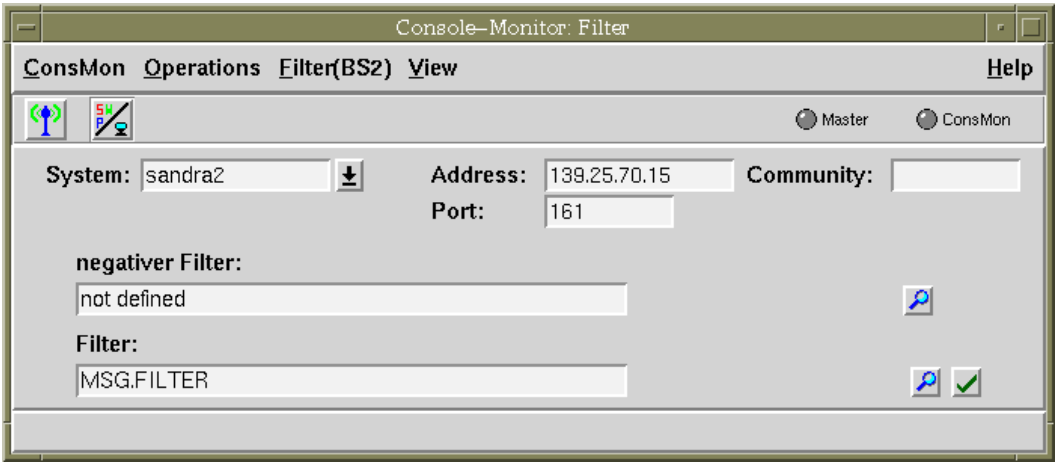


Figure 50: Filter window

The filter window is opened by calling the *Edit* function in the *Filter(BS2)* menu. It is used for displaying and modifying the BS2000 message filter file set on the Console Monitor. It allows traps to be filtered out before they are sent via the network to the management station, thus reducing the network load.

As with the trap window, the filter window is also divided into four areas:

1. the menu bar
2. the toolbar
3. the work area
4. the help line

Filter window menu bar

The menu bar comprises the menus *ConsMon*, *Operations*, *Filter(BS2)* and *View*.

The *Filter(BS2)* menu contains:

- The *Display* for querying the values for the negative filter and the filter file currently set at the agent and
- The *Modify* function for defining a new filter file.

The *View* menu contains the *Display Community* function for showing and hiding the community string.

Filter window toolbar

The toolbar contains a selection of the functions offered in the menu. It is provided to allow fast access to frequently used operations. A brief description of the functionality behind the icon is displayed if the mouse cursor remains on the icon for more than one second.

The toolbar comprises two groups:

– The *Test* function in the first group is used to check:
  – the accessibility of the specified system via ICMP
  – the status of the SNMP master agent and of the Console Monitor subagent
– The second group allows you to hide the community string.

Filter window work area

The filter window work area is divided into three parts:

1. area for setting the system and the transport parameters
2. area for displaying the file name for the negative filter
3. area for displaying and entering the name of the filter file.

System settings and status values

The parameters for SNMP are set in this area. The input fields are preset to the values from the calling window.

System Combobox Target system to which the command or reply is to be sent. The Combobox list is preset by the system list.

Specify the SNMP parameters in the second part.

negative filter

You can display the name of the BS2000 file currently set for the negative filter in this area.

The name of the current filter file is displayed in the *negative filter* output field if the *Show* action is executed.

Action button:

Activating this button displays the BS2000 negative filter file set for the specified system.

Filter

You can display the name of the currently set BS2000 message file in this area and, if necessary, set a different message filter.

The name of the current filter file is displayed if the *Show* action is executed in the *Filter* input/output field. The field can be edited. If the *Modify* action is executed, the name entered in the field is set as the BS2000 message filter file in the specified system.

Action buttons:

Activating this button displays the BS2000 message filter file set for the specified system. The SNMP message is output behind the input field if an error occurs.

Activating this button sends the entered filter to the Console Monitor which sets it as the current BS2000 message filter file. The SNMP message is output behind the input field if an error occurs. *General error* could be an indication that the specified filter file could not be set, e.g. because it does not exist.

Changes to the file can be made effective during current operation by resetting the displayed file name.

Filter window help line

The help line functionality corresponds to that of the trap window (cf. page 350).

9.2.1.5 Trap confirmation window

The *Trap-Confirmation* window is opened by calling the *Confirmation* function in the *Messages* menu. This window is used to display and change the trap confirmation.

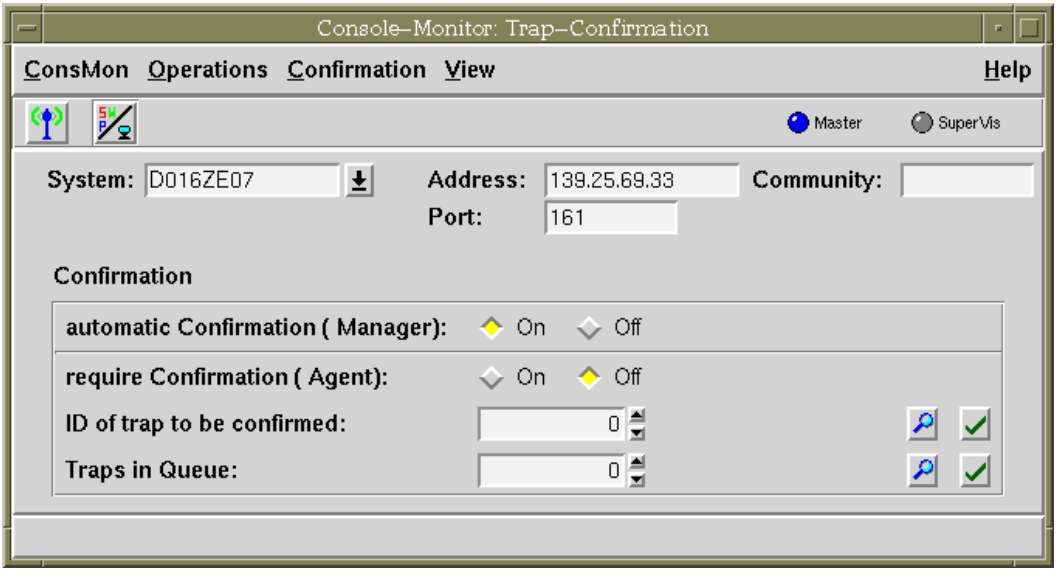


Figure 51: Trap-Confirmation window

The *Trap-Confirmation* window is divided into four areas:

– menu bar
– toolbar
– work area
– help line

Structure of the work area

The work area of the *Trap-Confirmation* window comprises three parts:

– area for setting the system and SNMP parameters

– area for setting automatic trap confirmation on the Console Monitor application

– area for setting automatic trap confirmation on the agent

System settings and status values

The SNMP parameters are set in this area.

System combobox

In the *System* combobox, specify the target system on which the message filter file is to be set. The combobox list is preset with the entries from the *System* list.

There are two possible display modes for the second part of the parameters; these can be set using either the menu or the toolbar:

– SNMP parameters
– agent status values

automatic Confirmation (Manager)

The *automatic Confirmation* option button can be used to enable and disable automatic trap confirmation on the manager.

require confirmation (Agent)

The *require confirmation (Agent)* option button can be used to enable and disable automatic trap confirmation on the agent.

ID of trap to be confirmed

The *ID of trap to be confirmed* dialog box displays the ID of the trap awaiting confirmation. If you set this value, the trap is considered confirmed.

Action Buttons:

Press this button to display the ID of the trap awaiting confirmation. This value can be changed (reduced).

Press this button to confirm the trap with the ID specified in the dialog box.

Traps in Queue

The *Traps in Queue* dialog box displays the number of traps located in the queue, because a trap awaiting confirmation has not yet been confirmed. If you set a lower value than that displayed, the number of traps in the queue is reduced to the specified value.

Action Buttons:

Press this button to display the number of traps in the queue.

Press this button to set the queue length specified in the dialog box.

9.2.1.6 Reactions dialog box

The *Reactions* dialog box is displayed when you call the *Options* → *Reactions* menu bar and it allows you to specify reactions for incoming messages.

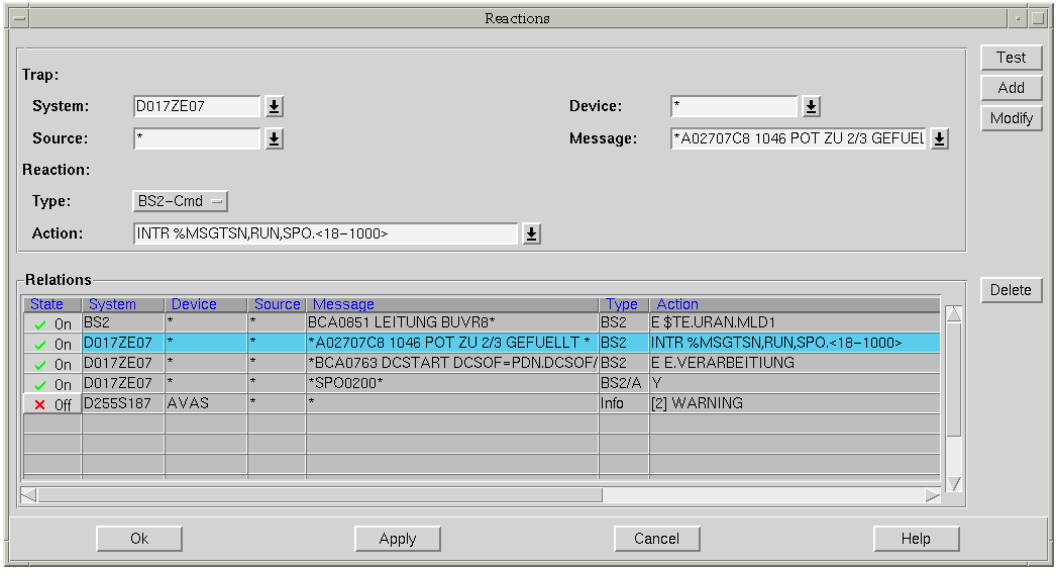


Figure 52: Reactions dialog box

The window is split up into four areas:

– Area for defining an event
– Area for defining the associated reaction
– Table for listing all defined reactions
– Action area (action buttons) for confirming the definitions and modifications.

Definition area / trap

In the area for the event definition, you specify the events for which reaction is to be initiated. You can use the following characters as wildcards:

– "*" for any string of characters
– "?" for any single character

Additional options are provided for the *Message* field. Here, the filter is formed by ANDing the parameters specified in the four combo boxes described below:

- Combobox *System*

  Specifying a system or a system group restricts the reaction to traps send by a particular system or system group. System groups must be specified in the form "[group name]". You do not have to enter the square brackets when entering the group names. You can edit system groups in the menu bar *Option* → *Settings* → *Group*. "*" is not allowed as a wildcard. The combobox list is pre-allocated by the system list and the system groups already defined.

- Combobox *Source*

  You can enter the source filters directly into the text field. The reaction is restricted to the relevant sources. Specifying "*" does not restrict the reactions to a particular source. The combobox list is pre-allocated filter entries defined in the menu bar *Option* → *Settings* → *Messages* under "Edit source filter".

- Combobox *Object*

  You can enter the object filters directly into the text field. The reaction is restricted to the specified objects. The combobox list is pre-allocated filter entries defined in the *Option* → *Settings* → *Messages* under "Edit object filter".

- Combobox *Message*

  You can enter the message filters directly into the text field. This filter is evaluated by comparing the specified string with to net message according to the "global style" rules. This also enables you to set a restriction for the weight of a message. The combobox list is pre-allocated filter entries defined in the menu bar *Option* → *Settings* → *Messages* under "Edit message filter".

  If the definition of the message filter start with "[RE]", the expression that follows is compared to the message as a regular expression. This means that you have more options than merely comparing according to "global-style" rules. In addition, almost any parts of expressions can be defined for adding to the command.

Definition area / reaction

In the definition are, you can also specify the reactions to defined events. A reaction is defined in two steps:

1. Specify type of reaction
2. Define reaction

You specify the type of reaction with the options menu *Type* which offers the following types of reactions:

Reaction type	**Description**	**Display in the "Reactions" table**
BS2-Cmd	BS2000 console commands without display of results	BS2
BS2-Cmd/out	BS2000 console commands with display of results	BS2/O
BS2-Response	BS2000 response to a query	BS2/A
Shell-Cmd	Local command without display of results	Shell
Inform	Colored message marking	Info
Delete	Delete a message	Del
Bell	Acoustic signal	Bell

Depending on the type of reaction you have defined in the options menu *Type*, you are offered a variety of input fields in which you can define the desired reaction.

- Reaction type *BS2-Cmd* or *BS2-Cmd/out*

  The *Action* combobox is offered, where the following operations are possible on BS2000 console commands with/without display of results:

  – direct entry in the text field of the combobox or
  – accept combobox list

  The combobox list is preassigned the standard commands that were defined previously in the menu bar *Options* → *Settings* →*Commands*.

- Reaction type *BS2 response*

  The *Action* input field is offered for you to enter a response. The TSN is automatically prefixed to the response.

- Reaction type *Shell-Cmd*

  An *Action* input field is offered to let you input the local command.

- Reaction type *Information*

  An *Action* input field is offered. To color the message, specify one of the keywords ERROR, WARNING or MESSAGE in the input field.

- Reaction type *Delete*
- Reaction type *Bell*

When you enter a command, the following options are available to accept part of the message in the command:

1. Keywords:

Keyword	**Meaning**
%ALL	refers to the entire message.
%DEVICE	refers to the object.
%SYSTEM	refers to the sender system.
%SOURCE	refers to the source
%MSGTXT	refers to the message text
%MSGWEIGHT	refers to the weight
%MSGNUMBER	refers to the message number
%MSGTSN	refers to the TSN
%MSGSEQNR	refers to the sequence number

2. Fixed position of the message:

   <a-b> denotes the characters between the positions a and b (including a and b); a maximum of 10 areas can be selected.

3. Regular partial expressions:

   %i: denotes the ith regular expression in the message.
   Prerequisite: the message filter must be defined as a regular expression.

4. Message inserts:

   &i denotes the ith insert of the message.
   Prerequisite: the message type must be stored in the file *bs2msge.dat*.

Optionally, you can specify a reaction delay of <time> minutes by prefixing “[<time>]” to the message. In this case, the reaction is not executed unless the message has not yet been deleted at start time.

Table of relations

In this area, the relations between the incoming messages and the generated reactions are displayed in tabular form. The table lines are assigned according to the system names. The table columns contain the following information:

Status (act)	Specifies whether the reaction is activated (*On*) or not (*Off*). Clicking the left mouse button on this field changes the status.
System	This parameter corresponds to the parameter *System* in the definition area / Trap (see page 350).
Object	This parameter corresponds to the parameter *Object* in the definition area / Trap.
Source	This parameter corresponds to the parameter *Source* in the definition area / Trap.
Message	This parameter corresponds to the parameter *Message* in the definition area / Trap.
Type	This parameter corresponds to the parameter *Type* in the definition area / Reaction (see page 371).
Action (Command)	This parameter corresponds to the parameter *Action* in the definition area / Reaction.

The reaction table is subject to the standard rules for extended selection.

Action buttons in the action area

Test	This action button is only active if at least one entry in the *Traps* table of the main window is selected. It is checked whether the specification in the *Definition area* / *Reactions* would cause a reaction on the occurrence of such an event. The result is indicated in a message window.
Add	The specifications in the *Definition area* / *Trap* and in the *Definition area* / *Reactions* are added to the table.
Modify	This action button is only active when exactly one entry is selected in the table. The specifications in the *Definition area / Trap* and in the *Definition area / Reaction* replace the selective entry.
Delete	This action button is only active when at least one entry is selected in the table. The selected entries are deleted.
Ok	The settings are accepted and the dialog box closed. Only definitions noted in the table are valid, not the definitions in the definition area.
Use	The settings are accepted and the dialog box remains open. Only definitions noted in the table are valid, not the definitions in the definition area.
Abort	The settings are rejected and dialog box closed.
Help	The specified help text is displayed.

Examples of reactions

Example 1:

When the following message arrives

```
%BCAM-000.060653 %BCA0763 /DCSTART DCSOF=PDN.DCSOF/ACK>
```

the following enter job is to be executed:

```
/E E.VERARBEITUNG
```

Message: *BCA0763 DCSTART DCSOF=PDN.DCSOF/ACK*
Type: BS2-Cmd
Action: E E.VERARBEITUNG

Example 2a:

Each query with the message number SPS0200, e.g.

```
?SPAA-000.060653 % SPS0200 TSN '7163':FORM 'FTPR' MOUNTED ON PRINTER 'L0'
REPLY (...)
```

should be answered with "Y":

```
/SPAA.Y
```

Message: *SPS0200*
Type: BS2 reply
Action: Y

Example 2b:

Each query with message number SPS0200 from printer "L0", e.g.

```
?SPAA-000.060653 % SPS0200 TSN '7163':FORM 'FTPR' MOUNTED ON PRINTER 'L0'
REPLY (...)
```

should be answered with "Y":

```
/SPAA.Y
```

Message: *SPS0200* MOUNTED ON PRINTER 'L0'*
Type: BS2-Response
Action: Y

Example 3:

When the following message arrives

```
%1848-000.060653 BINDEX A02707C8 1046 POT 2/3 full
```

the following command should be issued:

```
/INTR 1848,RUN,SPO,BINDEX
```

The TSN and program name “BINDEX” should be accepted.

Message: *A02707C8 1046 POT 2/3 full*
Type: BS2-Cmd
Action: /INTR %MSGTSN,RUN,SPO,<18-1000>

Example 4:

All messages from source “Hardware” should be logged in the file *./trace* on the local system.

Source: *Hardware*
Message: *
Type: Shell-Cmd
Action: cmd.exe /C "echo { %ALL} >>./trace"

Example 5:

Queries that are left unanswered for longer than two minutes are marked as warnings.

Source: *
Message: <*> \?*
Type: Information
Action: [2] WARNING

9.2.1.7 Trap filter

For traps in the general format, the trap filter allows you to assign values for “source” and “device” to the traps. In addition, the trap filter can be used to completely suppress traps. (This facility does not apply to Application Monitor-specific traps.)

Files and directories

A *trap.cnf* file is produced in the conversion. The *trap.cnf* file must be located under *<CMONBASE>/config*.

Entries in the *trap.cnf* file

An entry in the *trap.cnf* file has the following syntax:

```
<filter> <action>

<filter> ::= <IP address>:<community>:<trapoid>
```

<IP address>
IP address or * (any IP address)

<community>
Community string or * (any community)

<trapoid>
OID of the trap or * (any trap OID)

```
<action> ::= ignore | <host>:<source>:<device>:<text>
```

ignore
Trap is to be ignored.

<host>
IP address of the host or * (transfer unmodified)

<source>
Source name or * (transfer unmodified)

<device>
Device name or * (transfer unmodified)

<text>
Text or * (transfer unmodified)

In the case of <source>, <device> and <text>, the following options exist for transferring parts of the trap variable binding:

– `%V(<oid>)` is replaced with the variable binding value.

– `%I(<oid>)` is replaced with the variable binding instance.

The prerequisite here is that the MIB is known to the Console Monitor subagent.

Examples:

```
*:*:1.3.6.1.4.1.231.99.0.333   ignore
*:*:1.3.6.1.4.1.231.99.0.444   *:TestSource:TestDevc::This is text
*:*:1.3.6.1.4.1.231.99.0.555   ignore
*:*:1.3.6.1.4.1.231.99.0.666   139.25.105.176:TestSource:TestDevc:*
139.25.22.22:*:1.3.6.1.4.1.231.2.34.2.0.301   *:supervisor:%V(1.3.6.1.4.1.231.2.34.1.2.2.1.1):Started
```

9.2.1.8 Settings of the options dialog box

The *Options* dialog box is opened by calling the *Options* → *Settings* function and is used for setting parameters for the current session. If required, the settings can be saved when the management application in use is terminated.

The *Options* dialog box is described on page 389 ff.

9.2.2 PMBS2 application for the Performance Monitor

PMBS2 supports the output of SM2 data supplied by the SSA-SM2-BS2 Performance Monitor on the management station and offers a graphical overview of the performance data for all BS2000/OSD systems in a network.

Functionality

The Performance Monitor also supplies information on SM2 itself:

– status of SM2
– version
– measurement interval size
– sampling cycle

The actual measurement values can be divided generally into seven object groups, corresponding to the SM2 report groups.

Object group	Measurement values
BASIC	Basic group
TIMEIO	CPU utilization and I/O activity
MEMORY	Main memory and virtual address space utilization
CATEGORY	Main memory occupation by the four standard task categories
DEVICES	Input/output operations to peripheral devices during a measurement interval
UTM	Application-specific data from UTM applications
PERTASK	Resource utilization values of separate tasks

Two types of objects are discriminated for displaying the values:

Scalar objects: have only one instance. The BASIC, MEMORY and CATEGORY object groups belong to these.

Table objects: are objects which may have several instances. This allows, for example, the information on CPU utilization and the measurement values for the I/O report to be displayed separately for the generated logical machines as well as for the complete system. The values for DEVICES, UTM and PERTASK are also listed in tabular form. A specific instance of a table object can be regarded as a scalar object.

Starting and stopping PMBS2

PMBS2 is started from the shell level on UNIX systems by calling the *PerfMon* procedure. The application main window is opened.

On WindowsNT, the application is started from the *SNMP Management Applications* program group. Alternatively, you can call it by double-clicking on *PerformanceMonitor* in the *<tcldir>\appl\pmbs2* directory.

PMBS2 can be stopped from any window by selecting the *Terminate PerfMon* function in the *PerfMon* menu. A dialog box prompts you to confirm this action, thereby preventing unintentional termination. Parameter changes are either saved, rejected or queried, depending on the parameters set for *Save*.

9.2.2.1 Main window

The main window is opened by calling PMBS2. Windows for diagrams can be opened from here. PMBS2 is terminated if this window is closed.

The window is divided into the following four areas:

1. menu bar
2. toolbar
3. work area
4. help line

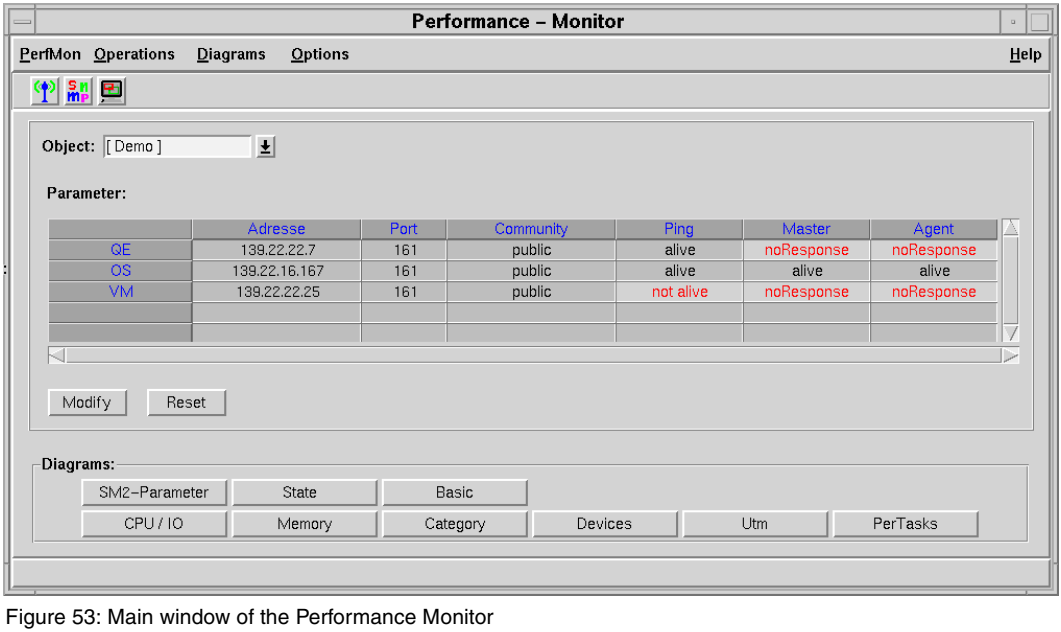


Figure 53: Main window of the Performance Monitor

Menu bar

The menu bar comprises the *PerfMon*, *Operations*, *Diagrams*, *Options* and *Help* menus.

The *PerfMon* menu contains the *Terminate PerfMon* item.

The *Operations* menu contains:

- The *Ping test* function, with which the reachability of the specified system via ICMP can be checked. If the system can be reached, the word *alive* is entered in the *Ping* column of the table in the work area for the system, otherwise *not alive* is entered. This function is only active UNIX systems. Parameters for the ping function can be set in the *Options* → *Settings* → *Ping* dialog box.
- The *SNMP test* function, with which the status of the SNMP master agent can be checked by calling the *SysUpTime*. If the master agent for the specified system can be reached, the word *alive* is entered in the *Master* column of the table, otherwise the SNMP error message is output.
- The *Pmon test* function, with which the status of the Performance Monitor can be displayed by querying the version. If the subagent of the specified system can be reached, the word *alive* is entered in the *Agent* column of the table, otherwise the SNMP error message is output.

The above three test options are listed in ascending order.

The *Diagrams* menu contains the entries for calling the diagram window for the different predefined diagram groups. Each diagram group corresponds to an SM2 measurement value group which is checked by the Performance Monitor. The special appearance of the diagrams and the values to be displayed depend on the diagram group and whether one or several systems are being monitored.

Options contains the following entries:

– *Settings*, which opens an options dialog box in which the settings can be edited.

– *Save* → *Settings*, which saves all values of the setting areas in the configuration file immediately. The menu entry *Save* is only active when all dialog boxes are closed.

The *Help* menu provides information on the version, on *Saving, the* active window, MIB objects and index (overview).

Toolbar

The toolbar contains a selection of the functions offered in the menu. It allows fast access to frequently used operations. A brief description of the functionality behind the icon is displayed if the mouse cursor remains on the icon for more than one second. Both main and diagram windows have a toolbar.

The toolbar comprises the

Ping test function, with which the reachability of the specified system via ICMP can be checked. *Ping test* is active only for UNIX systems.

– *SNMP test* function, with which the SNMP master agent status can be checked by calling the *SysUpTime*

– *PerfMon test* function, with which the status of the Performance Monitor can be checked by querying the version

Work area

The main window work area has two parts:

1. the area for setting the system to be monitored or the system group and its parameters, and their tests
2. buttons for calling the windows for the predefined diagram classes

Settings and tests

The target system IP address, a port and a community string must be specified to enable a message to be sent correctly to a system. The community string defines the access authorization and its extent. These settings can be made in this area.

- *Object* Combobox

  The lines for the subsequent table and the system names for further windows are preset by specifying an object. Groups are represented by [Groupsname]. The square brackets need not be specified when entering group names. To edit groups, use *Options* → *Settings* → *Groups*. The combobox list is preset by the system list and the system groups already defined.

- *Parameters* table

  The SNMP transport parameters and the reachability of the system, master agent and performance subagent for each system are displayed in the table.
  The address, port and community can be edited directly in the table.Changes must be confirmed with the *Modify* button. The values can be reset with the *Reset* button as long as the changes have not been confirmed.

The reachability of the system via ICMP is indicated in the *Ping* column when the *Operations → Ping test* function is called.

The reachability of the master agent is documented in the *Master* column when the *Operations → SNMP test* function is called.

The reachability of the performance subagent is displayed in the *Agent* column when the *Operations → Pmon test* function is called.

Action buttons

Modify Makes changes made to the SNMP parameters effective. However, the changes do not affect already opened diagram windows.

Reset Resets non-confirmed changes.

Diagrams

This area contains the buttons for calling the windows of predefined diagram classes. Each diagram class corresponds to an SM2 measurement value group checked by the agent. The windows can also be opened via the *Diagrams* menu.

9.2.2.2 Diagram window

A diagram window is called from the main window and takes over the entry for the system or system group as well as the SNMP transport parameters.

As with the main window, this window is also split into four areas:

1. the menu bar
2. the parameter bar
3. the work area
4. the help line

Menu bar

The menu bar comprises the *PerfMon*, *Operations*, *Thresholds* and *Help* menus.

The *PerfMon* menu contains the *Terminate PerfMon* item.

The *Operations* menu contains the *Close* entry which closes the diagram window

The *Threshold* menu is only activated for curve diagrams. The *Active* check button activates/deactivates threshold monitoring. The *Settings* menu option opens a dialog box for setting the threshold parameters.

Active activates/deactivates threshold monitoring

Setting opens a dialog box for setting the threshold parameter

The help menu is identical to that of the main menu and provides information on version, active window, MIB objects and the index.

9.2.2.3 Parameter bar

Settings for influencing the display of diagrams can be made in the parameter bar.

System Combobox

System or system group for which the performance values are to be displayed. The input field is preset to the system or system group selected in the main menu. The entry for a single system can also be changed in the diagram window, but this is not possible for system groups. The Combobox list is preset by the entries in the system list.

Poll interval scale

Time interval used by PMBS2 for polling the values of subagents. Since SM2 only updates its values in a specified cycle with a default 120 of seconds (*SM2Parameter* → *Interval*), poll times shorter than this are not meaningful.

Action buttons:

– *Start*

  Starts and stops the poll. A single request is sent if the poll interval is set to “0”.

– *Grid*

  Superimposes a grid on curves and histograms.

– *Stack*

  Toggles between parallel and stack mode display for histograms.

Work area

The diagram window work area is in the form of a “notebook”, with one or more diagrams shown on each page. The diagram type depends on the type of MIB object to be displayed and the number of monitored systems.

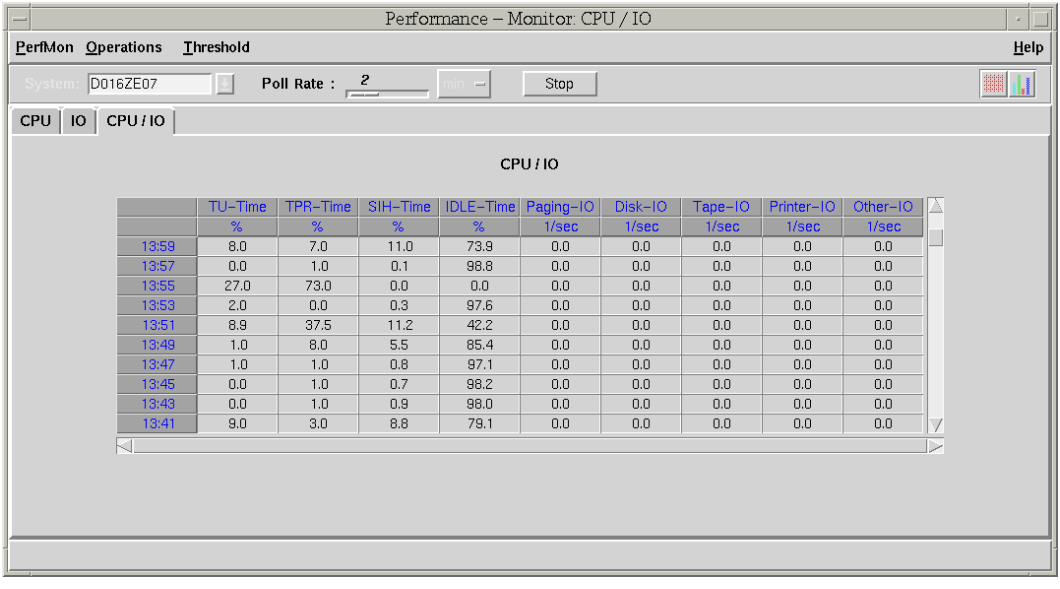
The following diagram types are possible:

Form

Forms are used for displaying the current values of scalar MIB objects for a single system. The values are shown in simple output fields.

Table

Tables are used for displaying the current values of scalar MIB objects for multiple systems. They are also used for displaying changes over time for the values of a single system. The values are listed in tabular form.



	TU-Time	TPR-Time	SIH-Time	IDLE-Time	Paging-IO	Disk-IO	Tape-IO	Printer-IO	Other-IO
	%	%	%	%	1/sec	1/sec	1/sec	1/sec	1/sec
13:59	8.0	7.0	11.0	73.9	0.0	0.0	0.0	0.0	0.0
13:57	0.0	1.0	0.1	98.8	0.0	0.0	0.0	0.0	0.0
13:55	27.0	73.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13:53	2.0	0.0	0.3	97.6	0.0	0.0	0.0	0.0	0.0
13:51	8.9	37.5	11.2	42.2	0.0	0.0	0.0	0.0	0.0
13:49	1.0	8.0	5.5	85.4	0.0	0.0	0.0	0.0	0.0
13:47	1.0	1.0	0.8	97.1	0.0	0.0	0.0	0.0	0.0
13:45	0.0	1.0	0.7	98.2	0.0	0.0	0.0	0.0	0.0
13:43	0.0	1.0	0.9	98.0	0.0	0.0	0.0	0.0	0.0
13:41	9.0	3.0	8.8	79.1	0.0	0.0	0.0	0.0	0.0

Figure 54: Display the values of scalable MIB objects in a table

Curve diagrams

Curves are used to document the changes of values for one or more systems over time.

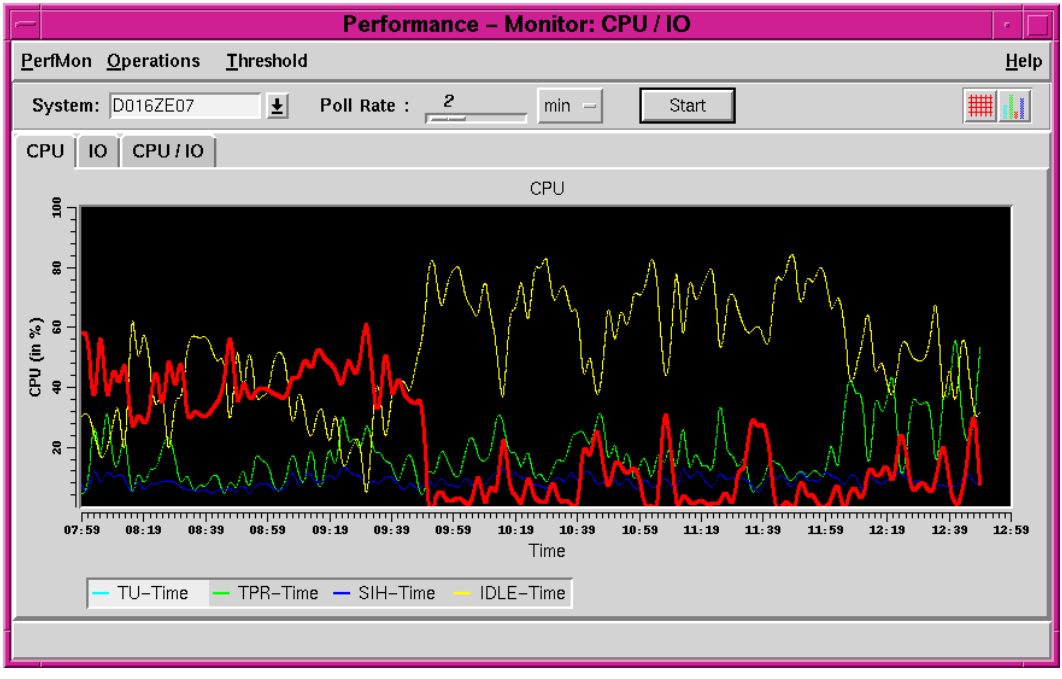


Figure 55: Curve diagram

Pull the mouse cursor over an area of the curve (click - pull - click) to zoom the selected area. Click the right mouse button to return to the original setting.

When the mouse cursor is positioned on an element of the legend, the associated curved is highlighted.

Histograms

Histograms show the current values of multiple objects for multiple systems.

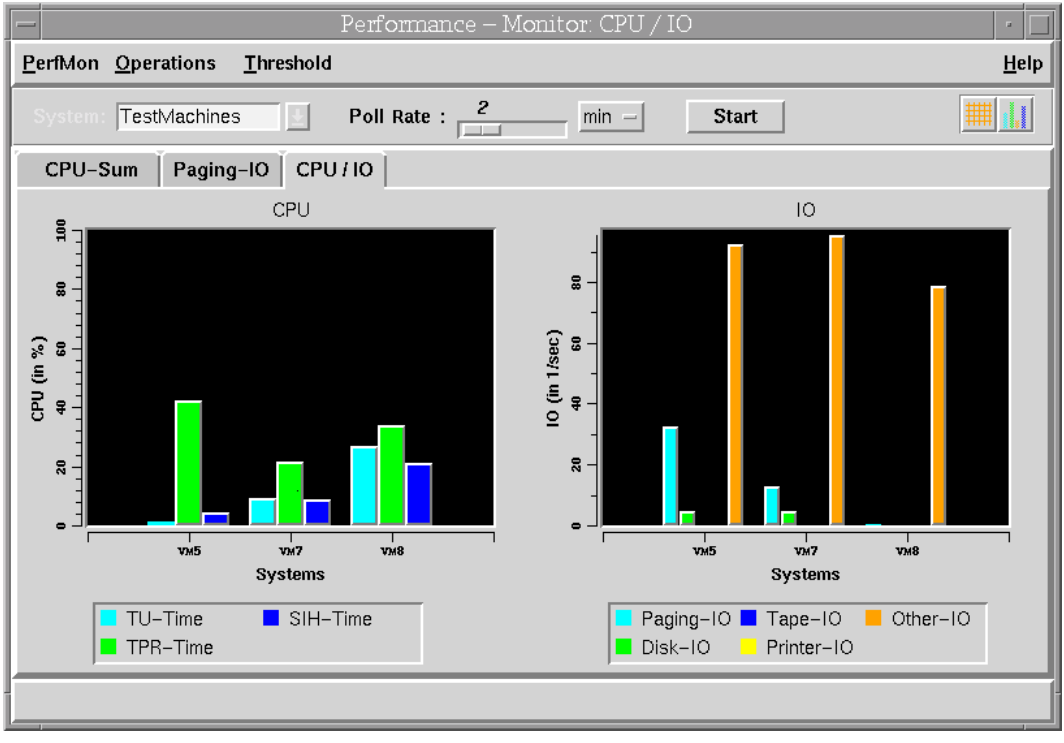


Figure 56: Display of the current values of several systems ina histogram

Help line

The configuration of a group is displayed in the help line by double-clicking on a group name in the system input field. The SNMP transport parameters are displayed if a single system is entered under system. SNMP error messages are also shown in the help line.

9.2.2.4 Generating diagrams

A series of diagrams are generated for each measurement value group for monitoring a single system or a group of systems. The description files named *gen....tcl* should not be changed if possible. Most of the parameters are self-explanatory or are commented in the files. Small changes and adjustments can be made without difficulty if you have sufficient knowledge of the performance MIB.

The status diagram plays a special role. It shows the SM2 measurement range that the subagent currently has access to and, if necessary, why it does not.

9.2.2.5 Threshold dialog box settings

The *Thresholds* dialog box is opened by calling the *Thresholds* → *Settings* function and is used for setting the threshold parameters for the current session. Threshold parameters can only be set for curve diagrams.

The dialog box is shown in the form of a “notebook” and structured according to the curve defined in the corresponding diagram window.

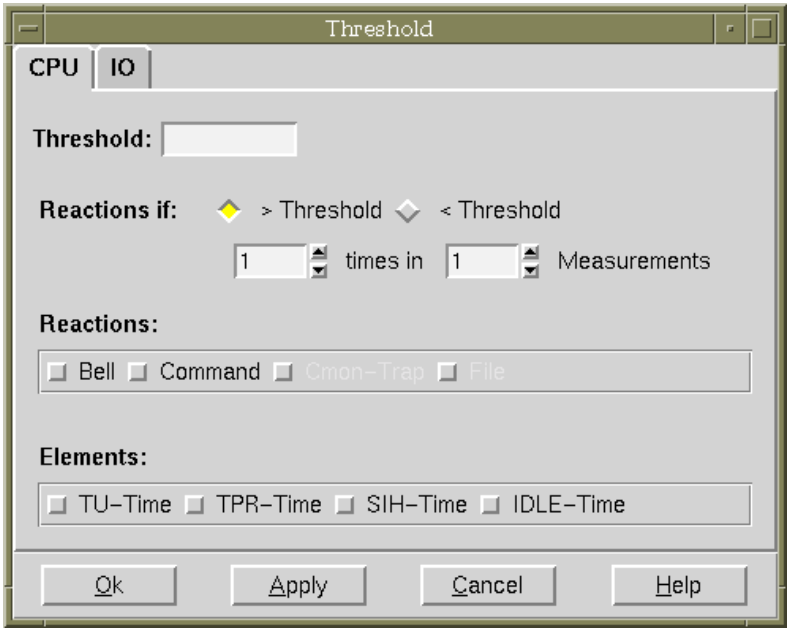


Figure 57: Dialog box “Threshold values“

Field	Meaning
Input field *Threshold*	Specifies the threshold value. No threshold is evaluated if none is defined.
Radio button *Reaction to*	Specifies whether the reaction is to be triggered if the value exceeds or falls below the limit.
Input fields *x* "times in" *y* "Measurements"	A reaction is only triggered if the threshold is exceeded at least *x* times in the last *y* measurements **and** the value of the current measurement exceeds the specified threshold.
Check button *Reaction*	Specifies which reaction is to be triggered if the threshold value is exceeded. The shell command and the file are specified in the parameter file.
Check button *Elements*	Specifies the diagram elements to which threshold monitoring is to be applied.
OK	Accepts the settings and closes the dialog box.
Apply	Accepts the settings and leaves the dialog box open.
Cancel	Cancels the settings and closes the dialog box.
Help	Displays the help text for the threshold dialog box.

If a threshold is set, it can be shifted from within the diagram window using the middle mouse button.

9.2.2.6 Options dialog box settings

The *Options* dialog box is opened by calling the *Options* → *Settings* function, and is used for setting parameters for the current session. For example, the interface language (German/English) can be set. If required, the settings can be saved when the management application in use is terminated. The following description applies to both CMBS2 and PMBS2.

The dialog box is shown in the form of a “notebook”. The following table shows the ranges available for the relevant management application:

Range	CMBS2	PMBS2
Systems	X	X
Groups	X	X
Ping	X	X
SNMP	X	X
Messages	X	
Commands	X	
Reactions		X
Protocol	X	
Save	X	X

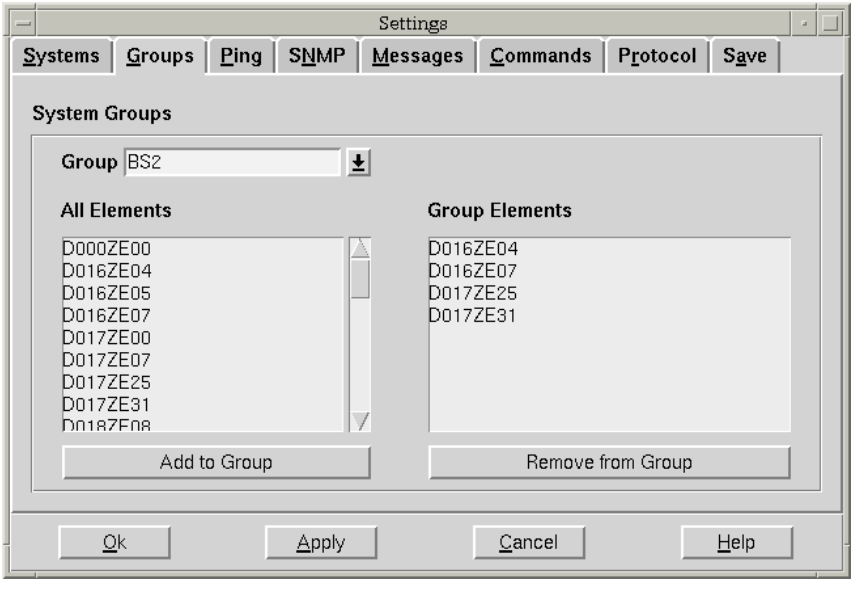


Figure 58: Option dialog box of the console monitor

Systems

Systems applies both to CMBS2 and PMBS2.

The Console Monitor generates a system list when it is started. Each target system in the list is defined by the system and the SNMP parameters (IP address, port and community).

The system list is generated in three steps:

1. The system list is created by reading the system descriptions from the configuration file, which contains the systems from the previous sessions in the setting range.
2. Systems listed in */etc/hosts* but not yet in system list are added to the system list. The ports set to the default port 161 and the communities to the default *public* (UNIX only).
3. The system list is sorted alphabetically. The system list is edited in the *Systems* setting range:
   – SNMP parameters of individual systems can be modified,
   – New systems can be added to the system list.

Field	Meaning
Combobox *Name*	Current system list with unique system names. Entry in the input field: blank No system selected. The SNMP parameters *Address*, *Port* and *Community* also remain blank. existing system name The SNMP parameters are set accordingly. new system name Any SMNP parameter set are retained; unassigned parameters are given default values. *OK* or *Apply* in the action range expand the system list to include the new system names with the relevant SNMP parameters for the current session. You can save the new system list in the configuration file under *Options* → *Settings* → *Save*.
Address	IP address of the target system in four-byte, decimal point notation. Default: 127.0.0.1.
Port	Numerical port of the SNMP service on the target system. Default: 161

Field	Meaning
Community	String that defines the access authorization and authorization scope for the SNMP management at the target system. Default Community: *public*

Groups

Groups applies to both CMBS2 and PMBS2.

Systems can be combined to form groups. The group names, unlike system names, are output in square brackets *[groupname]*.

Field	Meaning
Combobox *Group*	Current group list with unique group names. Entry in the input field: blank No group selected. The *Group elements* list box also remain blank. existing group name The *Group elements* contains the system name of the group. new group name Any existing system names are retained in the *Group elements* list box Click *OK* or *Apply* in the action range to create the new group and make them available in the console monitor for the current session. You can save the new group in the configuration file under *Options* → *Save*.
Combobox *All elements*	All system names from the current system list.
Combobox *Group elements*	System names of the group entered in the *Group* Combobox.
Action button *Add to group*	The system names selected in the *All elements* list box are taken over into the *Group elements* list box.

Field	Meaning
Action button *Remove from group*	The system names in the *Group elements* are removed from this list box. It is possible to delete an existing group completely by removing all its elements and pressing *OK* or *Apply* in the action area.

Ping

Ping applies to both CMBS2 and PMBS2.

Field	Meaning
TimeOut	Defines the time (in seconds) after which a ping request is taken to be unanswered.
Retries	Number of ping retries.
Delay	Time (in seconds) between the separate retries.
Radio button *Separate window*	Possible values: Yes Results of ping requests are displayed in a separate window. No Results of Ping requests are not displayed in a separate window.

SNMP

SNMP applies to both CMBS2 and PMBS2.

Global default value for the system entered when the main window is opened. The Combobox list is preset with the hosts entered in the system list.

Field	Meaning
Version	SNMP version in current use. *Version* is an output field.
TimeOut	Defines the time (in seconds) after which an SNMP request is rejected as unanswered.
Retries	Defines the number of attempts to be made for determining if an answer is available for an issued BS2000 command. The value 0 means that there is no limit.
Input field *Port* (CMBS2 only)	Port for receiving traps (162 or 1024)

Field	Meaning
Delay	Defines the time (in seconds) between the retries to determine if an answer is available for an issued BS2000 command.

Messages

Messages applies to CMBS2 only.

You can use the *Messages* setting range to set parameters for the trap window.

Field	Meaning
Input field *Max. number*	Maximum number of messages that can be displayed in the trap window. The oldest message is deleted if this number is exceeded.
Radio button *Delete answered questions*	Yes Questions answered are deleted in the trap window. No Questions answered are not deleted in the trap window.
Folder *Edit object filter* *Edit source filter* *Edit message filter*	The filter input field is preset to the relevant value of the trap window.The *Add* action button can be used to preset the filters in the list box. The input field also enables new filters to be added. If a filter has been selected in the list box, it appears in the input field. The *Modify* thus provides a user-friendly way of changing the selected entry in the list box.
Action button *Add*	Adds valid entries as filters in the list box, provided that they do not already exist.
Action button *Modify*	If a filter is selected in the list box, it appears in the input field. Pressing the *Modify* action button replaces the filter selected in the list box with the one modified in the input field.
Action button *Listbox*	Lists the current filters.
Action button *Delete*	Pressing the *Delete* action button deletes the selected filter from the list.

Commands

Commands only applies to CMBS2.

When the Console Monitor is started, *Standard commands* are read from the configuration file. You can create *Temporary Commands* by inputting new commands in the command window or using this setting range. The commands area allows you to edit *Temporary* and *Standard Commands*, and convert them back and forward.

Field	Meaning
Input field	The command input field is used to input new *Temporary commands* or to modify existing commands in the *Temporary* or *Standard commands* list boxes. You can use the *Add* action button to add a new command to the *Temporary commands* list box. In order to make a change, you must select a command in the relevant list box. Then press the *Modify* action button to replace the selected command with the one modified in the input field, provided that the modified command does not already exist in one of the list boxes.
Listbox *Temporary Commands*	Lists the temporary commands. This list is empty at the start of a session. These commands are not saved after the current session, i.e. they are no longer available in the next session.
Listbox *Standard Commands*	Lists the standard commands. This list contains the standard commands of the previous session. If standard commands are modified and are still to be available for future sessions, you must save them under *Options* → *Save*
Action button *Add*	Valid entries in the input field are added as new commands in the *Temporary commands* list box, provided that the do not already exist in either of the list boxes.
Action button *Modify*	A command selected in a list box appears in the input field. The command modified in the input field replaces the one selected in the list box when you press the *Modify* action button.
Action button →	The entries selected from the *Temporary commands* list box are moved to the *Standard commands* listbox. This extends the scope of the *Standard Commands*.
Action button ←	The selected entered are moved from the *Temporary commands* list box. This reduces the scope of the *Standard commands*.

Field	Meaning
Action button *Delete*	The selected entries are deleted from the *Temporary command*s list box. *Standard commands* can only be removed completely via the *Temporary commands* list box.

Logging

In this area, you can set the logging for reactions.

Field	Meaning
Radiobutton *Log*	Possible values: Yes Reactions are logged. No Reactions are not logged.
Input field *File*	Name of the file used as log file. Currently, this name cannot be changed.
Radiobutton *Start mode*	Possible values: – Overwrite: The file is overwritten after each restart. – Append: The file is continued after a restart. – Save and new: The existing file is saved and a new one created.
Radiobutton *OnLine*	Possible values: Yes The result of *SetRequests* for a BS2000/OSD console commands is displayed in the *Traps* table of the main window. Prerequisite: the radio button *Log* is active. No The result is not displayed.

Reactions (PMBS2 only)

The *Reactions* setting range defines general parameters for reactions when thresholds are exceeded. These reactions require that a threshold is defined in the *Threshold* dialog box and that the *Active* button is pressed in the *Thresholds* menu.

Field	Meaning
Input field *File*	Specifies the file in which threshold violations are logged.
Input field *Command*	Command executed when a threshold is exceeded. Certain keywords can be used here.
Input field *Trap:Manager*	Specifies the system to which the trap is to be sent.
Input field *Trap*:*Port*	Specifies the port to which the trap is to be sent on the target system.
Input field *Trap*:*Object*	Identifies the object in the trap. The object is displayed in the “Object” column of the Console Monitor application table.
Input field *Trap*:*Source*	Identifies the source in the trap. The source is displayed in the “Source” column of the Console Monitor application table.
Input field *Trap*:*Message*	Message text of the trap. The message text is displayed in the “Message” column of the Console Monitor application table.

Save

Save applies to both CMBS2 and PMBS2. The settings for saving are specified here.

Field	Meaning	
Radiobutton *Save on exit*	Yes	When you terminate the application normally, the values of the setting ranges are stored automatically in the configuration file without prompting.
	No	Changes to values in the settings range are not saved when you exit the application.
	Query	A dialog box appears when you terminate the application normally. When you answer the query, you can save changes to the settings in the configuration file.
Radiobutton *Language*	Select the language for the restart.	

10 Web access to management information

In addition to processing SNMP requests, the master Agent can also provide access to managed information through the World Wide Web. This allows both traditional SNMP management applications and web browsers to retrieve and change the information made available by any EMANATE Subagent.

10.1 Overview

The Master Agent listens to the network for two kinds of requests:

– On the SNMP port (usually UDP 161) the Master Agent listens for SNMP Get and Set requests.

  In response to the received SNMP requests, the Master Agent sends SNMP GetResponse messages.

– On the web-based Management port (usually TCP 280), the Master Agent listens for HTTP connection requests.

  In response to the received SNMP requests, the Master Agent sends SNMP GetResponse messages. In response to the HTTP message received, the Master Agent sends an HTML page back to the web browser. The HTML page may be a pre-defined custom page or an automatically generated page containing the values of MIB variables found by “walking” the MIB tree.

  The Master Agent is designed to allow safe, controlled access to the managed information HTTP-Engine.

HTTP requests are processed in the same manner as SNMP requests. After analysis of an SNMP or an HTTP request, the master agent stores the relevant components of the request in the EMANATE event queueing subsystem, an internal queue of the master agent. From here, the master agent obtains the information in the usual manner via the EMANATE subagent.

As soon as the master agent has received the information, it generates an SNMP GetResponse message or an HTML page, depending on the type of request, and forwards this to the sender of the original message along with the desired information.

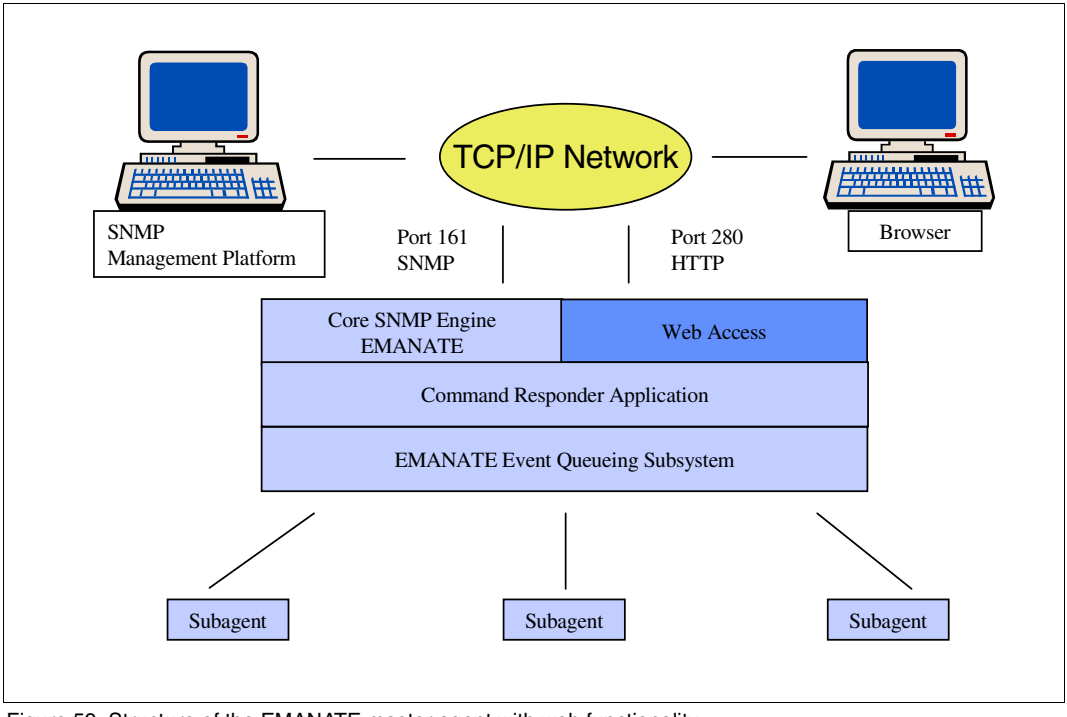


Figure 59: Structure of the EMANATE master agent with web functionality

10.2 The BS2000/OSD web agents interface (web interface)

This chapter discusses the following topics:

– Connection to the BS2000/OSD web agent

– Automatically generated web pages (subtree functionality)

– Custom pages (custom page functionality)

10.2.1 Connecting 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:

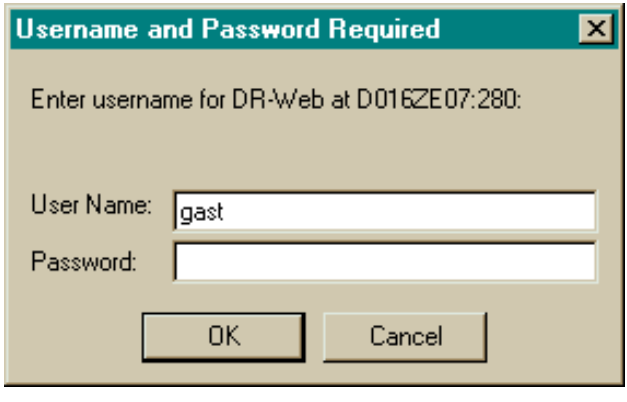


Figure 60: Entering the username and password

The username and password must be configured at the agent (see section “Security configuration” on page 28). 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).

When the user has successfully entered a valid username and password, the “welcome page” is presented by the browser.

BS2000/OSD Web Agent Welcome Page

The figure below shows the standard welcome page with hyperlinks to the subtree and custom page functions.

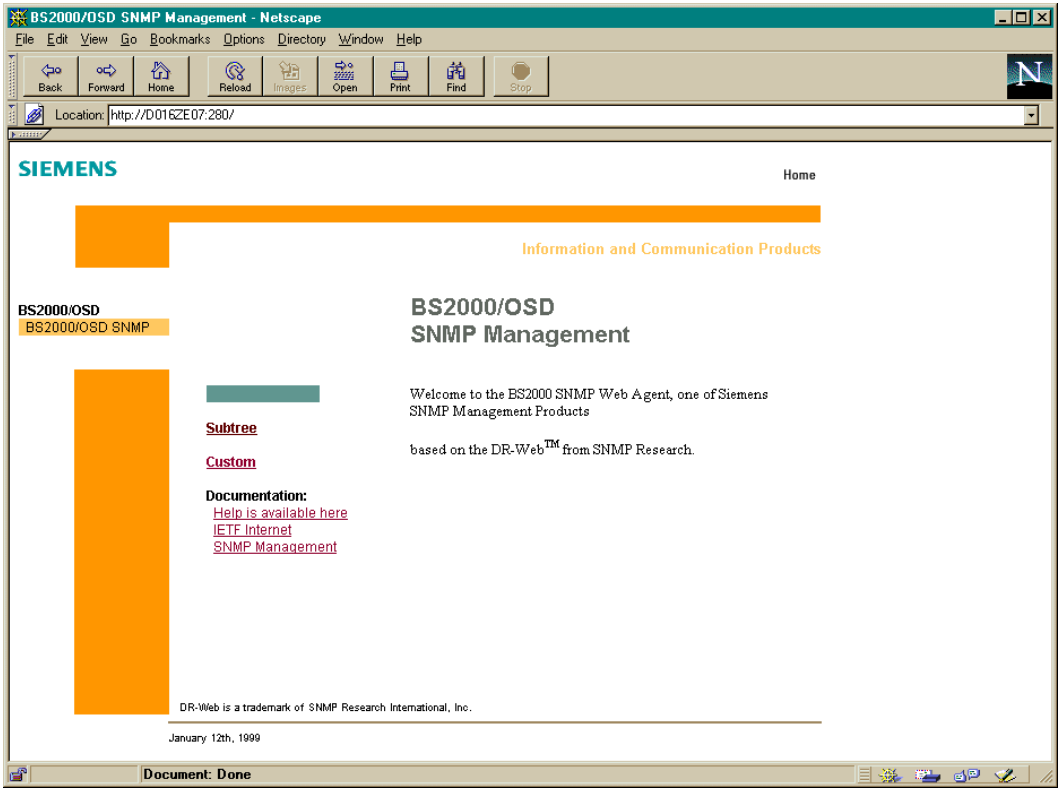


Figure 61: Welcome page of the web agent

10.2.2 Subtree functionality

Click on the *subtree* hyperlink in the DR-Web Welcome Page to access the subtree functionality to facilitate MIB browsing.

10.2.2.1 Subtree page of the web agent (DR-Web subtree page)

The subtree functionality are offered via the subtree page of the web agent. To access the subtree page with the browser, click on the subtree hyperlink on the welcome page or enter ”http://*ip_adresse*:280/subtree/” in the address field of your browser.

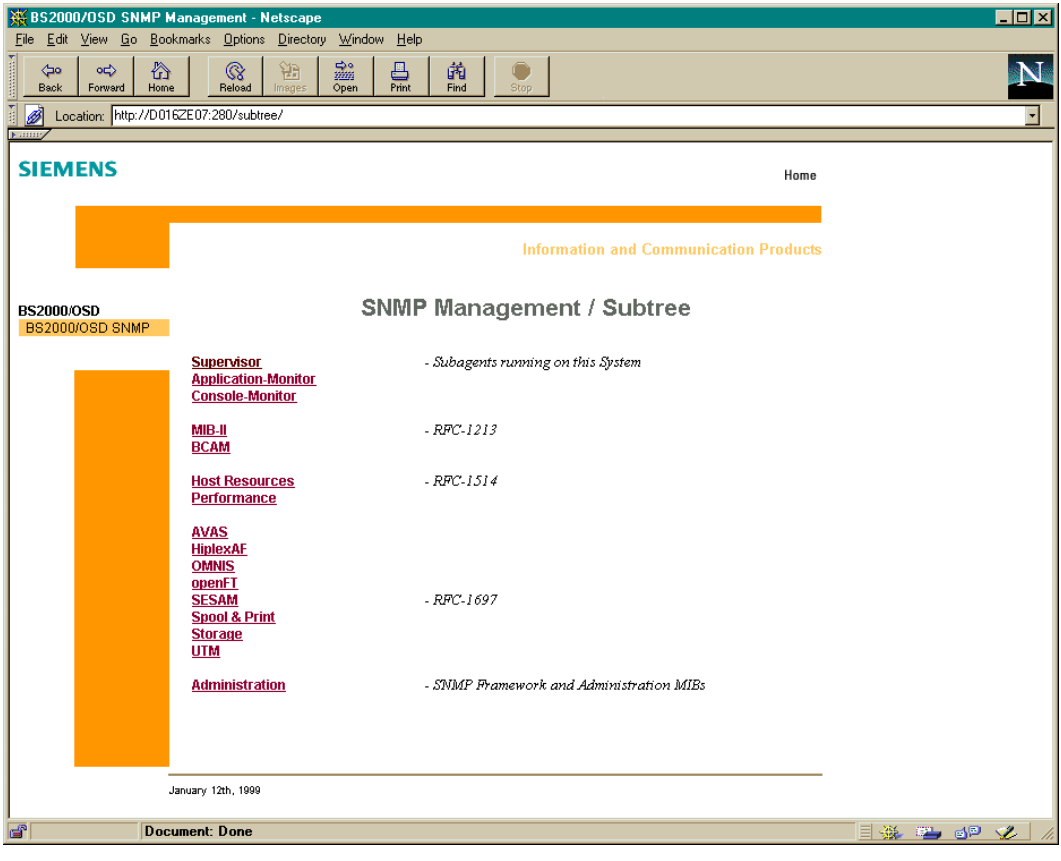


Figure 62: DR-Web subtree page

The subtree page contains hyperlinks to subtree addresses, which can be used to display management information that is accessible via the web agent. This page is pre-configured for fast access to all MIBs supported by SNMP Management in BS2000/OSD. For example, to view the entire MIB-II subtree, click on the MIB-II hyperlink.

To view only the system group of the MIB-II, enter the URL "http://*ip_adresse*:280/subtree/system/" in the address field of your browser.

10.2.2.2 Subtree URL - GetRequest functionality

The web agent can be directed to retrieve specific MIB subtrees when they receive a URL containing an OBJECT IDENTIFIER (OID). The OID may be specified in English or numeric form, because the web agent can perform name-to-OID translation.

Example: Supervisor MIB

The Supervisor MIB is used to monitor all the subagents logged on at the master agent. If the supervisor subagent is running in the target agent, then the MIB objects described by this MIB are available through DR-Web.

To retrieve the entire supervisor subtree, enter either:

– "subtree/sniSupervisor" in the address field of your browser or

– "subtree/1.3.6.1.4.1231.34" in the address field of your browser or

– click on the *supervisor* hyperlink on the DR-Web subtree page.

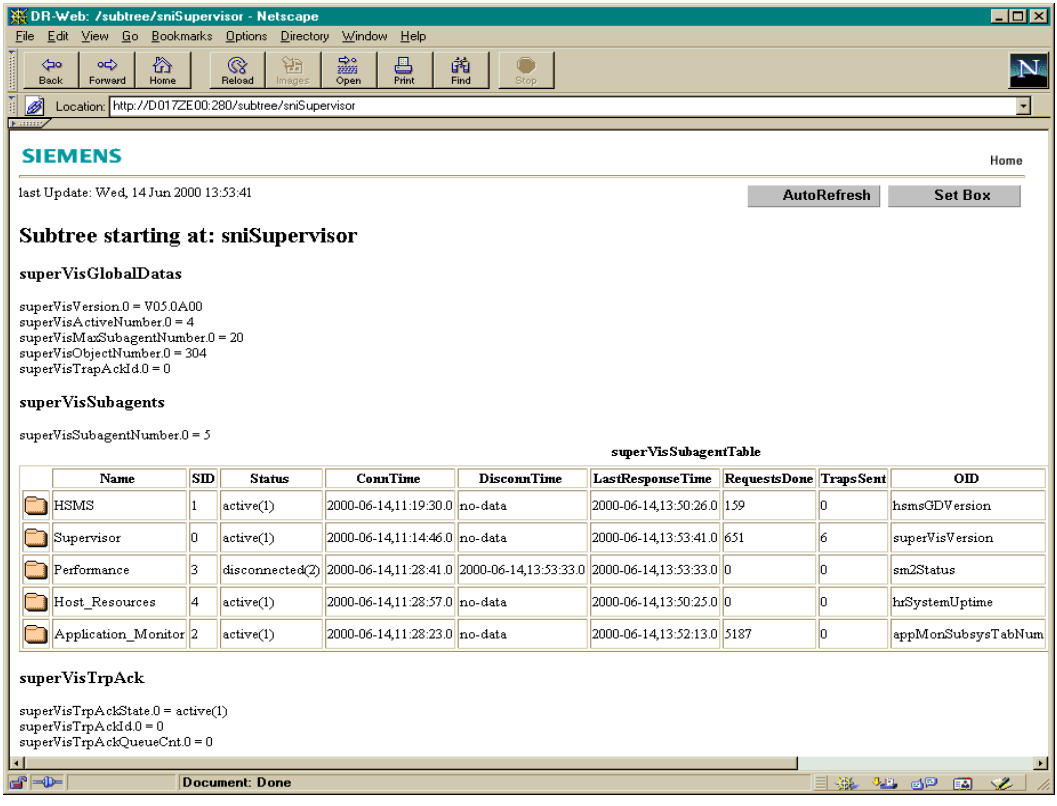


Figure 63: *subtree/sniSupervisor* page

If you only want to display GlobalDatas from the MIB, enter “subtree/superVisGlobalDatas” in the URL in the address field of your browser.

If you want to extract only the variables from the *VisSubagentTable*, enter “Subtree/superVisSubagentTable” or “subtree/superVisSubagentEntry” in the address field of your browser.

10.2.2.3 The row URL - selecting single table rows

To retrieve only a single row from the *superVisSubagentTable* specify the “row/superVisSubagentName”, followed by an instance. For example, to get information about the instance MIB-II (77.73.66.95.73.73 in ASCII - 6 is the length) specify “row/superVisSubagentName.6.77.73.66.95.73.73”. Normally this will not be done by entering the URL explicitly but clicking on the folder icon of the desired row. The layout is similar to the display for scalar objects.

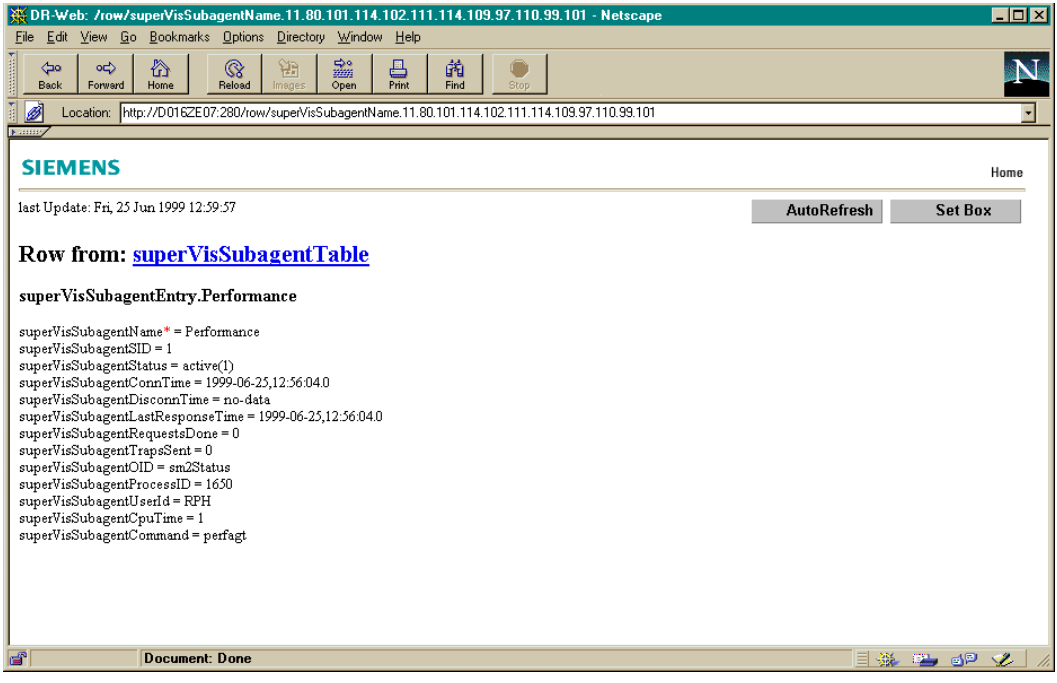


Figure 64: The row/superVisSubagentName.6.77.73.6.73.73 page

10.2.2.4 The raw-URL - representation of MIB information in “raw data” format

The DR-Web interface provides a way to access management information in a “raw data” format. The feature is intended for Java applets as a way to conveniently parse information about MIB objects. To retrieve a MIB subtree in raw data, change subtree in the URL to subtree+raw. For example, to retrieve the MIB-II subtree in “raw data” format, specify “subtree+raw/mib_2” at the browser prompt.

10.2.2.5 The refresh URL - Automatic refresh of management information

The DR-Web interface provides a way to refresh management information automatically at regular intervals. In order to perform an autorefresh in DR-Web, the DR-Web entity must send a page to the browser which contains the meta tag
<meta http-equiv=“Refresh” content=60>.

To active the AutoRefresh facility, either

– replace the string “subtree” in the URL with “subtree+refresh” or

– click on the *AutoRefresh* button located in the upper right-hand corner of most DR-Web pages.

The default interval is set to 60 seconds. The refresh time can be defined individually for each custom page by setting “RefreshTime = *value*” within the <body> tag attributes.

10.2.2.6 The set URL - SetRequest functionality

Besides retrieving MIB values, it is also possible to perform SetRequests at the DR-Web interface and in this way change the values of MIB variables (MIB objects), which are displayed in the browser.

To display a DR-Web page in the browser with fields for modifying the MIB variables by either:

– replace the string “subtree” in the URL with “subtree+refresh” or

– click on the *AutoRefresh* button located in the upper right-hand corner of most DR-Web pages.

The current value of a modifiable variable is displayed in the input field. Specify whether the value is to be changed in the checkbox to the right of the input field. Alternatively, you can assign a button or a pull-down menu to these MIB variables followed by a checkbox in each case.

The following requirements must be satisfied in order to be able to change the value of an MIB variable:

– the MIB object is defined as *read-write* or *read-create* in the MIB and implemented as *read-write* or *read-create* in the agent.

– You have write access to the MIB object.

The current security configuration (see page 28) is considered during generation of the web page.

Proceed as follows for each MIB object that you want to change at your browser:

1. Enter the desired value.
2. Click the associated checkbox.

When you have completed these steps for all MIB objects, click on the Set button, which is positioned below the MIB variables.

Setting scalar variables

The DR-Web page shown below enables you to change several objects in the system group of MIB-II:

– The MIB objects *sysContact.0*, *sysName.0* and *sys.Location.0* have *read-write* status. The associated values are shown in the input fields (*sysName*, an MIB object has the status *read-write* but write access has been prohibited in the current implementations).

– Five objects are *read-only* and therefore cannot be changed.

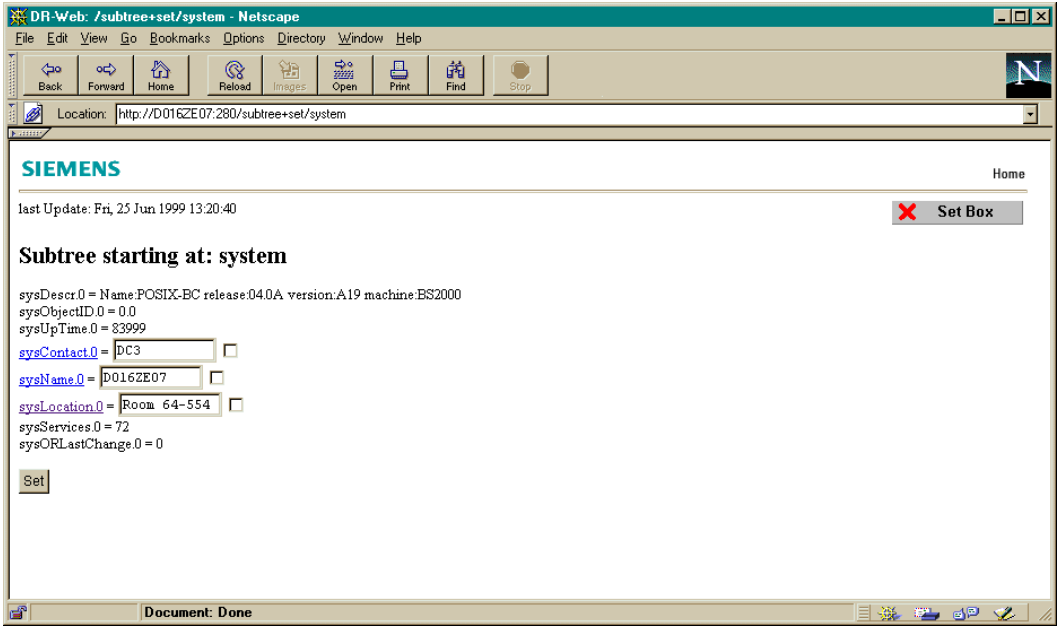


Figure 65: Subtree+set/system page

Setting table variables

To change a value in a table row, either:

– click on the folder icon for the relevant page or
– enter a URL containing “row+set/*table index*”, followed by an instance name.

Just as with scalar variables, each settable variable appears inside an input field. Settable variables can also be assigned a button or pull-down menu, followed by a checkbox.

To change a MIB object, proceed as for scalar objects:

1. Enter the desired value.
2. Click the associated checkbox.

When you have completed these steps for all MIB objects, click on the Set button, which is positioned below the MIB variables.

Creating a table instance

When the *subtree+set* URL is used with a table, the DR-Web page generated differs from a subtree page only if the table in the MIB supports entity generation via the RowStatus TEXTUAL-CONVENTIONS.

To create a new row in the table, proceed as follows:

1. click on the folder icon beside the text *New Row* or enter a URL containing “row+set/*table*” Note that no instance information should follow the URL.
2. For each MIB object to be changed, which is to be displayed on the web page, carry out the following steps:
   – Initialize the associated MIB variable or change the default value.
   – Activate the associated checkbox.
3. Click on one of the buttons *Create and Go* or *Create and Wait*.

10.2.3 Custom Page functionality

If you want to user the HTML subagent on your system, click on the *Custom* hyperlink of the DR-Web welcome page (see page 402) to access the custom page functionality. This allows you to use your pre-configured web pages or create custom web pages, which besides the components, text, graphics, etc. also include macros for accessing individual MIB objects. Furthermore, you can group information according to your personal viewpoint. The section entitled section “Configuring a custom page” on page 419 describes how to generate custom pages.

To access a custom page, enter the URL after the keyword “custom” in the network address.

10.2.3.1 Preconfigured Custom Pages

Preconfigured custom pages are offered for the following tasks:

– SNMP management
– Network management
– System management
– Application management

The preconfigured custom pages can be used as work examples. Each step contains a timestamp, the most important information form the system group of MIB-II and the following task-specific information:

- SNMP management
  - SNMP parameters
  - SNMP security information
  - SNMP web configuration
- Network management
  - ICMP statistics
  - Interface table
  - Routing tables and routing information
- System management
  - System resources
  - Graphic display of CPU values
- Application management
  - Subsystems
  - User and BCAM applications

A user ID id defined for each task area. This ID has read and write access only to the MIB variables that belong to the task area.

Task area	**ID**	**Password**
SNMP management	snmpAdmin	admin
Network management	netAdmin	admin
System management	systemAdmin	admin
Application management	applicationAdmin	admin

10.2.3.2 DR-Web menu page

To access the menu page of the web agent, either

– click on the *custom* hyperlink ion the welcome page of the web agent or

– enter the URL “http://*netzadresse*:280/custom” (e.g. http://D016ZE07:280/custom) in the address field on the welcome page.

Figure 66 on the next page shows an example of a menu page.

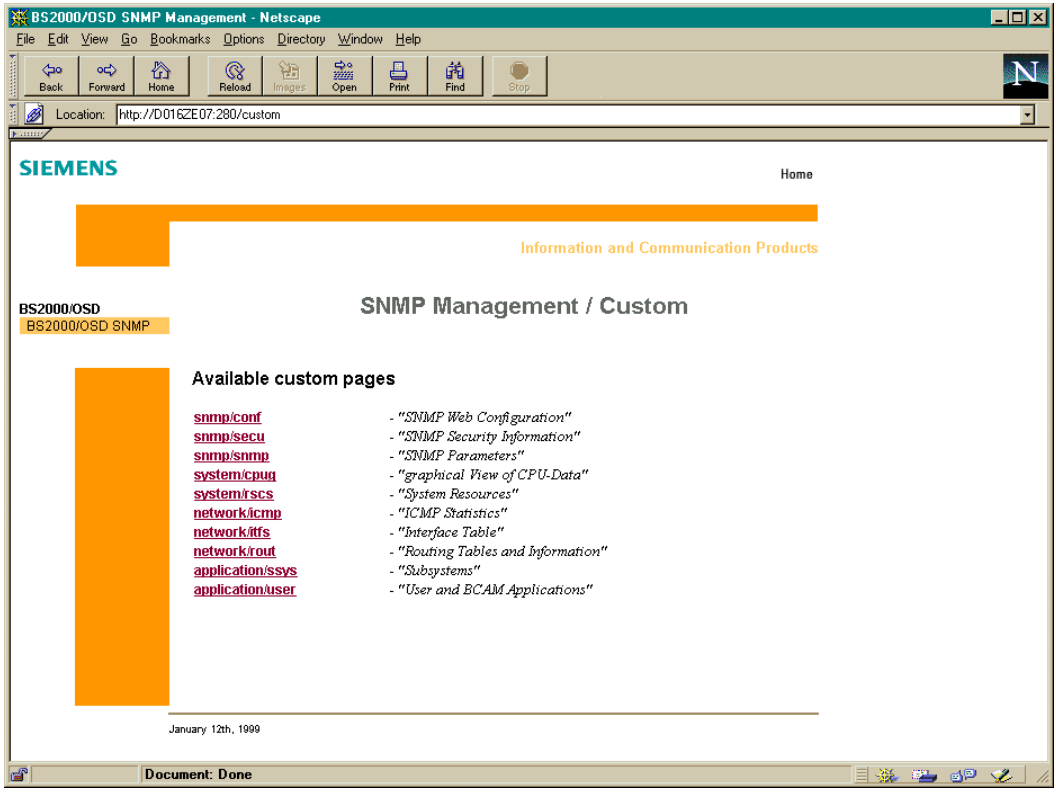


Figure 66: Menu page of the web agent

You can click on the relevant hyperlinks to display the custom pages supported by the web agent at your browser. If, for example, you want to see the page *SNMP parameters*, click on the *snmp/snmp* hyperlink. Entering a suitable URL that ends with “custom/snmp/snmp” ending in the address field of your browser produces the same results. The custom page *SNMP parameters* is shown in figure 67.

If you create your own web pages, these are automatically added to the list of customer pages. Note that you cannot create your own web pages unless you are use the HTML agent.

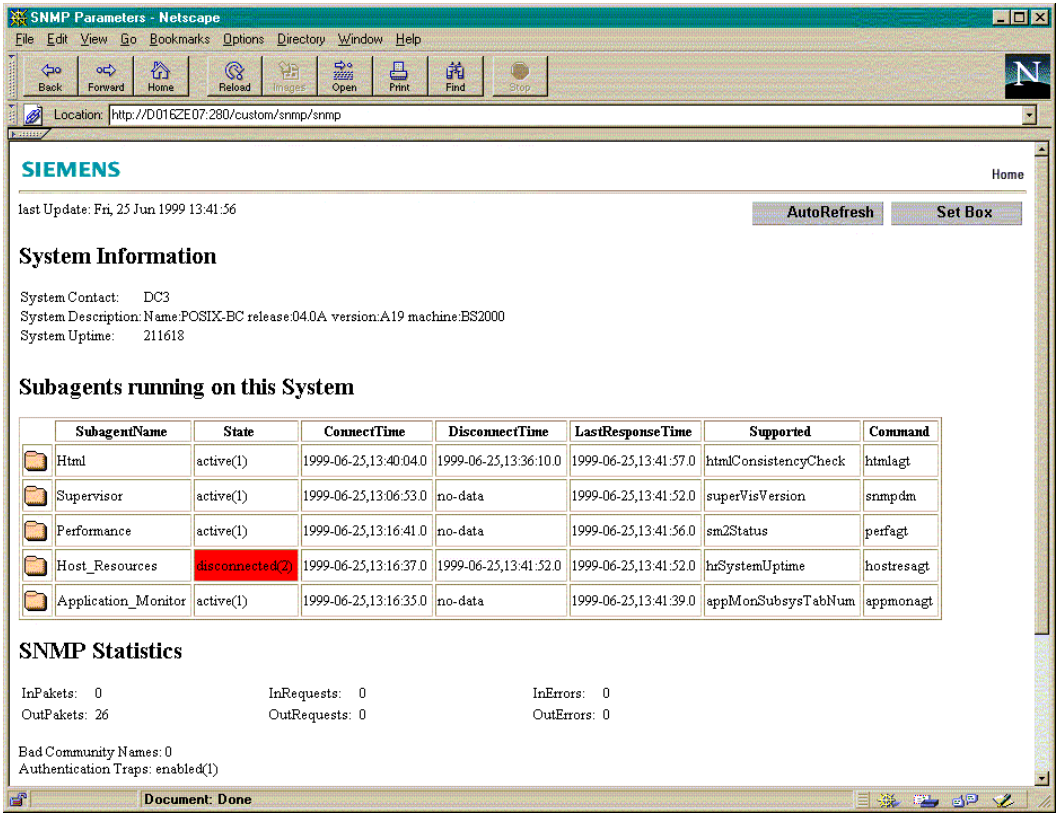


Figure 67: Custom page *SNMP parameters*

10.2.3.3 Parameterizing the custom page

Custom pages can be configured so that user entries are required. In this case, you are requested to enter parameters. This is the case with the *Interface Overview* page, for example. You access this page when you enter the associated URL ending with “custom/interface/overview” at your browser prompt. To display the information via a particular interface, the web agent must which interface you wish to use. The web agent determines the interface by sending a page to your browser to poll the input parameters and interface number. An typical parameter page is shown below.

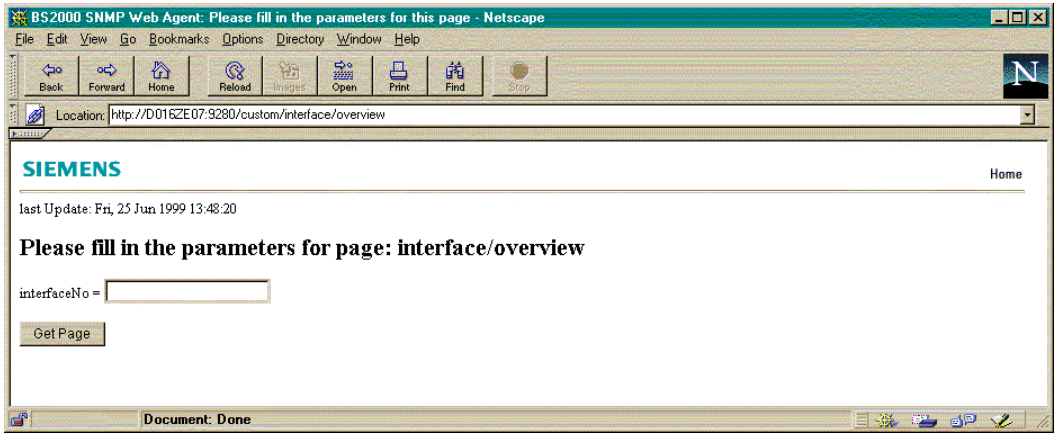


Figure 68: Entering parameters for the *interface/overview* custom page

Type the requested information into the widget text and click on the *Get Page* button. If you enter the value “1” as interface number on a particular system, the custom page shown on the following page is displayed on your browser

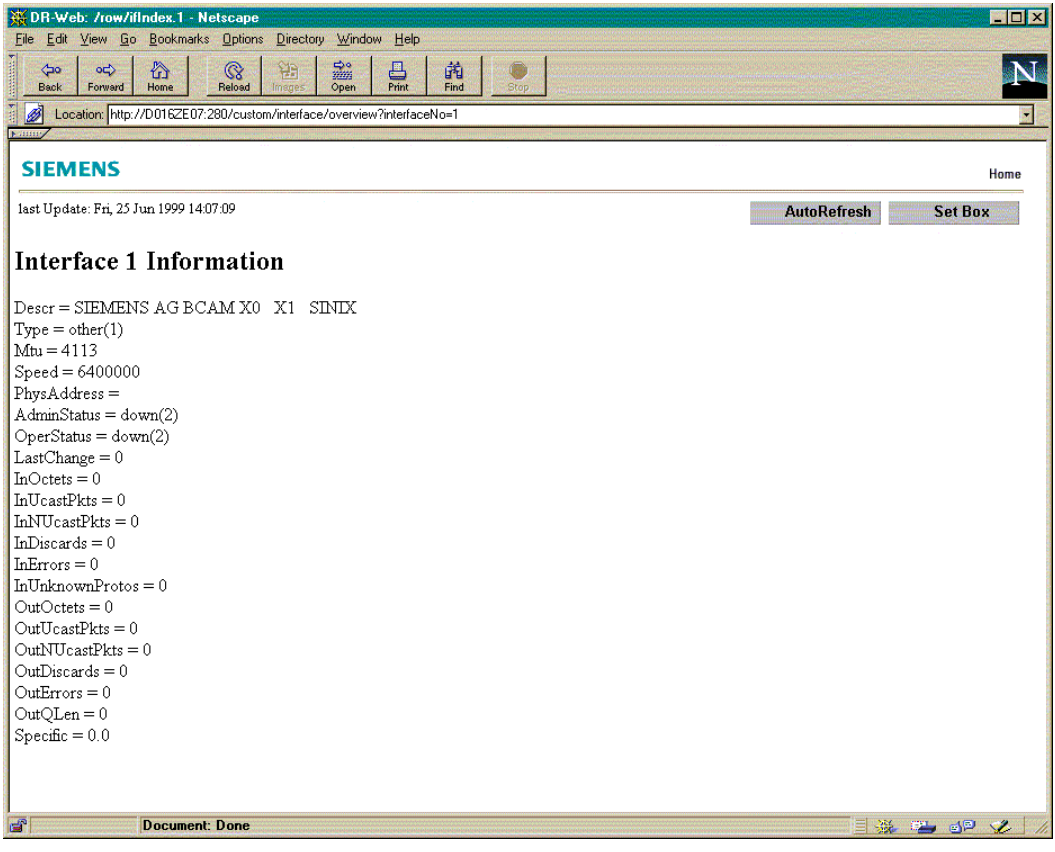


Figure 69: The *interface/overview* custom page: interface No = 1

10.2.4 Trap display in the web browser

You can use the DR web interface to call up a web page that displays the 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.

Requirements on the management station

The following software must be available on the management station:

– Browser with Java plug-in capability, e.g. Netscape Communicator ≥ V4.7

– Java plug-in x-java-applet ≥ V1.2.2

  If this plug-in is not available on the management station, the system displays a WWW address under which it can be obtained.

Requirement on the SNMP agent

The IP address of the individual management stations must be configured as a trap destination with port 9162 on the SNMP agent whose traps are to be displayed in the web browser. The port can be modified, but must be non-privileged (i.e. > 1024) and must correspond to the PORT parameter in the *trap.html* file.

The following files and directories are loaded by the SNMP agent for displaying traps in the browser:

/etc/snmp/dr-web/doc/root-a.htm	HTML page containing the link to *trap.html*
/etc/snmp/dr-web/doc/trap.html	HTML page containing the Java applet
/etc/snmp/dr-web/doc/img/Snmp.jar	Library of the necessary Java classes

Applet parameters

The following overview shows the parameters that can be used to configure the applet in *trap.html*.

Name	Value	Meaning
PORT	> 1014	Trap receiver (see files and directories on the agent) Default: 9162
MAXMESS	_	Maximum number of traps displayed Default: 100
SHOWMESS	_	Number of traps displayed (determines the size of the table)
WIDTH / HEIGHT		Width and height of the applet
TRACE	true / false	Trace display enabled/disabled
DEBUG	true / false	Debug display enabled/disabled

Editing the trap table

The following options are available for editing the trap table:

– Double-click a selected table line to display this line completely in a separate window.

– Press the DELETE key to delete selected table lines.

– You can also rearrange the columns, change the column width, and scroll through the table.

Error messages

The following Java exceptions can occur:

SnmpException	Port possibly occupied
IllegalArgumentException	Invalid port (e.g. port number < 0)
SecurityException	Port in privileged area (port number < 1024)

10.2.5 Using the web agent as a web server

The web agent can function as a general web server (general web server facility). In this case, you specify a URL containing the keyword “doc” after the network address, followed by “/” and the name of the document to obtain access to a simple HTML document. In this way, you can enter “doc/custom.html” to access the page *custom.html* where the online documentation for custom pages is stored, for example. The hyperlink *Help_is_available_here* on the welcome page of the web agent is a link to URL “doc/custom.html”.

If you do not specify the name of a simple HTML document in the URL, the web agent searches for the file *index.html*.

10.2.6 Customizing the DR-Web interface

The the subagent’s welcome page, the subtree page and the custlist page are only examples of web pages and can be customized conveniently.

The HTTP engine performs a special mapping of the root URL, subtree URL and custom page URL:

– Specifying the root URL (“/”) is the same as specifying the URL “doc/root-a.html”.

– Specifying the subtree URL is the same as specifying the URL “doc/subtree.html”.

– Specifying the custom URL is the same as specifying the URL “doc/custlist.html”.

To change the welcome page or the subtree page, therefore, one simply needs to modify the corresponding file which is served by the general web server facility. The custlist page must contain the keyword ****CUSTOMTABLE****. This key will be replaced by the list of custom pages.

10.3 Configuring a custom page

This section describes,

– how to use MIB objects and parameters to create a custom page based on a standard HTML document,

– how to configure the custom page in the HTML MIB by incorporating, meta information for the custom page in the tables of the HTML MIB.

10.3.1 Creating the custom page

The steps below are required to create a custom page and are explained using an example. Custom pages can be used to display the values of MIB objects in a variety of layouts, for example. To generate the custom page, the developer should be able to build on a web document that already possesses the required layout, including graphics, text, Java Applets, Java scripts, etc. This initial document can be produced manually or using any HTML editor.

Initial HTML document

Figure 70 shows the layout of an initial HTML document. This layout already contains the labels for the desired MIB information; the values themselves are not displayed, however.

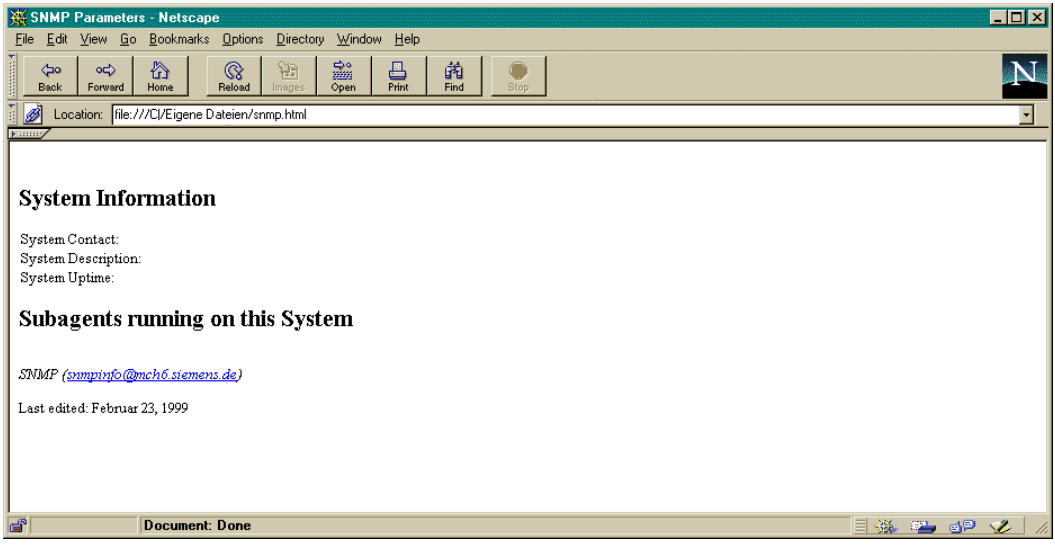


Figure 70: Layout of an initial document without MIB values

The layout shown in figure 70 contains only simple text passages without graphics. The associated initial HTML documents look s like this:

```
<html>
<head>
       <title> SNMP Parameters </title>
</head>
<body>
       <h2> System Information </h2>
            System Contact:<br>
            System Description:<br>
            System Uptime:<br><p>
       <h2>Subagents running on this System</h2>
       <address>
            SNMP(<a href=mailto:snmpinfo@mch6.siemens.de>
                              snmpinfo@mch6.siemens.de</a>)
       </address><p>
       Last edited: February 23rd, 1999
</body>
</html>
```

Adding MIB objects to the HTML initial document

The custom page shown in figure 70 requires three values of the system group of MIB-II and the subagent table of the supervisor MIB to be displayed. To convert a normal web page to a custom page, add a
`<mibobj> ... </mibobj>` tag to the HTML document in each place where the name and value of an MIB object are to be displayed. To display only the value of the MIB object (without the name of the MIB), add the attribute “value” to the tag.

For example, to get the string “sysContact.0” followed by the current value of sysContact.0, add `<mibobj>sysContact.0</mibobj>` to the custom page. However, if you only want to see the current value of sysContact.0, add `<mibobj value> sysContact.0</mibobj>` to the HTML page.

MIB tables may also be included in the custom page. An entire MIB table is added to the HTML page in the same way as the scalar variable `<mibobj> ... </mibobj>`. Use the *columns* attribute to select which columns of an MIB table are to be displayed.

For example, consider the *superVisSubagentTable* in the supervisor MIB. To include the entire MIB table, the custom page developer could add `<mibobj> superVisSubagentTable </mibobj>` to the web page. To display in this table only the objects *superVisSubagentName*, *superVisSubagentStatus*, *superVisSubagentConnTime* and *superVisSubagentLastResponseTime*, the custom page developer should add the following tag to the HTML page:

```
<mibobj coumns='superVisSubagentName, superVisSubagentStatus,
superVisSubagentConnTime, superVisSubagentResponseTime'</mibobj>
```

The complete example is shown on page 423.

Using Tag Attributes in the HTML document

- **name-value**

  name-value is the default value and displays *name* = *value* pairs.

  Example:

  `<mibobj name-value>sysContact.0</mibobj>` shows “sysContact.0 = HelpDesk” where “HelpDesc” is the current value of the MIB object *sysContact.0* here.

- **value**

  value displays only values

  Example:

  `<mibobj value>sysContact.0</mibobj>` shows “HelpDesk”.

- **columns = '<column> [,<column>] ...'**

  In a table, only the columns specified in *list-of-columns* are displayed.

  Example

  The following tag shows all lines in the *SuperVisSubagentTable* table, but only the columns *superVisSubagentName* and *superVisSubagentStatus*:

  ```
  <mibobj columns= 'superVisSubagentName,superVisSubagentStatus'>
  superVisSubagentTable</mibobj>
  ```

- **columns = <column:title> [,<column:title>] ...**

  Only the columns with the specified titles are shown in a table.

  Example:

  To show all lines in the table *SuperVisSubagentTable* burt only the columns *superVisSubagentName* and *superVisSubagentStatus* with the titles *SubagentName* or *Status*, you must specify the following in the HTML page:

  ```
  <mibobj columns= 'superVisSubagentName:SubagentName,
  superVisSubagentStatus:Status'>superVisSubagentTable</mibobj>.
  ```

- **rowselect=<column>,<color>:<value> [,<color>:<value>] ...**
  with the values: **a | a-b | a- | -b**

  In a table, change the color of the column specified in <column> (attribute) to *color* if the attribute value is "value".

  Examples

  1. ```
     <mibobj rowselect='superVisSubagentStatus,#FFFF00:3,#FF0000:2'>
     superVisSubagentTable</mibobj>
     ```

     produces the following result for the attribute *superVisSubagentStatus*:

     – a yellow field for the value 3 (undefined)
     – a red field for the value 2 (disconnected)

  2. ```
     <mibobj rowselect='sm2TimeIOMachTabIdleTime,#FFFF00:200-500,
     #FF0000:-200'>sm2TimeIOMachTab</mibobj>
     ```

     produces the following result for the attribute *sm2TimeIOMachTabIdleTime*:

     – a yellow field if 200 < *idletime* < 500
     – a red field if *idletime* < 200

Using parameters in the HTML document

Some HTML documents require input parameters. Configured properly, the web agent will look for the parameter values in the URL and forward them to the custom page. To reference a parameter, specify its name *parametername* as follows:

```
&$parametername;
```

Example

To transfer the value of the parameter *interfaceNo* to the custom page, add "&$interfaceNo;" in the HTML text editor of the customer page.

Custom page ready to be configured

The additions described create a custom page on the basis of the initial HTML page, which can be used to modify the configuration of the web agent.

The complete HTML code for the custom page is as follows:

```
<html>
<head>
        <title> SNMP Parameters </title>
</head>
<body>
        <h2>System Information </h2>
              System Contact: <mibobj value>sysContact.0</mibobj><br>
              System Description: <mibobj value>sysDescr.0</mibobj><br>
              System Uptime: <mibobj value>sysUpTime.0</mibobj><br><p>
        <h2>Subagents running on this System</h2>
              <mibobj rowselect='superVisSubagentStatus,#FFFF00:3,#FF0000:2'
              columns='superVisSubagentName,superVisSubagentStatus,
              superVisSubagentConnTime,superVisSubagentDisconnTime,
              superVisSubagentLastResponseTime,superVisSubagentCommand
              superVisSubagentOID'>
              superVisSubagentTable</mibobj><p>
        <address>
              SNMP (<a href=mailto:snmpinfo@mch6.siemens.de>
                        snmpinfo@mch6.siemens.de</a>)
        </address><p>
        Last edited: Februar23, 1999
</body>
</html>
```

Note that a web browser which does not have the DR-Web extensions will ignore <mibobj> tags. The web agent, on the other hand, replaced these tags with MIB values and sends the modified page to the browser.

10.3.2 Configuring the custom page using HTML-MIB

Once the custom page has been completed, you can configure it in the HTML-MIB. This section first provides an overview of how to configure a custom page. It then explains three options for carrying out the configuration.

10.3.2.1 Configuring the HTML-MIB tables

The following tables of the HTML-MIB must be supplied with information about the custom page:

– htmlPageTable
– htmlPageParameterTable
– htmlPageContentTable

Configuring the *htmlPageTable*

The *htmlPageTable* of the HTML-MIB contains information about the properties of the custom page, for example the name and title. An entry in the *htmlPageTable* means that the custom page exists. The *htmlPageTable* also contains information found in the page header, starting at the beginning of the custom page and going through the invocation of the <body> tag, including any <body> tag attributes:

```
<html>
       <head>
              <title> SNMP Parameters </title>
       </head>
<body>
```

The *htmlPageTable* also contains information that is normally found in the footer of the custom page: the contact address between <address> ... </address> and the data last changed.

```
       <address>
          SNMP (<a href=mailto:snmpinfo@mch6.siemens.de>
                                              snmpinfo@mch6.siemens.de</a>)
       </address><p>
       Last edited: Februar23, 1999
</body>
</html>
```

When configuring the web agent, it is important to note that the HTML generator of the web agent will use the object values in the *htmlPageTable* to generate the HTML code for the browser. Many tags are generated, so they should not be included in the value of the MIB objects. For example, the value of *htmlPageTitle* should not contain the strings <title> and </title>. Also, *htmlPageBodyArgs* should only contain the attributes of the <body> tags, not the <body> itself.

Configuring the *htmlPageParameterTable*

The *htmlPageParameterTable* contains information about parameters which are referenced by the custom page, if any. Typically, the value of parameters are not needed (or known) until the user of the web browser attempts to access the custom page. If a default value for a parameter defined in the *htmlPageParameterTable* table is not given, the user is prompted by the browser. For this purpose, the parameter name and an input field with a delimiter is displayed at the browser.

Configuring the *htmlPageContentTable*

The *htmlPageContentTable* contains information about the <body> of the custom page, everything between the <body> and </body>, except for the contact address (<address>...</address>) and the data of the last update. Again, note that no row of the *htmlPageContentTable* may contains the strings “<body>” or “</body>”.

```
<h2> System Information </h2>
       System Contact: <mibobj value>sysContact.0</mibobj><br>
       System Description: <mibobj value>sysDescr.0</mibobj><br>
       System Uptime: <mibobj value>sysUpTime.0</mibobj><br><p>
<h2>Subagents running on this System</h2>
       <mibobj rowselect='superVisSubagentStatus,#FFFF00:3,#FF0000:2'
       columns='superVisSubagentName,superVisSubagentStatus,
       superVisSubagentConnTime,superVisSubagentDisconnTime,
       superVisSubagentLastResponseTime,superVisSubagentCommand
       superVisSubagentOID'>
        superVisSubagentTable</mibobj><p>
```

10.3.2.2 Configuring the custom page in a configuration file

By far the simplest way to configure the custom pages is to edit the HTML files on which the DR-Web pages are based.

If you want to configure a new custom page with the name *custompage* directly in the configuration file of the web agent, proceed as follows:

1. Add the following lines to the configuration file:
   - one *htmlPageEntry* line
   - zero, one or more *htmlPageParameterEntry* lines as needed
   - one or more *htmlPageContentEntry* lines

   Entries in the DR-Web configuration file correspond to the MIB objects defined in the DR-Web HTML-MIB (see page 199).
2. Create a file with the name of the custom page in the directory */etc/snmp/dr-web/pages/snmp* and add the suffix ".cnf" (*snmp.cnf* in the examples).
3. Make the new configuration file known to the web agent by adding the name of the custom page to the file */etc/snmp/dr-web/pages/pagelist*.

The structure and meaning for the individual entries is explained below.

htmlPageEntry - defining the properties of the custom page

To define the properties of the custom page, add a line to the DR-Web configuration file with the tag <htmlPageEntry>.

The format of the VALUE clause is:

```
htmlPageName htmlPageTitle htmlPageAddressInfo htmlPageLastUpdated
htmlPageBodyArgs
```

where:

- *htmlPageName* specifies the name of the custom page
- *htmlPageTitle* specifies the title which will be displayed at the top of the custom page (e.g. SNMP parameters).
- *htmlPageAddressInfo* contains the contact information to be displayed on the custom page. This entry may contain HTML text, e.g. a hyperlink to sending e-mail:

  ```
  "SNMP(<ahref=mailto:snmpinfo@mch6.siemens.de>snmpinfo@mch6.siemens.de
  </a>)"
  ```
- *htmlPageLastUpdated* is a character string containing the data when the custom page was last updated (e.g. `"Last edited: Februar23, 1999"`)

– *htmlPageBodyArgs* is a list of attributes to be used in the <body> tag (e.g. “#bgcolor=’EFEFEF’”).

Note that all the fields of this VALUE clause except *htmlPageName* may have a zero-length octet string as a value. A zero-length octet string is represented by a dash (-) in the configuration file.

Example

The following entry in a custom page configuration file demonstrates how to configure the properties for the web page shown in

```
htmlPageEntry  \
snmp/snmp \
"SNMP Parameters" \
"SNMP
   (<ahref=mailto:snmpinfo@mch6.siemens.de>snmpinfo@mch6.siemens.de</a>)"\
"Last edited: Februar23, 1999" - -
```

htmlPageParameterEntry - Defining custom page parameters

To define the parameters of a custom page, add a line to the configuration file of the web agent with the tag <htmlPageParameterEntry>.

The format of the VALUE clause is:

```
htmlPageParameterName htmlPageParameterDefault htmlPageName
```

where:

– *htmlPageParameterName* specifies the name of the parameter which is referenced within the HTML text of the custom page.

– *htmlPageParameterDefault* is the value which will be returned by an invocation of the parameter. IF the parameter has no default value (e.g. zero-length octet string), the user of the web browser is prompted for the value of the parameter when the web page is accessed. A zero-length octet string is represented by a dash (-) in the configuration file.

– *htmlPageName* is the name of a *htmlPageEntry* which identifies the custom page with which this parameter is associated.

htmlPageContentEntry - Defining the contents of a custom page

To define the contents of a custom page, add a line to the DR-Web configuration file with the tag <htmlPageParameterEntry>.

The format of the VALUE clause is:

```
htmlPageContentIndex htmlPageContentText htmlPageName
```

where:

– *htmlPageContentIndex* is any whole number for this line of HTML text.

– *htmlPageContentText* is a line of text comprising the content of the custom page. The *htmlPageContentText* entry may contain valid HTML text, references to parameter and `<mibobj> ... </mibobj>` elements.

– *htmlPageName* is the name of a *htmlPageEntry* which identifies the custom page with which this HTML text is associated.

Example

The following lines from a custom page configuration file demonstrate how to configure the contents for the sample web page in figure 63 on page 405. Note that the VALUE clauses shown here do not have a line break in the middle of the quoted strings ("): the lines are merely folded to fit.

```
htmlPageContentEntry \
1 " <h2> System Information </h2>" snmp/snmp
htmlPageContentEntry \
2 " System Contact: <mibobj value>sysContact.0</mibobj><br>" snmp/snmp
htmlPageContentEntry \
3 " System Description: <mibobj value>sysDescr.0</mibobj><br>" snmp/snmp
htmlPageContentEntry \
4 " System Uptime: <mibobj value>sysUpTime.0</mibobj><br><p>" snmp/snmp
htmlPageContentEntry \
10 " <h2> Subagents running on this System </h2>" snmp/snmp
htmlPageContentEntry \
11 " <mibobj rowselect='superVisSubagentStatus #FFFF00:3,#FF0000:2'
columns='superVisSubagentName,superVisSubagentStatus,superVisSubagentConnTime,
superVisSubagentDisconnTime,superVisSubagentLastResponseTime,superVisSubagentCommand
superVisSubagentOID'>
superVisSubagentTable</mibobj><p>" snmp/snmp
```

10.3.2.3 Configuring the custom page using SNMP requests

Since all the information concerning the custom page configuration parameters are stored in the HTML-MIB, you can use SetRequest statements to create or modify the custom-pages. The HTML-MIB is described on page 199.

10.3.2.4 Configuring the custom page using the DR-Web interface

Custom pages can also be configured using the DR-Web interface. To do this proceed as follows:

1. Define the properties of the custom page:

   Access “subtree+set/htmlPages” and create a new row in the *htmlPageTable* of the HTML-MIB.

2. Define the parameters of the custom page:

   Access “subtree+set/htmlPageParameterTable” and create zero, one or more lines in the *htmlPageParameterTable* of the HTML-MIB.

3. Specify the contents of the custom page:

   Access “subtree+set/htmlPageContentTable” and create one or more lines in the *htmlPageContentTable* of the HTML-MIB.

Entries in the DR-Web configuration file correspond to the MIB objects defined in the HTML-MIB (see page 199).

10.4 DR-Web user configuration

To access the information provided on the web agent, the user must enter his or her user name and password at the web browser (see page 401). This information is serves to implement a user configuration which determines

– the information to which the user has read access,
– the information to which the user has write access.

The DR-Web user configuration is described on page 28 in the section “Security configuration”.

11 Trap server for Solaris and Reliant UNIX

The trap server is a simple daemon process in Solaris and Reliant UNIX which receives traps and forwards them to configured ports on the local host or remote hosts.

The trap server thus fulfils the following tasks:

– It multiplies and distributes traps.

– It enables programs that do not have root authorization to receive traps.

The *trpcmd* command program can be used to control the trap server locally or remotely (see page 435).

The trap server is installed with *pkgadd* in accordance with the package procedure. It is a component of the product BS2-SNMP-SO or BS2-SNMP-SX, located on the supplied CD.

The CD contains the following:

Package	**Meaning**
trpsrv from BS2-SNMP-SO	Trap server for Solaris
SMAWtrpsv from BS2-SNMP-SX	Trap server for Reliant UNIX

11.1 Files and directories

During the installation of the trap server, a directory called *trpsrv* containing the following programs is created under */opt/lib/emanate*:

- *trpsrv* (server program)
- *trpcmd* (command program)
- *trpsnd* (trap send program)
- *trpmsg* (trap send program for the special BS2 console format)
- *trprcv* (trap receive program)

The start/stop procedure *ptrpsrv* for the server process is stored in the */etc/init.d* directory. The *rc* procedures *S90trpsrv* (*rc2.d*) and *K10trpsrv* (*rc0.d* and *rcS.d*) merely represent links to the *ptrprsv* file.

11.2 Environment variables

The trap server and associated programs use the environment variables described below.

Environment variables of *trpsrv* and *trpcmd*

TRPSRVPORT	Receive port
TRPSRVCNFDAT	Target configuration file (full path name)
TRPSRVCOMPORT	Communication port between the server and command program
TRPSRVTGSPORT	Start port for dynamic port assignment
TRPSRVTGRANGE	Range for dynamic port assignment

Environment variables of *trpsnd* and *trpmsg*

TRPSNDPORT	Sender port
TRPSNDADDR	Send address
TRPSRCADDR	Sender address

Environment variable of *trprcv*

TRPSRVPORT	Receive port

11.3 Trap server process *trpsrv* (daemon process)

The installation of the trap server is configured such that the server process is started following the installation and with each system start.

Starting the server program

The server program is started with the command:

```
trpsrv [-p <port>][-l][-t {c|e}]
```

-p <*port*>
Specifies the trap receive port.

-l
Only local connections of a command program are permitted.

-t
Specifies the trace level.

The following trace levels are available:

– c
Trap receive and trap distribution can be traced.

– e
Messages from the ERROR class are output.

Incoming trap port

The traps sent by the SNMP agent are accepted by the trap server process at the incoming trap port. The trap server determines the incoming trap port as follows:

1. If a port was specified using the *-p* switch, this port is used as the incoming trap port. Otherwise, the trap server continues with step 2).
2. If the environment variable TRPSRVPORT is set, TRPSRVPORT is evaluated and the port specified here is used as the incoming trap port. Otherwise, the trap server continues with step 3).
3. The *snmp-trap* service in the */etc/services* file is evaluated and the incoming trap port is selected accordingly. If this is not possible, port 162 is used as the incoming trap port.

Distributing traps

The trap server process distributes the received traps without modification to the configured receive ports.

The receive ports can be defined as follows:

– with an entry in a target configuration file
– with the *trpcmd* command program (see page 435)

Defining receive ports in the target configuration file

The *trpsrvtargets* file stored in the */opt/lib/emanate/trpsrv* directory is used as the default target configuration file. If you want to use a different file as the target configuration file, notify the trap server using the TRPSRVCNFDAT environment variable by specifying the desired target configuration file with its full path name.

The entries in the target configuration file that can be used to specify trap destinations (receive ports) are structured as follows:

<*port*> [<*system*>]

port
Number of the port on the specified <*system*> system to which the traps are to be sent.

system
IP address of the system on which the receive port *port* is located.
Default: IP address of the local system.

Receive ports configured using the *trpsrvtargets* file or another target configuration file cannot be deleted using the *trpcmd* command program.

11.4 Command program *trpcmd*

The command program can be used to configure the trap server.

Communicating with the trap server process

The *trpcmd* command program communicates with the trap server via a TCP connection, whereby port 5410 is used by default. You can use the environment variable TRPSRVCOMPORT to set a different port. In this case, you must restart the command program and trap server in the modified environment.

By setting the appropriate parameters, the server process can also be accessed remotely with the command program, provided the trap server does not reject remote instructions (see “Starting the server program”, -I switch, on page 433).

The command for configuring the trap server is structured as follows:

```
trpcmd [-s <server>] {-a <port>[/<system>] | -n | -r | -d <port>[/<system>] | i } [-t {c | e}]
```

-s <*server*>
: The command program is directed at the trap server on the <*server*> system. Specify an IP address for <*server*>.

-a <*port*>[/<*system*>]
: Adds a new trap destination. Specify the port number for <*port*> and specify the IP address of the new trap destination for <*system*>. If you do not specify a value for <*system*>, the local system is assumed.

-n
: Adds a new trap destination, whereby the system assigns the associated port number dynamically. By default, port assignment begins with port 16000 and covers a range of 50 ports. These values can be changed using the environment variables TRPSRVTGSPORT and TRPSRVTGRANGE.

  Dynamic port assignment is only possible on the local system. The number of the port selected by the system is output to *stdout* by the command program.

Example

```
% trpcmd -n
16001
```

-d <*port*>[/<*system*>]
Deletes the specified trap destination from the distribution process. It is only possible to delete trap destinations if the respective port was included in the distribution with the *-a* or *-n* switch using the command program.

-r
Deletes all trap destinations. The *trpsrvtargets* file with the configured receive ports (trap destinations) is reloaded.

-i
Outputs trap distribution information to standard output.

Example

```
% trpcmd -i
Receive port: 9999
_____________________________________

No.     Type     Port     Address

_____________________________________

000     PERM     08822    127.0.0.1
001     PERM     05566    139.25.105.176
002     PERM     00162    139.25.104.105
```

-t
Sets the trace level on the server.

The following trace levels are available:

– c
Trap receive and distribution can be traced.

– e
Messages from the ERROR class are output.

Results of the command program

The command program returns the following return codes:

0	OK
1	general error
2	not found
3	already exists
4	socket-create failed
5	check failed
6	maxclient reached
7	maxrange reached

11.5 Trap send program *trpsnd*

The trap send program *trpsnd* can be used to send a trap in the general format.

Calling *trpsnd*:

`trpsnd <switch> ...`

<switch>	Meaning	Default value
-d	destination address	mandatory parameter, no default!
-p	destination port	162
-a	sender address	local address
-c	community	public
-g	generic trap number	0
-s	specific trap number	-
-u	time ticks	-
-o	object list	-

11.6 Trap send program *trpmsg*

The trap send program *trpmsg* can be used to send a trap in Application Monitor-specific format.

Calling *trpmsg*:

```
trpmsg <switch> ...
```

<switch>	Meaning	Default- value
-d	destination address	mandatory parameter, no default!
-p <*port*>	destination port	162
-a	sender address	local address
-s	source => BS2-<*source*>	BS2Console
-o	object	-
-w	weight	0
-m	message	noMessage

11.7 Trap receive program *trprcv*

The trap receive program *trprcv* is used to receive a trap.

Calling *trprcv*:

```
trprcv [-p <port>] [-t]
```

-p <port>
Incoming trap port
Default: 162

-t
Activates the trace

12 Configuration examples

A management station offers the administrator three information levels:

- central monitoring of the functional integrity of all components in the network,
- selective information on parameters, capacity utilization and statistical values,
- control through manual or automatic intervention in the system based on the information obtained.

Thus, it is possible to integrate BS2000/OSD systems into the comprehensive options for information and alarm management in modern management stations.

This chapter focuses on an explanation of the four monitoring examples:

1. basic monitoring
2. monitoring the console
3. monitoring enterprise-critical applications
4. monitoring system performance.

The examples are related but do not have to be configured in a single operation. This means that, after installation, you will be able to perform basic monitoring of your system after performing a few tasks.

There are two basic ways of monitoring the functioning and state of the central components in a network:

– polling and
– traps.

When polling is performed, the management station queries the state of the systems being monitored at regular intervals. The manager assumes the active role in the communication and controls the activities. The advantage of polling is its reliability: should an agent, the manger or the system fail temporarily, this has no effect on the result once communication has been restored. Important is the definition of an appropriate polling rate, which should represent a reasonable compromise between the network load and the amount of time information is delayed.

A trap is an asynchronous message sent by an agent which relates to a problem state. The agent assumes the active role in communication whereby performance is of foremost importance. The network load is reduced to a minimum and the message is unsolicited.

Often a combination of the two is used. A trap to the management station activates polling or reduces a high polling rate.

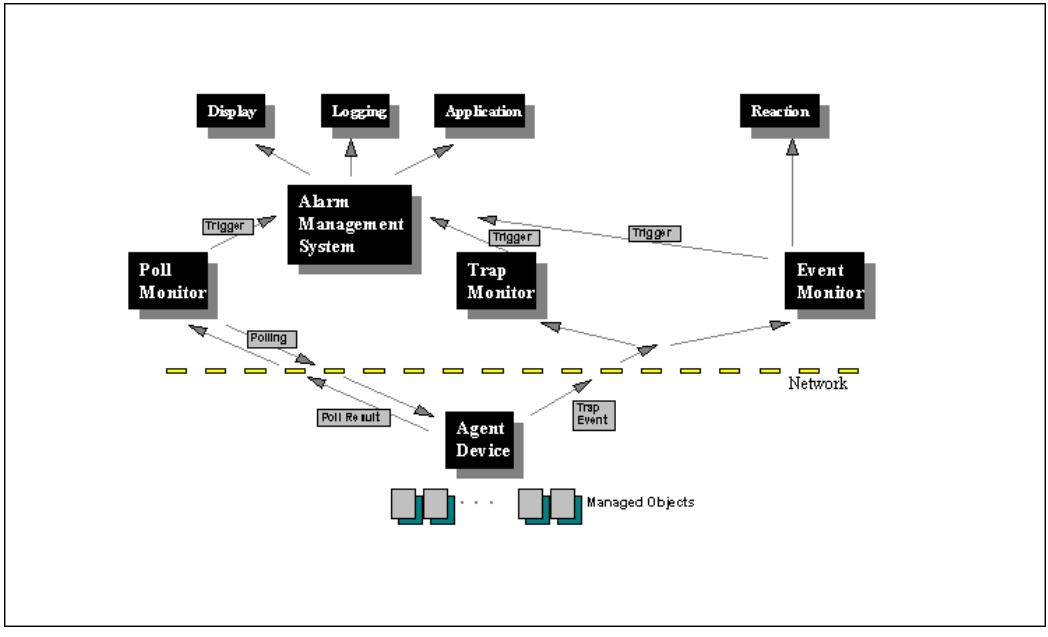


Figure 71: Monitoring with polling and traps

12.1 Basic monitoring

Task

If, in future, you always want to be informed of whether the SNMP agent on your system is functioning properly, you must configure your management system in such a way that an object belonging to the central system group (e.g. *sysDescr.0* above) is queried at regular intervals. Depending on the features provided by your management system, you can set up an visual or acoustic error indicator which is triggered if no response is received.

Configuration

Carry out the following configuration steps at the management station.

1. Configuring the network map (TransView):

You can create icons for the BS2000/OSD system in the network map. Enter the name, IP address of the system and the Community according to the agent configuration file *snmpd.cnf*. Set the device to *managed*.

TransView device overview:

1.	Name:	<systemname>
2.	Address:	<ip address of the BS2000/OSD systems>
3.	Community:	master
4.	managed:	managed

2. Defining an alarm (TransView)

A *Status* alarm is already defined in TransView. Check that this alarm is activated.

Alarm:

Alarm name Status check ON

3. Defining a poll (TransView)

Use the *Sys_poll* provided. This poll queries an object in the system group at the subagent at three minute intervals (default setting). If no answer is received, an alarm is initiated. Make sure that the poll is activated.

Poll:

Poll name Sys_poll check ON

Change the property of the poll from *system* to *system-mib-II*.

Result

Now stop the master agent in BS2000 using the command:

```
/STOP-SNMP-MASTER
```

Within the next three minutes (polling rate), TransView will send a request to the agent and will not receive a response. Because of this *NoResponse* information, the icon representing your BS2000/OSD system in the network map will turn yellow, thus indicating an error state.

Then restart the master agent with the command:

```
/START-SNMP-MASTER
```

After three minutes at the latest, the alarm will be reset.

If you think three minutes is too long, you can reduce the polling rate for *Sys_poll*. This does, however, increase the network load.

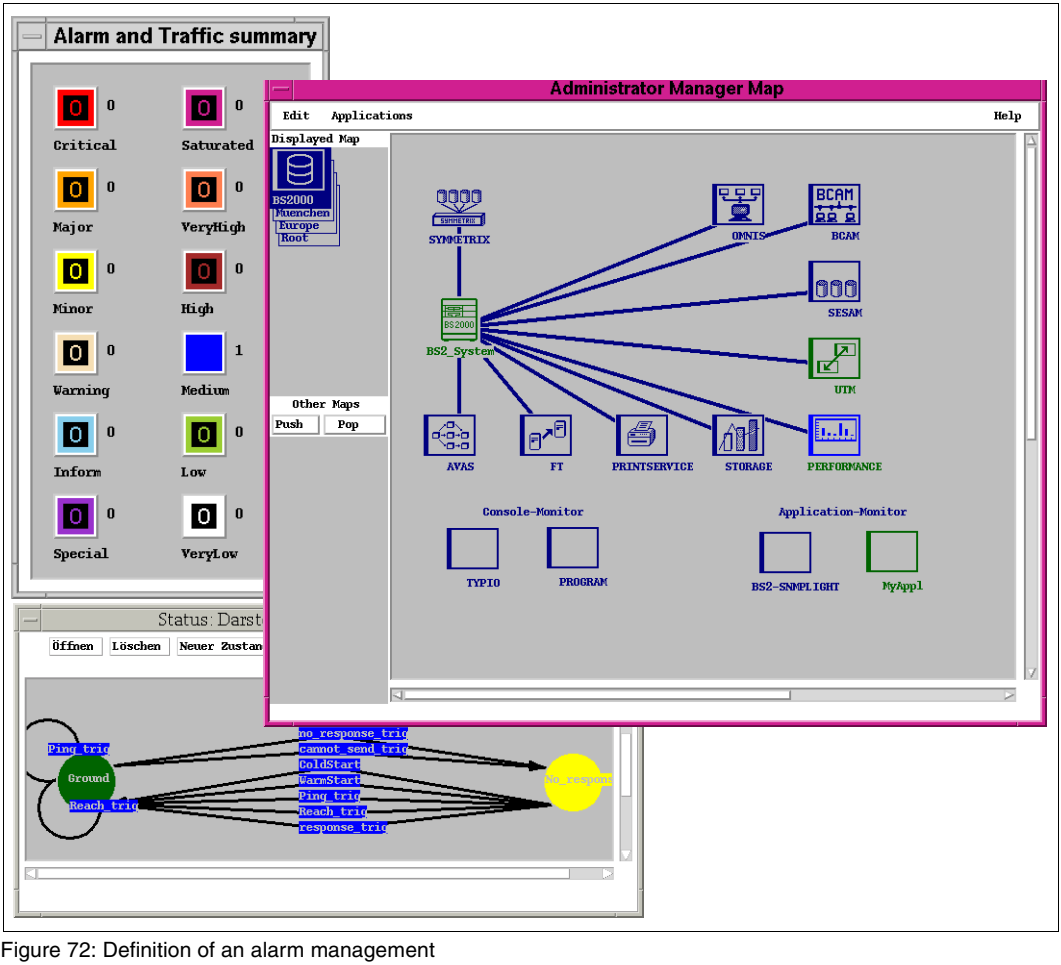


Figure 72: Definition of an alarm management

12.2 Monitoring messages with the Console Monitor subagent

Task

You want the following three *open*UTM events to be reported to the management station:

– termination of an application
– loss of a connection
– violation of security measures.

*open*UTM is represented by an icon in the TransView network map. The various event classes are to be represented by the different colors the *open*UTM icon can turn.

Configuration

Carry out configuration steps 1 and 2 in BS2000/OSD, and steps 3 to 6 at the management station.

1. *open*UTM applications:

Message destinations are defined in the *open*UTM connection module and can be modified with the utility routine KDCMOD. The modified message module must be linked with the subprograms of an application. See the UTM manuals for more information.

In this way, the following UTM messages are output to the console:

Event	Message
End of application / end of task	K056, K058, K059, K060
Loss of a connection	K032, K036, K069
Violation of security measures	K004, K005, K006, K031

You can vary the selection of the messages.

2. Console Monitor subagent

The filter file of the Console Monitors must b expanded to include the following entry:

```
< UTM0100 [weight] SOURCE=UTM DEVICE=UTM >
```

If *open*UTM outputs one of the above-mentioned messages to the console, a trap is generated by the Console Monitor subagent, in which the source is supplied with *BS2-UTM* and the object and Community are supplied with *UTM*.

3. Configure the network map (TransView)

Add an icon with the name *UTM*, the IP address of your BS2000/OSD computer and the Community *UTM* to the network map. Since *Device name* and *Community* are identical, the application is detected to be an “Object without own IP address” by TV-CC. Add the *UTM* object to the list of *Systems in the domain* and update the TransView Control Center.

TransView device overview:

1. Name: UTM
2. Address: <ip address of the BS2000/OSD system>
3. Community: UTM
4. managed: managed

Then update TV-CC

4. Define an application (TransView)

An application with the name *BS2-UTM* must be defined in the *Integrated Applications* window. No action is associated with this application.

5. Define an event (TransView)

Three events must be defined. The events correspond to the required message groups and are characterized by different alarm levels.

Event	Name	Alarm level	Pattern list
End of application/task	utmTermEv	Serious	.*K056.*, .*K058.*, .*K059.*, .*K060.*
Loss of connection	utmConnEv	Inform	.*K032.*, .*K036.*, .*K069.*
Security protection violation	utmAccEv	Special	.*K004.*, .*K005.*, .*K006.*, .*K031.*

6. Define the associations (TransView)

Use the *Integrated Application → Activate events and reactions* menu to connect node, application, events and, if necessary, reactions. Specify the *UTM* object as node, *BS2-UTM* as the application and the events defined in section 5 above.

Node:	UTM
Application:	BS2-UTM
Event:	utmTermEv
Event:	utmConnEv
Event:	utmAccEv

Result

The subagent forwards all messages written to the console with message number *UTM0100* to the management station. This trap information contains *UTM* as the object and *BS2-UTM* as the source. TransView has three events associated with this source and this object. The event triggered depends only on the UTM error number Kxxx contained in the actual message. The defined alarm is initiated in this way. The *open*UTM icon changes color according to the alarm level.

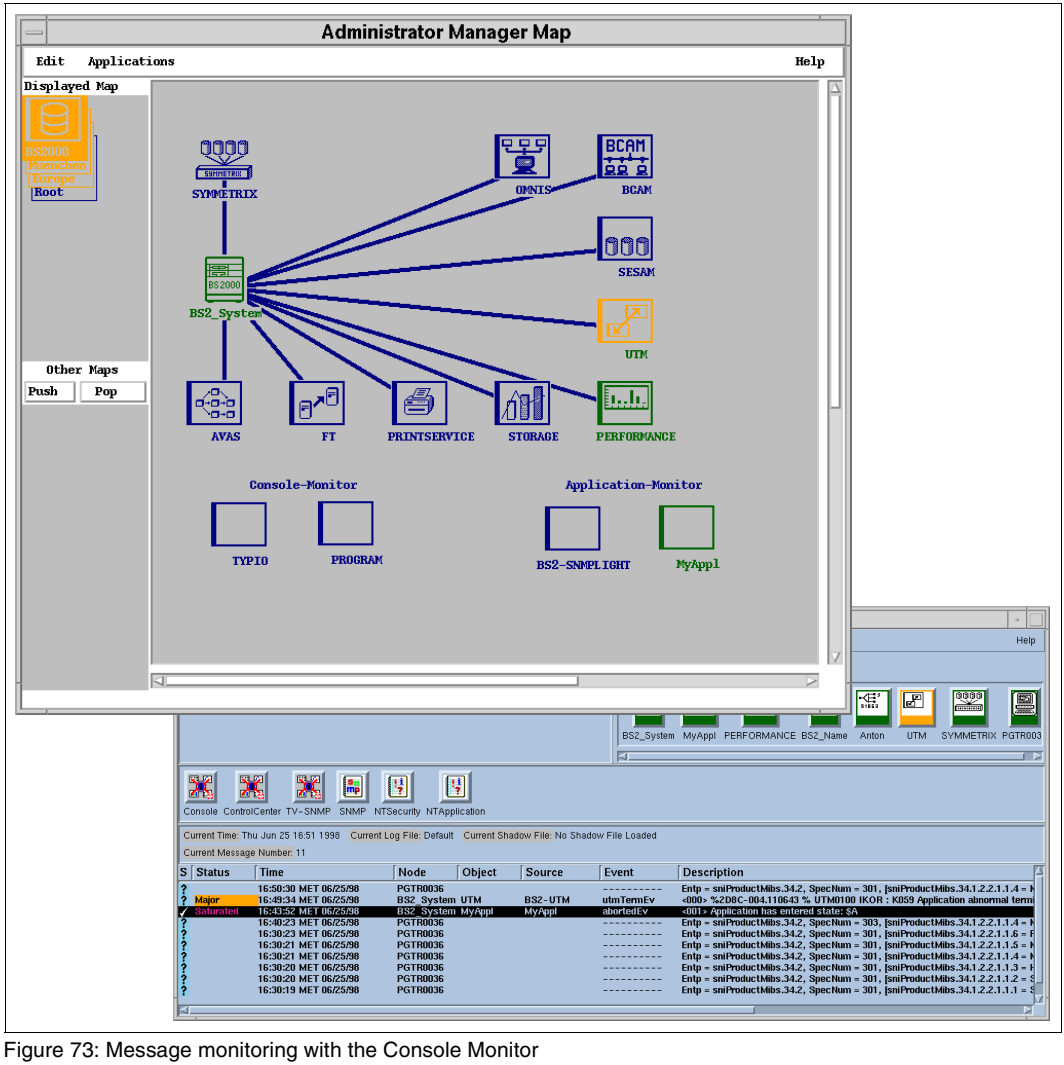


Figure 73: Message monitoring with the Console Monitor

12.3 Monitoring applications with the Application Monitor subagent

Task

You want to monitor certain applications which are especially important to you so that you are informed of any failures immediately. To do this, you add the relevant applications as icons to the TransView network map. If an application terminates abnormally, the corresponding icon is to change color. This allows you to deal with problematic applications quickly and efficiently. In addition, the event manager displays a message informing you of the name of the application and the system on which the application is running. You can select the alarm level according to the importance of the application. The alarm is reset when the application is restarted.

Configuration

Carry out the first configuration step in BS2000/OSD and steps 2 to 5 at the management station.

1. Application Monitor Subagent

Modify the ADD-APPLICATION-RECORD entry in the configuration file of the Application Monitor.

a) Enter the name of you application at APPLICATION-NAME. The entry TYPE=*USER indicates that the application is a user application.

b) For JV-NAME, enter the name of the monitor job variables, which you use to start the application.

c) The trap conditions *A* and *R* mean that changes of state in MONJV to $A (aborted) and $R (running) are signalled.

d) ICON = *YES means that a separate object with the name of the application is expected for this application in the network map of the management station.

```
//&ADD-APPLICATION-RECORD  -
APPLICATION-NAME = <application>, -
TYPE = *USER, -
JV-NAME = <jv name>, -
TRAP-CONDITION=(A,R), -
ICON = *YES
```

Start the agent with the new configuration file.

The subagent subscribes each change in the specified MONJV to the specified states and forwards the status changes to the management station as a trap, provided that the application was started with MONJV.

2. Configure the network map (TransView)

Add an icon with the name of the application to be monitored to the network map. The address of the BS2000/OSD system on which the application runs is entered in the overview window of this device. *Device name* and *Community* must be identical to the application name. This means that the application is recognized by the TV-CC as “Object without its own IP address”. Update the TransView Control Center as in the previous example.

TransView device overview:

1.	Name:	<application>
2.	Address:	<ip address of the BS2000/OSD system>
3.	Community:	<application>
4.	Group:	tcc
5.	managed:	managed

Then update the TV-CC

3. Define an application (TransView)

Define an application with the name of the object to be monitored in the *Integrated applications* window. No action is linked to this application.

4. Defining an event (TransView)

Two events must be defined. The first is triggered when MONJV signals a change of state to ‘$A’ (aborted); the second, when the state changes to ‘$R’ (running). The second event resets the first one.

Event	Name	Alarm level	Pattern list
User application running	runningEv	Normal	R$
User application aborted	abortedEv	Full	A$

5. Define the links (TransView)

Nodes, applications and events must be linked via the *Integrated applications → Activate events and reactions* window. Now, specify the object in the network map defined under 2 as application, that defined under 3 as event and the one defined under 4 as events.

Node:	<application>
Application:	<application>
Event:	runningEv
Event:	abortedEv

Result

The subagent forwards all changes in its MONJV to the management station. This trap information notes the name of the application as source. You have linked two events using the application icon and this source. The *abortedEv* event is triggered if MONJV signals a change to $A. The *High* alarm level defined is displayed by a change in color of the application icon. The *runningEv* event occurs when MONJV assumes the value $R and resets the previous event. If you specify ICON=*YES, the *Object* parameter is supplied the application name. TransView now forwards the alarm to this object (without own IP address) in the network map.

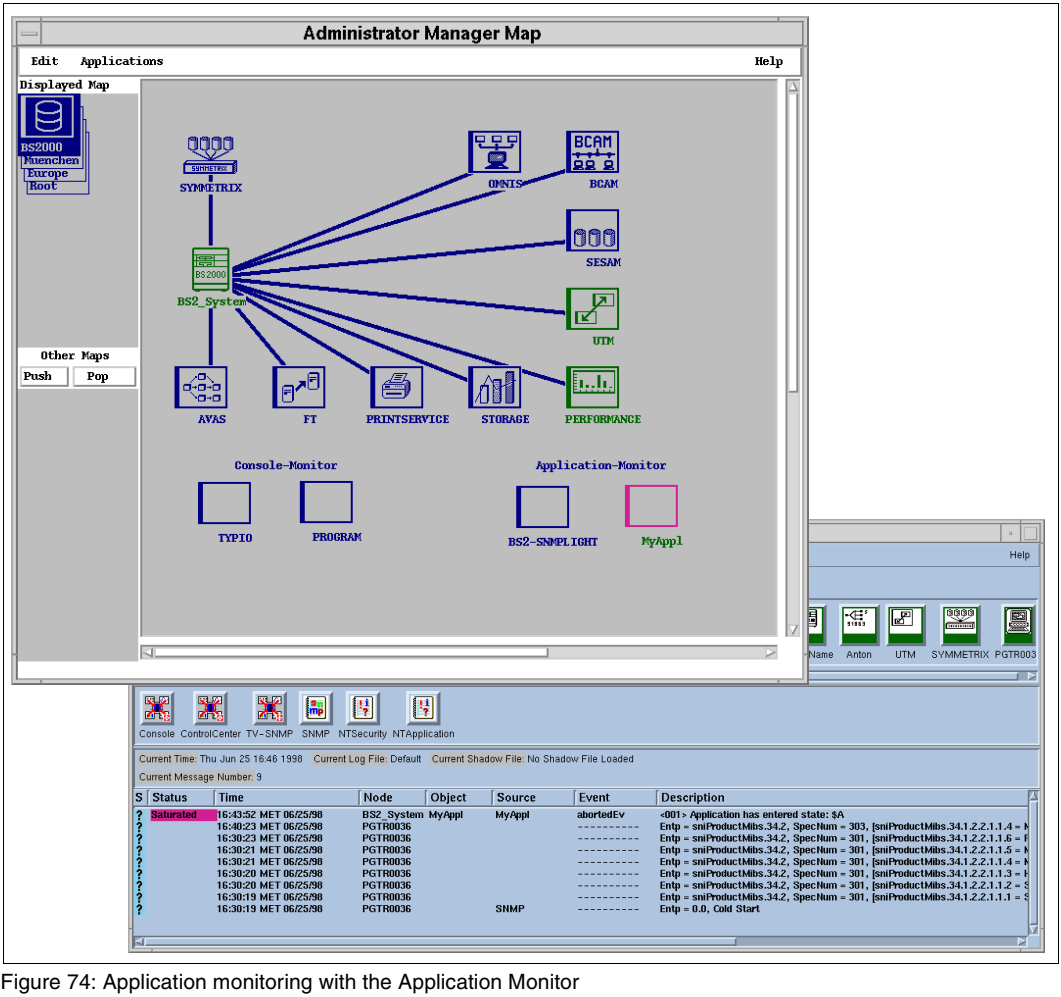


Figure 74: Application monitoring with the Application Monitor

12.4 Monitoring the systems using the Performance Monitor subagent

Task

You want to display the CPU load in different colors for the performance icons in the network map. Light green indicates a low load, blue and medium load and brown a high load. These colors correspond to the alarm levels in TransView SNMP.

Configuration

No special configuration is required in the Performance Monitor subagent. You should therefore carry out all the steps below at the management station.

1. Configure the network map (TransView)

You have already set up an icon for the BS2000/OSD system or are using the PERFORMANCE icon from the TV-SMBS2 installation. Set the device to *managed*.

TransView device overview:

1.	Name:	<system name> (PERFORMANCE)
2.	Address:	<IP address of the BS2000/OSD system>
3.	Community:	master (PERFORMANCE)
4.	managed:	managed

2. Define the polls (TransView)

Define the following four polls.

Poll name / trigger	Cycle	Meaning	Property
PerfStart	10 min	*sm2TimeIOMachTabEntry.* *sm2TimeIOMachTabIdleTime* exists	*sm2TimeIOMachTabEntry*
PerfGering	2 min	*sm2TimeIOMachTabEntry.* *sm2TimeIOMachTabIdleTime* ≥ 700	*sm2TimeIOMachTabEntry*
PerfMittel	2 min	*sm2TimeIOMachTabEntry.* *sm2TimeIOMachTabIdleTime* < 700 **and** *sm2TimeIOMachTabEntry.* *sm2TimeIOMachTabIdleTime* ≥ 200	*sm2TimeIOMachTabEntry*
PerfHoch	2 min	*sm2TimeIOMachTabEntry.* *sm2TimeIOMachTabIdleTime* < 200	*sm2TimeIOMachTabEntry*

The first poll is only for initializing the instances and has therefore been defined with a higher polling rate. The polling rates of the other polls corresponds to the standard cycle for SM2.

3. Define an alarm (TransView)

Define the following alarm diagram:

Alarm

Alarm name:	Performance
States:	Low, medium, high
Property:	NO_PROP
Trigger:	according to the following diagram

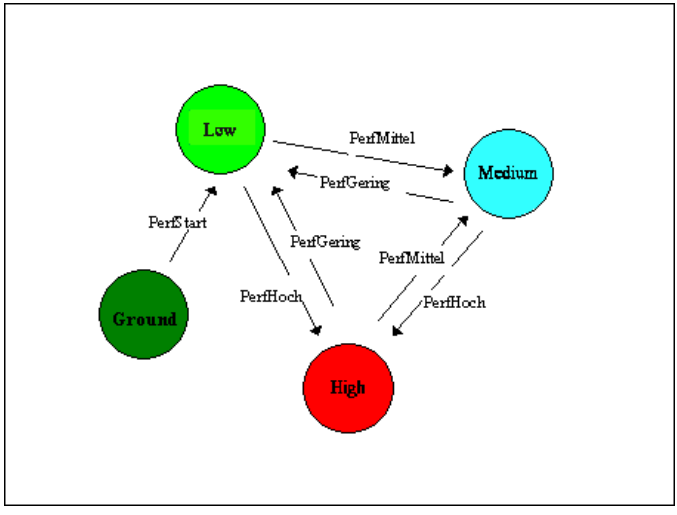


Figure 75: Alarm diagram for performance monitoring

Result

TransView polls the *IdleTime* of the system at the interval defined in the polling rate. If TransView is returned a value less than 200, for example, (the time in indicated in thousands), the *PerfHoch* trigger sets the alarm diagram of the system to *High*.

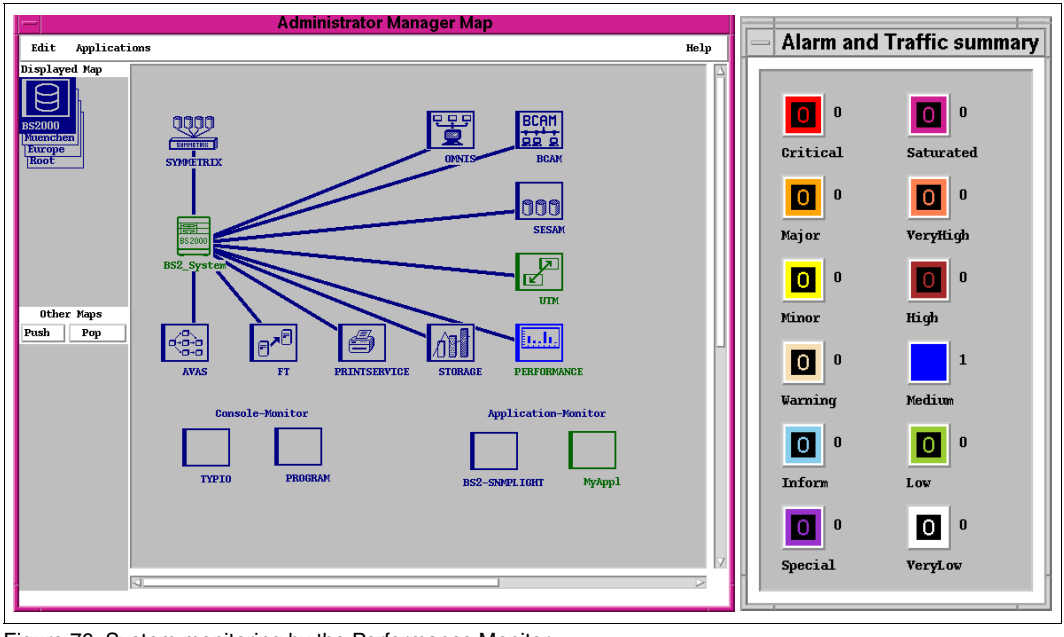


Figure 76: System monitoring by the Performance Monitor

13 Appendix: DCAM return codes

If the YOPNCON call is rejected immediately and a negative return code issued, the state is set to 'unknown'.

Exceptions:

Code	Meaning	State
0C 08	Partner already connected to the DCAM application.	running
0C 4C	Unable to reach partner; if DCAM application, the application is not open.	terminated

DCAM return codes and resulting derived states:

Macro call successful (Code: 00)

Code	Meaning	State
00	Macro call successful	running

Macro call terminated with warning (Code: 04)

Code	Meaning	State
04 0C	No request or no request with matching EDIT option for connection setup in the queue	running
04 10	Macro call terminated due to timeout (TOVAL)	running
04 18	A request in a queue for connection setup has been annulled (system) timeout	running
04 20	Connection message truncated	running
04 44	No printer ready to print out data	running

Macro call rejected due to current state of DCAM application (Code: 08)

Code	Meaning	State
08 04	DCAM application not opened by calling task (invalid AID)	unknown
08 20	Warning: forced termination of DCAM application	unknown
08 24	Forced termination of DCAM application	unknown
08 28	Forced termination of DCAM application due to a DCAM error	unknown
08 2C	Forced termination of DCAM application due to specification of an invalid contingency/event ID by the primary task	unknown
08 38	Termination of the DCAM application by a request from the primary task	unknown
08 40	Too many calls of the same type made for this task (max. 8, for YOPNCON ACQUIRE 128 permitted).	running
08 60	Too many connections for application that has not been pre-defined	running

Macro call rejected due to current state of partner (Code: 0C)

Code	Meaning	State
0C 08	Partner already connected to the DCAM application.	running
0C 0C	A request from this partner has already been place din the queue (no ACQUIRE possible).	running
0C 10	The DIP control block is not active (invalid DID).	running
0C 18	The connection has been closed down by a user request, or by a YCLSCON.	running
0C 34	The position of the distribute code exceed the maximum message length.	running
0C 40	Partner system has rejected the connection. No reason specified.	unknown
0C 44	Partner requesting incorrect protocol	running
0C 48	System timeout for connection request	running
0C 4C	Unable to reach partner; if DCAM application, this application is not open.	terminated
0C 50	Partner not processing any request to close down connection (DCAM application in STOP state.)	terminated
0C 54	Partner refuses to accept requests to set up connection (DCAM application has attributes ATTR=NLOGON.)	running

Code	Meaning	State
0C 58	Invalid password (LOGPW)	running
0C 5C	Request to set up connection rejected by partner (e.g. REJLOG-macro call). Only for DCAM(NEA) transport service applications.	running
0C 60	Partner characteristics not accepted by partner.	running
0C 64	Error in station service protocol	running
0C 68	Partner refuses to process request to close down connection; request to close down connection follows from partner.	running
0C 6C	Error while activating VTSU support for partner	running
0C 70	Partner belongs to a different DCAM application.	running
0C 74	DIP control block does not address a DCG control block	running
0C 78	Connection closed down by partner or system immediately after setup.	running
0C 90	Proposed transport service class not accepted by partner.	running
0C 94	Proposed data network priority not accepted by partner.	running
0C 98	Error in processing (e.g. fault in X.25)	running
0C 9C	Request to close down connection rejected by management. In this case, contact the system administrator.	running

Macro call rejected due to current status of the communication access system (DCM) (Code: 10).

Code	Meaning	State
10 04	DCAM: lack of space	terminated
10 08	Warning: terminating DCM	terminated
10 0C	Terminating DCM	terminated
10 10	DCM is not active	terminated
10 14	DCM error	terminated

Invalid use of macro call (Code: 14)

Code	Meaning	State
14 04	Macro call cannot be output to a secondary task.	unknown
14 0C	Unable to use macro call in conjunction with DCAM applications that have the attribute ATTR=NLOGON.	unknown
14 10	Equivalent macro calls already pending (OPTCD=(ACQUIRE,ASY) or OPTCD=(ACCEPT,SPEC,ASY))	unknown
14 14	DCAM application not authorized to issue this macro call.	unknown
14 18	Synchronous call already entered in queue for these tasks OPTCD=(SYN,Q).	unknown

Macro call rejected due to incorrect operands (Code: 18).

Code	Meaning	State
18 04	Invalid ACB control block address	unknown
18 08	Invalid CCB control block address	unknown
18 0C	Invalid DCG control block address	unknown
18 10	Invalid DIP control block address	unknown
18 20	Invalid PTNNAME address	unknown
18 28	Invalid address for event code	unknown
18 2C	Invalid AREA address	unknown
18 3C	Invalid partner name	unknown
18 40	Processor not active (no /BCACT) or not generated, etc.	unknown
18 44	CCB control block seized for an asynchronous macro call (active CCB)	unknown

Code	Meaning	State
18 54	Invalid PRONAME address	unknown
18 58	Negative AREALN	unknown
18 5C	Incorrigible error affecting message processing	unknown
18 60	Invalid distribute code length (CODELN)	unknown
18 7C	Inconsistency between ROUTL and ROUTN	unknown
18 80	Invalid ROUTL address	unknown

Macro call rejected due to incorrect addressing or wrong register (Code: 20).

Code	Meaning	State
20	Macro call rejected due to incorrect addressing or wrong register.	unknown

Call not executed because DCAM subsystem is either not loaded or is in HOLD/DELETE state and the affected task cannot yet issued DCAM calls (Code: 24).

Code	Meaning	State
24	Call not executed because DCAM subsystem either not loaded or in HOLD/DELETE state and the affected task cannot yet issued DCAM calls.	unknown

Glossary

agent

The agent is also known as the management agent. It is the implementation of a management protocol that exchanges management information with a management station. An agent is a program that runs on a device and reports the current information about the device to a manager or corresponding management application.

alarm

A group of states and state transitions. The states correspond to entities of an object type with attribute values that are specified by the network administrator. Whenever the monitored object type of a device or line changes into a state that the administrator has defined as an alarm state, TransView SNMP reports the event by displaying a corresponding icon and changing the color of the alarm and device icons.

alarm action

A state transition occurs when an alarm receives a trigger signal. The network administrator can configure an alarm such that an action is triggered when a specific state transition occurs. This can, for example, be one of the following actions: logging the transition information, sending the *mail* or *beep* actions, sending a paging device signal, sending a trap, execution of a shell command or an application that has been called.

alarm instance

An alarm instance is assigned to a management device or the entities of a management device object type.

attribute

An attribute is part of an object type definition in an MIB module. It defines one characteristic in an object type. If an object type contains more than one instance, the attributes define the columns and the entities the rows in the object type table. The table entries are the instance values for the attribute.
See also *object type* and *object instance*.

characteristic

This can be either an object type or a characteristic string. Both can belong to one characteristic group. A characteristic string is a characteristic (but not an object type) that is added to a characteristic group by the manufacturer or network administrator for the purpose of restricting the scope of polls and alarms. A characteristic string defines the menu and submenus that are available in the device overview under the *objects* button.

It also defines the applications that are available in the device view under the *applications* button.

community string

A simple password that is shown in the network map when a device icon is added. The agent that runs on the device requires this password from the manager before information about the device is made available.

connection

The object instance that describes a (line) connection to a network management device.

connection instance

An object instance of a connection to a device. See *object instance*.

Both ends of a TransView SNMP line icon can be assigned to one device. This connection has two aspects. Firstly, it is a graphical representation of part of the physical network; secondly, it is an object type of the device (e.g. an object type for connection or junction information).

device

A network system, router, hub or other addressable equipment within the network, but not a line, tap or network map icon.

device icon

A TransView SNMP icon that represents a system, router, bridge, hub or other manageable equipment within the network (can be polled). TransView SNMP contains a number of icons that represent the different device types. Device icons can also be user-defined.

event report

Event reports indicate errors, state changes and similar important events in the system. They occur asynchronously (spontaneously), are command-independent, object-oriented and are delivered to an arbitrary network management station within the network, according to the requester principle.

gateway

A gateway connects heterogeneous networks.

HTML
HTML (HyperText Markup Language) is a standardized markup language that consists of a subset of the SGML Standard (Standard Generalized Markup Language). HTML documents can be exchanged between any computer systems using the standardized HTTP communications protocol.

HTTP
HTTP (HyperText Transfer Protocol) is the communications protocol used between systems in the World Wide Web (WWW). HTTP enables documents to be exchanged between any computer systems / applications.

Internet
The name used for a large number of interconnected networks that use the Internet protocol.

IP address
Representation of a connection point in the Internet (32 bits).

line
A line has two aspects in TransView SNMP. Firstly, it represents a graphical icon for displaying the network topology. Secondly, it is a pair of device object entities (one instance for each end of the line). It is assumed that the ends are connected to network management devices.

major trap number
The SNMP Standard (RFC 1157) defines seven trap categories with the numbers 0 to 6. These numbers are designated as major trap numbers.

manufacturer-specific extensions
Additional management objects for a device that is supplied by a manufacturer for the agents of this device and TransView SNMP. They are also frequently known as manufacturer's MIB.

MIB
MIB stands for "Management Information Base". This designates a data model that describes the network elements to be administered with network management (managed nodes), in an abstract form. This data model consists of the formal descriptions of object types (object classes) that are constructed according to the RFC1157 conventions.

MIB-II
MIB-II is a standard MIB whose use is obligatory in the Internet. It offers an adequate data model for managing devices. MIB-II is standardized and is defined in RFC1213. It is an extension of MIB-I (RFC 1156).

network management protocol
The protocol for exchanging management information.

network management station
A system in the network on which TransView SNMP or a similar management application runs.

network map
A collection of lines and icons that are arranged into a group of interconnected network maps. The corresponding network map background maps that display a network and its subnetworks are optional.

network map file
A text file that contains the configuration information for its network: the file names of background maps for network and subnetwork maps; the file names and positions of icons for systems, routers, hubs and lines; configuration information for polls, masks and alarms; characteristic groups. This file is also called the " Map Database" or "Map_db" file.

network map icon
A TransView SNMP icon that displays a network map in a group of nested network maps. The icon is displayed in the next higher network map. Network map icons can also be user-defined.

news
TRANSDATA network management asynchronous ("spontaneous") event reports that are generated by SINIX systems with ACX, BS2000 and PDN systems.

object
In an MIB: an object type or attribute.
On the graphical interface: device, line, tap, poll, mask or alarm, or a specific instance of this.

object instance
Represents the characteristics (attribute values) of a device. The entities are managed by the device agents.
The object instance is specified by the instance identifier or index.

object identifier
A notation that designates the position of an object in an MIB tree. For example, 1.3.6.1.4.1.231.1.3.2 (iso.org.dod.internet.private.enterprise.sni.1.3.2) specifies an RM600 system. There are also MIB names for the object identifier (e.g. *cisco* for a Cisco router).

object type
A class of object entities of the same type, that is defined by a formal description. There may be one or more entities on one device for an object type. The object type is in the form of a table if more than one instance is possible for an object type on one device. Each row of this table represents one object instance and the columns the attributes of the object type.
Object class is another name for object type.

pict file
A particular type of TransView SNMP file that is designated by the *.pict* extension. This file contains information about the background maps of network maps and the user interface object instance diagrams.

ping
A protocol with which the IP levels connectivity from one IP address to another is checked.

poll
Cyclic request for information about MIB object types. The network administrator can carry out configuration.

poll cycle
The poll cycle is the parameter that defines how often TransView SNMP contacts an agent on a device in order to call up information from the MIB of this device.

protocol
A number of rules with which systems communicate with each other
See also *SNMP* and *ping*.

RFC
Request for comments. The series of documents that describe the Internet protocol and related standards.

SNMP
SNMP stands for "Simple Network Management Protocol". SNMP is a standard protocol for network management in TCP/IP networks.

state
Alarm state: an element in an alarm definition (see *alarm*).
MDC state: the *Domain Table View* window shows a code under the *state* entry. This code describes whether a local or remote Client Manager sends or receives a domain.

state transition
Alarm change of state that is activated by a trigger.

subnetwork
A physical network within an IP network.

subnetwork icon
An icon in a root or subnetwork map that represents a nested subnetwork map one level below the current network or subnetwork map.

tap
A tap represents the connection point between a device and the network in a network map. A tap can be created, configured and deleted, but it cannot be managed.

TCP/IP
TCP/IP stands for "Transmission Control Protocol/Internet Protocol", i.e. the Internet protocol. A number of rules that define how systems communicate with each other in an open (not manufacturer-bound) environment. This is normally a large communication infrastructure (Internet).

trap
Under SNMP, traps are problem reports that are sent automatically by a device agent.

trigger
A trigger is a message that is sent by the poll or mask system to the alarm system. An alarm executes a state transition when a specific trigger is received.

URL
URL (Uniform Resource Locator) is a character string which users enter in their Web browser to access a WWW document.
The URL for the WWW contains the address of the required Website and consists of the following components: protocol, computer address (host domain name or IP address), port number if required, path and file name if required, and (optionally) details of a place within the document text.

variable
Under SNMP, a variable is the result of linking an object instance name with an assigned value.

view
A window with information that is collected from a device or line.

World Wide Web (WWW)
The World Wide Web, called Web for short, is an internet service which enables users to access and publish multimedia data (text, graphics, videos, animations and audio). Documents in the World Wide Web must be in HTML format.

Related publications

Please apply to your local office for ordering manuals.

TransView SNMP manuals

TransView Control Center (UNIX)
Enterprise Management for Client/Server Environments
Manager (UNIX) and Agents (UNIX, Windows NT)

User Guide

Target group
Administrators and operators of management applications.
Contents
With the TransView Control Center you can integrate a very wide range of applications for system, application and network management. The manual describes all functions of the Control Center Manager under UNIX and of the agents under UNIX and Windows NT.

TransView SNMP (UNIX)
Configuration
Uaser Guide

Target group
– Network operators
– Network planners
– Programmers of network management applications
Contents
The manual describes how to manage a TCP/IP network using the TransView SNMP graphical user interface.

TransView SNMP (UNIX)
Programming Interface
Target group
Programmers of network management applications
Contents
For programmers with a knowledge of C and Motif, this manual describes the elements of the programming interfaces of the network management product TransView SNMP.

TRANSVIEW Extensible Agent
SINIX V5.41
User Guide

Target group
Network administrators of local networks based on TCP/IP and SINIX system administrators.
Contents
The manual describes how you can use the Extensible Agent to implement your own MIBs, modify MIB configuration files remotely, execute remote operations spontaneously via SNMP and send SNMP traps.

TransView SNMP-Proxy BS2000/PDN (SINIX)
TRANSDATA Network Management via the SNMP Manager
User Guide

Target group
The manual addresses network administrators who wish to handleTRANSDATA network management via the SNMP Manager.
Contents
The manual describes installation/configuration of the product Proxy BS2000/PDN, plus operation of the product using the SNMP Manager or the graphical user interface. It also explains how you should proceed when you expand the functional scope (creating new MIB objects).

TRANSVIEW NMC/NMA/NMAE
TRANSVIEW AutoOperator
(TRANSDATA, SINIX)
General Functions
User Guide
Target group
Users of TRANSVIEW NMC and TRANSVIEW NMA (SINIX)

Contents
TRANSVIEW NMC and TRANSVIEW NMA (SINIX) implement the manager and agent for network management in TRANSDATA networks. The manual describes the general basic functions and the automatic operator for responding to event messages.

TransView NMC/NMA/NMAE
TransView NTAC2/NTAC2E
DCAM
(TRANSDATA, SINIX, BS2000, PDN)
Messages and Halt Codes
User Guide
Target group
Operators of TransView NMC and TransView NMA/E.
Contents
The manual contains all network management messages for the operating systems BS2000, PDN and SINIX, as well as explanations of the messages output. The halt codes provide information on software halts for PDN systems.

TransView-NMA/-NMAE V1.2A, TransView-NTAC2 V7.1A, NTAC2E V5.1A (TRANSDATA, BS2000)
Network Management in BS2000
User Guide

Target group
The manual addresses network planners, network administrators, network operators, and maintenance and diagnostics engineers.
Contents
It deals with network management for BS2000 systems. It describes how these products are used, their mode of operation and their interaction with other products. The manual is task-oriented; the complete reference for all the network management commands is no longer contained here, but in the “Network Management Commands” manual.

TransView NMA/NMAE (TRANSDATA, PDN)
Network management in PDN
User Guide
Target group
PDN agent users
Contents
TransView NMA (PDN) and TransView NMAE (PDN) implement the agent functionality as defined in the TransView concept. The manual describes the entire functionality from PDN system administration through network planning to maintenance and diagnostics.

BS2000/OSD manuals

BCAM Volume 1 (BS2000/OSD)
User Guide

Target group
The BCAM manuals are intended for network planners, generators and administrators who define BS2000 systems.
Contents
Volume 1 describes BCAM itself, how it is embedded in TRANSDATA and open networks, plus generation and administrative activities.
Volume 2 contains the commands, macros and error messages required for generation and operation.

BCAM Volume 2 (BS2000/OSD)
Reference Manual

Target group
The BCAM manuals are intended for network planners, generators and administrators who define BS2000 systems.
Contents
Volume 1 describes BCAM itself, how it is embedded in TRANSDATA and open networks, plus generation and administrative activities.
Volume 2 contains the commands, macros and error messages required for generation and operation.

***open*Net Server V1.0**
SNMP Management for *open*Net Server
User Guide

Target Group
The manual is intended for people responsible for networks and systems who want to use SNMP-based network and system management.
Contents
The manual provides a detailed description of the MIBs supplied with *open*Net Server, and the installation and operation of the subagents. A separate chapter contains an in-depth description of how to operate the BCAM Manager.

***inter*Net Services V1.0** (BS2000/OSD)
Administrator's Guide

Target group
This manual is intended for network planners, generators, and administrators who with to use Internet Services in BS2000/OSD.
Contents
*inter*Net Services replaces the delivery units TCP-IP-AP and TCP-IP-SV.
The manual describes the functionality of the Internet Services BOOTP/DHCP, TFTP, DNS, FTP, LDAP, and NTP in BS2000/OSD, and also outlines how to use the FTAC interface for FTP. Installation, administration, operation, and logging and diagnostic options of the various components are also covered in the manual.

***inter*Net Services V1.0** (BS2000/OSD)
User Guide

Target group
This manual is intended for network planners, generators, and administrators, and also users who wish to use Internet Services in conjunction with BS2000/OSD.
Contents
*inter*Net Services replaces the delivery units TCP-IP-AP and TCP-IP-SV.
The manual provides an introduction to the components of *inter*Net Services. It contains a detailed description of how to use FTP, the FTAC interface for FTP, and TELNET. Network administrators require this manual as a supplement to the Administrator's Guide.

BS2000/OSD-BC
Subsystem Management (DSSM/SSCM)
User Guide

Target group
This manual addresses systems support staff and software consultants of BS2000.
Contents
The following are described: BS2000 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 V3.1A (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 (BS2000/OSD)
Job Management
User Guide

Target group
This manual is addressed to AVAS users.
Contents
The manual contains descriptions of the following: menus and statements used for job management; the creation of jobs/S procedures and nets, condition descriptions, and calendars; the coupling of AVAS with MAREN; the AVAS-QUER utility routine.

AVAS (BS2000/OSD)
for the Administrator
System Administrator Guide

Target group
This manual is addressed to AVAS administrators.
Contents
The manual describes the operations required to set up the AVAS system, frequently recurring administration tasks, the external creation of AVAS elements, the AVAS program interface, and the AVAS server interface. It also discusses the option of using AVAS in BS2000 multiprocessor mode.

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 for BS2000 V7.0**
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-OS for supporting FTAM functionality. The command interface and messages are dealt with in detail.

***open*FT for BS2000 V7.0**
Enterprise File Transfer
Installation and Administration
System Administrator Guide

Target group
This manual addresses administrators who want to use *open*FT, *open*FT-OS 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-OS. 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.

***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

The following manuals are available:

- Unicenter TNG *Getting Started*
- Unicenter TNG *Concepts Guide*
- Unicenter TNG *Administrator Guide*
- Unicenter TNG *Release Summary*
- Unicenter TNG *Managing NetWare with Unicenter TNG*

Other related 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)

Ordering RFCs

If the Requests for Comments (RFCs) referred to in the text are not included with delivery, they can be ordered in hardcopy form (copying charge) or fetched as a file from "anonymous Internet FTP" or via e-mail.

Anonymous Internet FTP:

In order to fetch an RFC via the Internet from the system *nic.ddn.mil* (IP address 192.67.67.20), please proceed as follows:

– set up an FTP connection to the system: *ftp nic.ddn.mil*
– you can now load the required documents from the directory *rfc*; a list of the available documents can be found in the file *rfc-index.txt*.

e-mail:

If you do not have Internet access but can use electronic mail, you can request an RFC in this way. The document will be sent to you in response to your *Mail* query.

To do this, send a mail to the user *service* on the system *nic.ddn.mil*:

```
mail service@nic.ddn.mil
```

In the *Subject* field enter the number of the desired RFC, e.g `Subject: RFC 1155`

Written queries concerning RFCs should be submitted to:

DDN Network Information Center
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025, U.S.A.
Telefon: 415-859-3695

e-mail: nic@nic.ddn.mil

Environmental protection

Take-back, recycling and disposal

For details on take-back and reuse of devices and consumables within Europe, contact your Siemens branch office/subsidiary or our recycling center in Paderborn:

Tel. +49 5251 8180-10
Fax. +49 5251 8180-15

Further information on environmental protection

The Siemens AG representative for environmental protection will be pleased to answer any further questions you may have concerning environmental protection.

Siemens AG
Environmental Protection
Werner von Siemens Straße 6
86159 Augsburg

Tel. +49 821 599-2999
Fax. +49 821 599-3440

Index

.map_SMBS2
    network map file 104, 107
/etc/snmp/agt/snmpd.cnf 32
/etc/srconf/agt/snmpd.cnf
    configuration file 55

A

access check
    definition 34, 39, 43, 47
access rights
    for a group of management stations 30
    selective assignment 30
access to agents
    definition 33, 43
ADD-APPLICATION-RECORD
    statement for the Application Monitor 59
ADD-DATA-BASE-RECORD
    statement for the SESAM subagent 82
ADD-DCAM-APPLICATION-RECORD
    statement for the Application Monitor 60
ADD-DISK-RECORD
    statement for the storage management subagent 86
adding
    MIB object in HTML initial document 421
ADD-JV-RECORD
    statement for the Application Monitor 64
ADD-LOG-FILE-RECORD
    statement for the Application Monitor 62
ADD-PUBSET-RECORD
    statement for the storage management subagent 85
address
    input field 349
address check
    definition 35, 41
ADD-SERVER-PARAMETER
    statement for the SESAM subagent 84
ADD-SERVER-RECORD
    statement for the SESAM subagent 83
ADD-SUBSYSTEM-RECORD
    statement for the Application Monitor 61
alarm
    OmnisMsg 343
    RDBMS_relState 342
    SMBS2 342
    SubagentStatus 342
    SubagentStatus_e 342
    SuperVisBasic 342
    Symmetrix 344
APALL 166
APERROR 166
applet parameters
    trap display in web browser 417
application
    monitoring 182, 338, 346
application management 9
Application Monitor 12
    control 189
    sample configuration 448
    start 130
    stop 131
Application Monitor subagent
    ADD-APPLICATION-RECORD 59
    ADD-DCAM-APPLICATION-RECORD 60
    ADD-JV-RECORD 64
    ADD-LOG-FILE-RECORD 62
    ADD-SUBSYSTEM-RECORD 61

Application Monitor subagent (continuation)
change the configuration file in current session 58
create configuration file 56
DEFINE-OBJECT 66
DEFINE-TRAP-FORMAT 68
MIB 180
trap 180
application monitoring
control 56
Application Monitor-specific trap
structure 181, 194
variable binding 183, 184, 186, 187, 188
appmoncmd
control file monitoring 189
appMonConfFile
change configuration file 58
area
reply to console query 357
authentication of requests 31
automatic refresh
management information 406
AVAS
display job networks 210
display network state 211
display processes 208
display structure elements 211
generation file GENPAR 137
MIB 207
subagent starting 136
subagent stopping 136
system status 209
AVAS exit 77
AVAS subagent
AVASState 342
configure 76
GENPAR 76
JVCENTRAL 77
JVPLAMZD 77
JVUPAMZD 77
RZ exit 76
AVAS-RZ exit 76
AVASState
alarm (AVAS) 342

B

basic agent
icons 338
basic monitoring
configuration 441
sample configuration 441
basic performance monitoring
MIB 275
BCAM application
monitor (ADD-APPLICATION-RECORD) 59
BCAM subagent 14
software requirements 24
start 155
stop 155, 156
BCMAP 88
BMBS2
management application 20
BS2 Symmetrix 111
BS2000 log file 187
BS2000/OSD applications
monitoring 338, 346
bs2symm.def
Symmerix 111

C

CMBS2
configuration (UNIX) 122
filter window 363
installation (UNIX) 122
installation (Windows NT) 124
management application 20, 348
setting the user interface 349
start 348
stop 348
trap confirmation window 366
command
SEND-TCC-MSG 163
START-SNMP-TRAPSEND 161
trap send command 161
command program
MAIN MENU 167
start 165
subagent 165
trap server 435

command window
display 349
menu bar 359
schematic 358
toolbar 360
work area 361
communication
between SNMP manager and agent 7
SESAM subagent / SESAM/SQL server 79
UTM subagent / UTM application 87
with trap server process 435
community
input field 349
community string
configuration (examples) 36
definition 33
communityEntry 33
configuration
Application Monitor (example) 448
AVAS subagent 76
basic monitoring 441
basic monitoring (example) 441
CMBS2 (UNIX) 122
Console Monitor 69
Console Monitor (example) 444
examples 439
of Unicenter TNG (Solaris) 103
OMNIS subagent 78
*open*UTM subagent 87
Performance Monitor (example) 452
SESAM subagent 78
SMBS2 on OpenView 117
storage management subagent 85
Symmetrix 111, 112
TransView management station 91
trap distributor (Solaris) 103
configuration file
/etc/scronf/agt/snmpd.cnf 55
create for Application Monitor 56
example (Application Monitor subagent) 57
format 56, 57
GENPAR (AVAS) 76
of the Application Monitor, change 58
SESAM statememts 80
configuration file (continuation)
snmpd.cnf 32
target 434
configuration steps, security configuration 31
configuring custom page 419
via web interface 429
with HTML-MIB 424
with SNMP requests 429
configuring Unicenter TNG (Windows NT) 96
consmon.cnf
Symmetrix installation 111
consmonagt
start Console Monitor 132
consmoncmd
stop Console Monitor 132
consMonConfFile
Console Monitor 75
consMonMsgFilter
positive message filter 75
consMonNegMsgFilter
negative message filter 74
console interface
monitor 192
console message
filter 192
message code 192
routing code 192
Console Monitor
configuration 69
consMonMsgFilter 75
filter options 69
functionality 348
management application 348
message filter 71
message filter file 71
MIB 193
modify configuration file 75
QUESTION 73
sample configuration 444
start 132
stop 132
TYPE I/O messages 74
TYPIO 74

Console Monitor subagent
consMonConfFile 75
consMonNegMsgFilter 74
msgid 72
name convention (message filter file) 71
console query
reply 357
control
Application Monitor 189
application monitoring 56
the diagnosis (FT-BS2000) 217
create
configuration file (Application Monitor) 56
custom page 419
network map file 107
operator role 70
Symmetrix icons 112
table instance 409
curve diagram
PMBS2 386
custom page 12
configuring 419
creating 419
functionality 410
HTML initial document 420
interface/overview 414
parameter setting 414
preconfigured 410
ready to be configured 423
SNMP parameters 413
customer-specific web page
see custom page
customizing the web interface 418

D

daemon process, see trap server process
DCAM application
MIB 183
monitoring 183
DCAM return codes 457
default
Initial System Group 55
DEFINE-OBJECT
statement for the Application Monitor 66
DEFINE-TRAP-FORMAT
statement for the Application Monitor 68
definition 44
access check 34, 39, 43, 47
access to agents 33, 39, 43
address check 35, 41
community string 33
community string (examples) 36
DR-Web user ID 43
DR-Web user ID (example) 45
message filter 69
MIB branch 39, 43, 47
security entry 35, 40, 44, 48
security group 34, 40, 44, 48
sender address (trap) 49
SNMPv3 user 39
SNMPv3 user (example) 41
target address (trap) 46
target parameter (trap) 47
trap target 46
trap target (examples) 49
deinstallation 27
of TransView SNMP 110
of Unicenter TNG (Solaris) 103
of Unicenter TNG (Windows NT) 99
OpenView 119
SMBS2 110
SMBS2 (OpenView) 119
Unicenter TNG (Solaris) 103
deleting
SINLIB 25
description
trap window 350
detailed trap reception 31
diagnosis
control (FT-BS2000) 217
diagram
generate 388
diagram window
menu bar 383
parameter bar 384
PMBS2 383
work area 384

dialog box
commands 390
messages 390
PING 390
protocol 390
reactions 390
save 390
set options 389
set threshold 388
SNMP 390
systems 390
directories
trap server 432
display 333
command window 349
FT-BS2000 partner information 219
FT-BS2000 statistical information 217
FT-BS2000 system parameters 216
FT-BS2000 trap information 221
job networks (AVAS) 210
network state (AVAS) 211
object values 333, 339
processes (AVAS) 208
structure elements (AVAS) 211
trap window 349
values for DCAM applications 183
values for job variable 186
values for log files 187
values for objects 191
values for subsystems 184
display area
incoming traps 356
display filter
local filter 354
distribution of traps 434
DR-Web see web
DR-Web user ID
definition (example) 45
DR-Web 43

E

EMANATE 16, 399
end-of-line
BS2000 upicfile 87
environment variables
trap server 432
error
procedure in the event of 165
error handling 165
Event Queueing Subsystem 399
example
configuration 439
configuration file (Application Monitor subagent) 57
configuring the Application Monitor 448
configuring the basic monitoring 441
configuring the Console Monitor 444
configuring the Performance Monitor 452
monitoring UTM applications 444
performance monitoring 452
upicfile 87
exit
AVAS 77

F

figure
trap window 350
file
trap.cnf 377
trpsrvtargets 434
file monitoring
control 189
files
trap server 432
filter
console messages 192
filter options
Console Monitor 69
filter window
CMBS2 363
menu bar 363
toolbar 364
work area 364
form
PMBS2 384
format
of configuration file 56, 57
of the Initial System Group 55

FT subagent
start 138
stop 138
ftagt
starting the FT subagent 138
FT-BS2000
control diagnosis 217
MIB 215
partner information 219
public key encryption 217
start/stop 216
statistical information 217
system parameters 216
trap control 220, 221
trap groups 221
trap information 221
ftcmd
stop FT subagent 138
functionality 195
Console Monitor 348
HIPLEX OP 15
HNC-SNMP 15
PMBS2 379
proxy agent 15
SBA-BS2 12
SSA-OUTM-BS2 14
SSA-SM2-BS2 14
SSC-BS2 13
subagent 17
Symmetrix monitoring 195
trap security 203

G

generate
diagram 388
generation file
AVAS (GENPAR) 137
GENPAR 76, 137
GetNextRequest-PDU 8
GetRequest-PDU 8
GetResponse-PDU 8
groups dialog box 390

H

hardware requirements 22
header (SNMP) 8
help line
PMBS2 387
trap window 357
help text files
SMBS2 106
High-speed Net Connect, see HNC
HIPLEX 9
HIPLEX OP
functionality 15
HIPLEX-AF
MIB 222
HIPLEX-AF subagent
start 140
stop 140
histogram
PMBS2 387
HNC 9
HNC-SNMP
functionality 15
host
monitoring 338
host resources
MIB 232
Host Resources subagent
stop 144
HSMS
MIB 242
monitoring 290
HSMS subagent 242
global data 243
instances 243
requests 244
start 142
stop 142
HTML document
tag attribute 421
using parameters 423
HTML initial document
adding MIB objects 421
for custom page configuration 420

HTML subagent
overview 12, 13
starting 134
stopping 134
htmlGlobals
HTML-MIB 199
HTML-MIB
custom page configuration 424
tables 424
htmlPageContentTable 425
HTML-MIB 202
htmlPageEntry 426
htmlPageParameterEntry 427
htmlPageParameterTable 425
HTML-MIB 201
htmlPages
HTML-MIB 199
htmlPageTable 424
HTML-MIB 200
HTTP engine 399
HTTP request 399
httpUserNameEntry 43

I

icons
basic agent 338
incoming filter
local filter 354
set 354
incoming trap port 433
information
about statistics (FT-BS2000) 217
access via WWW 399
on installation 25
initial document (HTML)
for custom page configuration 420
Initial System Group 55
default 55
structure 55
input field
address 349
community 349
port 349
installation
CMBS2 (UNIX) 122
CMBS2 (Windows NT) 124
important information 25
in BS2000/OSD 25
management applications 120
management station 104
on Reliant UNIX 121
on TransView SNMP 105
on Unicenter TNG (Solaris) 101
on Unicenter TNG (Windows NT) 93
on Windows NT 124
PMBS2 (UNIX) 122
SBA-BS2 25
SMBS2 (requirements) 104
SMBS2 on OpenView 115
SMBS2 on TV Control Center 111
SMBS2 on TV SNMP 105
SSA-OUTM-BS2 26
SSA-SM2-BS2 26
SSC-BS2 25
Symmetrix 111
installation (UNIX)
tclset interpreter 121
installation (Windows NT)
interpreter tclset 124
integration
in management platform 91
in OpenView 114
in the user interface 333
in TransView Control Center 111
in TransView SNMP 104
in Unicenter TNG 333
in Unicenter TNG (Solaris) 100
in Unicenter TNG (Windows NT) 92
integration in the user interface
OpenView 346
TransView SNMP 338
Unicenter TNG 333
integration package 18
overview 19
interface/overview custom page 414

J
job networks
display (AVAS) 210
job variable
MIB 186
monitor (ADD-JV-RECORD) 64
monitoring 186
JVCENTRAL 77
JVPLAMZD 77
JVUPAMZD 77

L
local filter
display filter 354
incoming filter 354
set 354
log file 187
for monitoring 187
MIB 187
monitor (ADD-LOG-FILE-RECORD) 62

M
MAIN MENU
command program 167
main window
figure 380
menu bar 381
PMBS2 380
toolbar 382
work area 382
management
application 9
network 9
system 9
management application 18
BMBS2 20
CMBS2 20, 348
configuration (UNIX) 122
installation 120
installation (UNIX) 122
installation (Windows NT) 124
overview 91
PMBS2 20
SMBS2 333
management applications
overview 20
management architecture (SNMP) 6
management information
access via WWW 399
automatic refresh 406
web access 9, 20
Management Information Base 6
management platform
BS2000/OSD integration 18
integration in 91
TransView 18
Unicenter TNG 18
management station
installation 104
operation 333
setting the user interface 333
management station (TransView)
configuration 91
MAREN
monitoring 290
master agent 12
MIB 169
overview 12
start 128
stop 128
with web functions 400
MAX_OUTPUT_WAITING
Initial System Group 55
MAX_PDU_TIME
Initial System Group 55
MAX_SUBAGENTS 55
Initial System Group 55
MAX_THREADS
Initial System Group 55
menu bar
command window 359
filter window 363
main window 381
trap window 351
menu page
DR-Web 412
web agent 411

message code
console message 192
Console Monitor 71
message filter
definition 69
msgid 72
negative 69
positive 69
QUESTION 73
TYPIO 74
message filter file
Console Monitor 71
name convention 71
MIB 6
Application Monitor subagent 180
AVAS 207
basic performance monitoring 275
Console Monitor 193
FT-BS2000 215
HIPLEX-AF 222
host resources 232
HSMS 242
master agent 169
OMNIS 247
print service 286
SESAM 277
storage management 290
supervisor subagent 176
MIB branch
definition 39, 43, 47
MIB information
in raw data format 406
MIB objects
adding in HTML initial document 421
modify
configuration file (Console Monitor) 75
FT-BS2000 public key 217
table line 409
value of a scalar variable 408
monitor
with log file 187
monitoring
applications 182
BCAM application (ADD-APPLICATION-RECORD) 59
BS2000/OSD applications 338, 346
BS2000/OSD host 338
BS2000/OSD systems 333
console 444
console interface 192
DCAM applications 183
HSMS 290
job variable (ADD-JV-RECORD) 64
job variables 186
log file (ADD-LOG-FILE-RECORD) 62
MAREN 290
performance (example) 452
pubset 292
ROBAR 290
SPOOL device 286
subsystem (ADD-SUBSYSTEM-RECORD) 61
subsystems 183, 184, 191
Symmetrix control 195
system (MIB II) 169
TLS 290
user application (ADD-APPLICATION-RECORD) 59
UTM applications (example) 444
msgid
message filter 72
multithreading 16

N

name convention
message filter file (Console Monitor) 71
negative message filter 69
network management 9
network map file
create 107
map_SMBS2 107
SMBS2 107
network state
display (AVAS) 211
NMCP protocol 9

nmtrapd
trap daemon 122
notify entry
definition 46

O

object
MIB 191
object values 333
display 339
set 340
object view file
SMBS2 106
object views 106
object window
SMBS2 340
OMNIS
MIB 247
OMNIS subagent
configuration 78
OmnisMsg 343
start 144
stop 146
OmnisMsg 343
*open*UTM subagent
configuration 87
OpenView
configuring SMBS2 117
deinstallation 119
deinstallation SMBS2 119
installation on SMBS2 115
integration in 114
integration in the user interface 346
operating the management station 333
operator role
create 70
options
dialog box, set 389
overview
HIPLEX OP 15
HNC-SNMP 15
HTML subagent 12, 13
integration package 19
management applications 20
overview (continuation)
master agent 12
proxy agent 15
SNMP, administrable systems 10
SSA-OUTM-BS2 14
SSA-SM2-BS2 14
SSC-BS2 13
supervisor subagent 12

P

parameter bar
diagram window 384
parameter setting, custom page 414
parameters
set for current session 378, 389
parameters in the HTML document 423
PDU 8
type 8
Performance Monitor
PMBS2 management application 379
sample configuration 452
performance monitoring
example 452
performance subagent
(PerfMonF) start 157
start (PerfMonB) 154
PMBS2
curve diagram 386
diagram window 383
form 384
functionality 379
help line 387
histogram 387
installation (UNIX) 122
main window 380
management application 20
reactions 397
scalar objects 379
start 380
stop 380
table 385
table objects 379

port
input field 349
receive 434
SNMP 399
web-based 399
positive message filter 69
preconfigured custom page 410
print service subagent
start 146, 148, 150
PrintService subagent
stop 150
privileges
for starting the agents 125
for subagents required 126
procedure in the event of errors 165
processes
displaying (AVAS) 208
product structure
SNMP management for BS2000/OSD 11
protocol
NMCP 9
Protocol Data Unit (PDU) 8
proxy agent 15
functionality 15
public key encryption
FT-BS2000 217
pubset
monitor 292

Q

QUESTION
message filter 73

R

raw data format
MIB information 406
raw URL 406
rc scripts 127
RDBMS_relState
alarm (SESAM) 342
reactions
for PMBS2 397
Reactions dialog box 369
ready to be configured
custom page 423
receive port 434
refresh
management information 406
refresh URL 406
Reliant UNIX
trap server 431
reply
console query 357
request, authentication 31
requirements
hardware 22
installation (SMBS2) 104
start agents 125
RETRY_INTERVAL
Initial System Group 55
return codes, DCAM 457
RFC
ordering 478
RFC 1155 5
RFC 1157 5
RFC 1212 5
RFC 1213 5
RFC 1697 277
RFC 2271 5
RFC 2272 5
RFC 2273 5
RFC 2274 5
RFC 2275 5
ROBAR
monitoring 290
routing code
console message 192
Console Monitor 70
row URL 405
RSO
MIB 286
RSO device
monitor 286
runtime environment
UTM subagent 88
RZ exit
AVAS 77

S
SBA-BS2 12
functionality 12
installation 25, 26
software requirements 23
scalar objects
PMBS2 379
scalar variable
modify value 408
schematic
command window 358
trap window 350
security
traps 203
security configuration 28
configuration steps 31
example 53
security entry 32
configuration 40
definition 35, 44, 48
security group
definition 34, 40, 44, 48
security measures 55
security mechanism
in SNMPv3 30
selecting
table line 405
semicolon
BS2000 87
send
trap 161
sender adresss (trap)
definition 49
SEND-TCC-MSG 163
SESAM
MIB 277
SESAM subagent
ADD-DATA-BASE-RECORD 82
ADD-SERVER-PARAMETER 84
ADD-SERVER-RECORD 83
communication with SESAM/SQL server 79
configuration 78
RDBMS_relState 342
stop 148
set
current session parameters 378, 389
incoming filter 354
local filter 354
object values 340
options dialog box 389
SNMP parameters 349
threshold dialog box 388
trace extent 165
user interface CMBS2 349
set URL 407
SetRequest-PDU 8
Simple Network Management Protocol 5
SINLIB
deleting 25
SM2 subagent
(PerfMonF) stop 157
SMBS2
alarm 342
configuration on OpenView 117
deinstallation 110
deinstallation (OpenView) 119
help text files 106
installation on OpenView 115
installation on TV Control Center 111
installation on TV SNMP 105
installation requirements 104
management application 333
network map file 107
object view file 106
object window 340
software requirements 24
table window 339
SNMP 5
overview 10
SNMP agent
BCAM subagent 14
SNMP Basic Agent (SBA-BS2) 9, 12
SNMP Standard Collection (SSC-BS2) 9, 13
subagent for *open*UTM (SSA-OUTM-BS2) 14
subagent for SM2 (SSA-SM2-BS2) 14
SNMP header 8

SNMP management
architecture 6
for BS2000/OSD (product structure) 11
HNC 9
of BS2000/OSD 9
user interfaces 18
web-based 16
SNMP parameters
custom page 413
set 349
SNMP port 399
SNMP protocol elements 8
SNMP request
authentication 31
configuring the custom page 429
SNMP Standard Collection BS2000 9
snmpcmd
stopping the master agent 128
snmpdm
starting the master agent 128
snmpEnableAuthenTraps
Initial System Group 55
snmpNotifySourceEntry 49
snmpTargetAddrEntry 35, 41, 46
SNMPv1 5
SNMPv1 request
access to agents 33
SNMPv3 user
definition 39
definition (example) 41
software requirements
SNMP integration 23
Solaris
trap server 431
SPOOL
MIB 286
SPOOL device
monitoring 286
SSA-OUTM-BS2
functionality 14
installation 26
software requirements 24
start 159
stop 159
SSA-SM2-BS2
functionality 14
installation 26
software requirements 23
start 157
stop 157
SSC-BS2 13
functionality 13
installation 25, 26
software requirements 23
START 132
start
agents (necessary privileges) 125
Application Monitor 130
AVAS subagent 136
BCAM subagent 155
CMBS2 348
command program 165
Console Monitor 132
FT subagent 138
FT-BS2000 216
HIPLEX-AF subagent 140
HSMS subagent 142
HTML subagent 134
master agent 128
OMNIS subagent 144
performance subagent (PerfMonB) 154
PMBS2 380
print service subagent 146, 148, 150
rc scripts 127
SSA-OUTM-BS2 159
SSA-SM2-BS2 157
storage management subagent 152
supervisor subagent 55
trap server 433
UTM subagent 159
START-APPMONCMD
start command program 165
START-AVASCMD
start command program 165
START-BCAMCMD
start command program 165
START-CONSMONCMD
start command program 165

START-FTCMD
start command program 165
START-HSMSCMD
start command program 165
START-HTMLCMD
start command program 165
START-MASTERCMD
start command program 165
START-MIB2CMD
start command program 165
START-OMNISCMD
start command program 165
START-PERFMONCMD
start command program 165
START-PRINTCMD
start command program 165
START-SESAMCMD
start command program 165
START-SNMP-APPMON 130
START-SNMP-CONSMON 132
START-SNMP-FT 138
START-SNMP-HIPLEX-AF 140
START-SNMP-HOSTRES 144
START-SNMP-HSMS 142
START-SNMP-HTML 134
START-SNMP-MASTER 128
START-SNMP-MIB-BCAM 156
START-SNMP-MIB-MIB2 155
START-SNMP-OMNIS 146
START-SNMP-PERFMON 154, 157
START-SNMP-PRINTSERVICE 148, 150
START-SNMP-STORAGE 152
START-SNMP-TRAPSEND 161
START-SNMP-UTM 159
START-STORAGECMD
start command program 165
START-UTMCMD
start command program 165
statements
configuration file (SESAM) 80
statistical information
FT-BS2000 217
status
AVAS 209
stop
Application Monitor 131
AVAS subagent 136
BCAM subagent 155, 156
CMBS2 348
Console Monitor 132
FT subagent 138
FT-BS2000 216
HIPLEX-AF subagent 140
Host Resources subagent 144
HSMS subagent 142
HTML subagent 134
master agent 128
OMNIS subagent 146
performance subagent (PerfMonB) 154
performance subagent (PerfMonF) 157
PMBS2 380
PrintService subagent 150
SESAM subagent 148
SM2 subagent (PerfMonF) 157
SSA-SM2-BS2 157
storage management subagent 152
UTM subagent 159
STOP-SNMP-APPMON 131
STOP-SNMP-CONSMON 132
STOP-SNMP-FT 138
STOP-SNMP-HIPLEX-AF 140, 142
STOP-SNMP-HOSTRES 144
STOP-SNMP-HTML 134
STOP-SNMP-MASTER 128
STOP-SNMP-MIB-BCAM 156
STOP-SNMP-MIB-MIB2 155
STOP-SNMP-OMNIS 146
STOP-SNMP-PERFMON 154, 157
STOP-SNMP-PRINTSERVICE 148, 150
STOP-SNMP-STORAGE 152
STOP-SNMP-UTM 159
storage management subagent
ADD-DISK-RECORD 86
ADD-PUBSET-RECORD 85
configuration 85
MIB 290
start 152
stop 152

structure
Initial System Group 55
structure elements
display (AVAS) 211
subagent
command program 165
functionality 17
HSMS 242
Initial System Group 55
*open*UTM subagent 9
required privileges 126
SM2 subagent 9
SubagentStatus
alarm 342
SubagentStatus_e
alarm 342
subsystem
MIB 184
monitor (ADD-SUBSYSTEM-RECORD) 61
monitoring 183, 184, 191
subtree page (web agent) 403
subtree URL 404
subtree/sniSupervisor page 405
SupervisBasic
alarm 342
supervisor MIB 404
supervisor subagent
MIB 176
overview 12
start 55
sym-disk
Symmetrix alarm 344
sym-error
Symmetrix alarm 345
Symmetrix 195
alarms 344
configuration 111, 112
create icons 112
installation 111
sym-partner
Symmetrix alarm 344
sym-sp
Symmetrix alarm 344
sysContact 169
Initial System Group 55
sysDescr 169
Initial System Group 55
sysLocation 170
Initial System Group 55
sysName 169
sysObjectID 169
Initial System Group 55
sysServices 170
system
monitor 169
system list 391
system management 9
system parameters
FT-BS2000 216
system status
AVAS 209
SYSTRC.SNMP
trace file 127
sysUpTime 169, 338

T

table
PMBS2 385
table line
modify 409
selecting 405
table objects
PMBS2 379
table window
SMBS2 339
tag attribute in the HTML document 421
target address (trap)
definition 46
target configuration file 434
target parameter (trap)
definition 47
tclset
installation on Windows NT 124
threshold (dialog box)
set 388
TLS
monitoring 290

toolbar
command window 360
filter window 364
main window 382
trap window 353
trace extent
set 165
trace file 127
TransView 18
TransView Control Center
integration in 111
Symmetrix alarms 344
Symmetrix configuration 111, 112
Symmetrix installation 111
TransView SNMP
deinstallation 110
installation on 105
installation on SMBS2 105
integration in 104
integration in the user interface 338
TransView-SNMP-MIB 105
trap
Application Monitor 180
display in web browser 416
distribution 434
secured 203
send 161
trap confirmation window
CMBS2 366
work area 366
trap daemon
nmtrapd 122
trap distributor
configuation (Solaris) 103
trap filter 377
files and directories 377
trap format
Application Monitor subagent 68
Console Monitor subagent 70
trap receive program trprcv 438
trap reception
detailed 31
trap security
functionality 203
trap send command 161
trap send program
trpmsg 360, 438
trpsnd 437
trap serve process
communication 435
trap server 431
command program 435
environment variables 432
files and directories 432
incoming port 433
receive port 434
receive program 438
start 433
start server program 433
target configuration file 434
trap send program 437
trpcmd 435
trprcv 438
trpsnd 437
trap server process trpsrv 433
trap structure 180
Application Monitor-spec. trap 181, 194
TV-CC-MIB trap 181, 194
trap table 417
trap target
definition 46
definition (examples) 49
trap window
description 350
display 349
figure 350
help line 357
menu bar 351
schematic 350
toolbar 353
work area 354
trap-PDU 8
traps, incoming
display area 356
trp.cnf (trap filter file) 377
trpcmd (command program) 435
trpmsg (trap send program) 360, 438
trprec (trap receive program) 438

trpsnd (trap send program) 437
trpsrv (trap server process) 433
trpsrvtargets file 434
TV-CC-MIB trap
  structure 181, 194
  variable binding 183, 184, 186, 187, 188
TYPE I/O messages
  Console Monitor 74
TYPIO
  message filter 74

U
Unicenter TNG 18
  configuration (Solaris) 103
  configuration (Windows NT) 96
  deinstallation (Solaris) 103
  deinstallation (Windows NT) 99
  installation on Windows NT 93
  integration (Solaris) 100
  integration (Windows NT) 92
  integration in the user interface 333
Unicenter TNG (Solaris)
  installation on Solaris 101
Unicenter TNG deinstallation (Solaris) 103
Uniform Resource Locator see URL
upd-domain
  Symmetrix 112
upicfile 87
URL
  raw- 406
  refresh 406
  row 405
  set 407
  subtree 404
user application
  monitor (ADD-APPLICATION-RECORD) 59
user configuration
  DR-Web 429
user interface
  management station 333
  of OpenView 346
  of TransView SNMP 338
  of Unicenter TNG 333
usmUserEntry 39
UTM applications
  monitor (example) 444
UTM subagent
  communication with UTM application 87
  configuration 87
  runtime environment 88
  start 159
  stop 159

V
vacmAccessEntry 34, 35, 40, 44, 48
vacmSecurityToGroupEntry 40, 44, 48
vacmViewTreeFamilyEntry 34, 39, 43, 47
variable binding 183
  Application Monitor-spec. trap 183, 184, 186, 187, 188
  TV-CC-MIB trap 183, 184, 186, 187, 188

W
web access
  to management information 9, 20, 399, 431
web agent
  menu page 411
  subtree page 403
  use as web server 418
  welcome screen 402
web browser
  trap display 416
web interface 401
  custom page configuration 429
  customizing 418
web menu page 412
web page
  customer-specific, see custom page
  subtree/sniSupervisor 405
web server use of the web agent 418
web user configuration 429
web user ID
  definition 45
web-based
  management 16
  management port 399
welcome screen of the web agent 402
Windows NT 124

work area
command window 361
diagram window 384
filter window 364
main window 382
trap window 354
World Wide Web see WWW
WWW
access via 16

Contents

1	**Preface**	**1**
1.1	Contents of the manual	1
1.2	Target group	1
1.3	Summary of contents	2
1.4	Notational conventions	3
1.5	Changes compared to the previous version	4
1.6	README file	4
2	**Overview**	**5**
2.1	Basic features of the SNMP management architecture	6
2.2	SNMP management in BS2000/OSD - embedding and functionality	9
2.2.1	Product structure	11
2.2.2	Structure of the SNMP agent in BS2000/OSD	16
2.2.2.1	Master agent	16
2.2.2.2	Subagents	17
2.2.3	User interface for the SNMP management of BS2000/OSD	18
3	**Installation and configuration**	**21**
3.1	Software requirements	23
3.2	Installation in BS2000/OSD	25
3.2.1	Installing SBA-BS2 and SSC-BS2	26
3.2.2	Installing SSA-SM2-BS2	26
3.2.3	Installing SSA-OUTM-BS2	26
3.2.4	Version upgrading	27
3.2.5	Deinstallation	27
3.3	Configuring the agent in BS2000/OSD	28
3.3.1	Security configuration	28
3.3.1.1	Security mechanisms	28
3.3.1.2	Extended security mechanisms in SNMPv3	30
3.3.1.3	Configuration steps	31
3.3.1.4	Configuration file *snmpd.cnf*	32
3.3.1.5	Definition of access to the agent via SNMPv1 requests	33
3.3.1.6	Definition of access to agent via SNMPv3 requests	39
3.3.1.7	Definition of access to an agent via HTTP requests	43
3.3.1.8	Definition of the trap targets	46
3.3.1.9	Example	53

3.3.2	Configuring the master agent and supervisor subagent	55
3.3.3	Configuring the Application Monitor subagent	56
3.3.3.1	Statements for the configuration file	56
3.3.3.2	Change in the configuration file during the current session	58
3.3.4	Configuring the Console Monitor subagent	69
3.3.4.1	Defining message filters	69
3.3.4.2	Modifying the configuration file during operation	75
3.3.5	Configuring the AVAS subagent	76
3.3.6	Configuring the OMNIS subagent	78
3.3.7	Configuring the SESAM subagent	78
3.3.7.1	Communication between the SESAM subagent and the SESAM/SQL server	79
3.3.7.2	Configuration file statements	80
3.3.8	Configuring the subagent for storage management	85
3.3.9	Configuring the *open*UTM subagent (SSA-OUTM-BS2)	87
3.3.9.1	Preparation	87
3.3.9.2	Runtime environment	88
3.3.9.3	Diagnostic documents	89
3.4	Integration in the management platforms	91
3.4.1	Integration in CA Unicenter TNG under Windows NT	92
3.4.1.1	Installing on Unicenter TNG under Windows NT	93
3.4.1.2	Configuring Unicenter TNG	96
3.4.1.3	Deinstallation	99
3.4.2	Integration in CA Unicenter TNG under Solaris	100
3.4.2.1	Installation on Unicenter TNG under Solaris	101
3.4.2.2	Configuring Unicenter TNG	103
3.4.2.3	Configuring the trap distributor	103
3.4.2.4	Deinstallation	103
3.4.3	Integration in TransView SNMP	104
3.4.3.1	Installation on TransView SNMP	105
3.4.3.2	Configuring on TransView SNMP	107
3.4.3.3	Deinstallation	110
3.4.4	Integration in TransView Control Center	111
3.4.4.1	Installing on TransView Control Center	111
3.4.4.2	Configuring on TransView Control Center	112
3.4.5	Integration in the OpenView Network Node Manager	114
3.4.5.1	Installing on OpenView	115
3.4.5.2	Configuring on OpenView NNM	117
3.4.5.3	Deinstallation	119
3.5	Installing the management applications	120
3.5.1	Installing on Solaris and Reliant UNIX	121
3.5.2	Installing on Windows NT	124

4	**Operation**	**125**
4.1	Startup and shutdown	125
4.1.1	Master agent	128
4.1.2	BASIC AGENT subagents	130
4.1.2.1	Supervisor subagent	130
4.1.2.2	Application Monitor subagent	130
4.1.2.3	Console Monitor subagent	132
4.1.2.4	HTML subagent	134
4.1.3	Subagents in the STANDARD COLLECTION	136
4.1.3.1	AVAS subagent	136
4.1.3.2	Subagent for *open*FT	138
4.1.3.3	Subagent for HIPLEX-AF	140
4.1.3.4	HSMS subagent	142
4.1.3.5	Host Resources subagent	144
4.1.3.6	Subagent for OMNIS	146
4.1.3.7	Subagent for SESAM	148
4.1.3.8	Subagent for Spool & Print Service	150
4.1.3.9	Subagent for storage management	152
4.1.3.10	Subagent for basic performance monitoring with SM2 (PerfMonB)	154
4.1.4	Additive subagents	155
4.1.4.1	Subagents for *open*Net Server and *inter*Net Services	155
4.1.4.2	SSA-SM2-BS2 subagent for performance monitoring (with SM2)	157
4.1.4.3	Subagent SSA-OUTM-BS2 for *open*UTM applications	159
4.2	Trap send commands	161
4.2.1	START-SNMP-TRAPSEND	161
4.2.2	SEND-TCC-MSG	163
4.3	Procedure in the event of errors	165
5	**Functions of the BASIC AGENT**	**169**
5.1	System and SNMP management (master agent)	169
5.1.1	MIB-II values for the system group	169
5.1.2	MIB-II values for the SNMP group	171
5.1.3	SNMP framework MIB (SNMP engine)	175
5.1.4	Objects of other MIBs supported by the master agent	175
5.2	SNMP management for subagents (supervisor subagent)	176
5.3	Application Monitor subagent	179
5.3.1	Private MIB of the Application Monitor subagent	180
5.3.1.1	Trap structure	180
5.3.1.2	Monitoring the BCAM and user applications	182
5.3.1.3	Monitoring of DCAM applications	183
5.3.1.4	Monitoring subsystems	184
5.3.1.5	Monitoring job variables	186
5.3.1.6	Log file monitoring	187
5.3.1.7	Controlling file monitoring	189

5.3.1.8 Monitoring groups of associated elements . . . 191
5.4 Console Monitor subagent . . . 192
5.4.1 Acquiring console messages . . . 192
5.4.2 Symmetrix monitoring . . . 195
5.5 Custom pages (HTML subagent) . . . 199
5.6 Trap security . . . 203

6 Functions of the STANDARD COLLECTION . . . 207
6.1 SNMP management for AVAS . . . 207
6.2 SNMP management for *open*FT (BS2000) . . . 215
6.3 SNMP management for HIPLEX-AF . . . 222
6.4 SNMP management for Host Resources . . . 232
6.5 SNMP management for HSMS . . . 242
6.6 SNMP Management for OMNIS . . . 246
6.7 SNMP management for basic performance monitoring with SM2 . . . 275
6.8 SNMP management for SESAM databases . . . 277
6.9 SNMP management for Spool & Print Service . . . 286
6.10 SNMP management for storage management . . . 290

7 SNMP management for extended performance monitoring with SM2 . . . 299

8 SNMP management for monitoring *open*UTM and *open*UTM applications . . . 311

9 Operating the management station . . . 333
9.1 Integration in the user interface . . . 333
9.1.1 Integration in the user interface of Unicenter TNG . . . 333
9.1.1.1 NodeView display . . . 336
9.1.2 Integration in the user interface of TransView SNMP . . . 338
9.1.2.1 Monitoring the BS2000/OSD computer . . . 338
9.1.2.2 Monitoring of BS2000/OSD components . . . 338
9.1.2.3 Alarms . . . 342
9.1.3 Integration in TransView Control Center . . . 344
9.1.4 Integration in the user interface of HP OpenView . . . 346
9.2 Management applications CMBS2 and PMBS2 . . . 348
9.2.1 CMBS2 application for the Console Monitor subagent . . . 348
9.2.1.1 Setting the user interface . . . 349
9.2.1.2 Trap window . . . 350
9.2.1.3 Command window . . . 358
9.2.1.4 Filter window . . . 363
9.2.1.5 Trap confirmation window . . . 366
9.2.1.6 Reactions dialog box . . . 369
9.2.1.7 Trap filter . . . 377
9.2.1.8 Settings of the options dialog box . . . 378

9.2.2 PMBS2 application for the Performance Monitor . . . 379
9.2.2.1 Main window . . . 380
9.2.2.2 Diagram window . . . 383
9.2.2.3 Parameter bar . . . 384
9.2.2.4 Generating diagrams . . . 388
9.2.2.5 Threshold dialog box settings . . . 388
9.2.2.6 Options dialog box settings . . . 389

10 Web access to management information . . . 399
10.1 Overview . . . 399
10.2 The BS2000/OSD web agents interface (web interface) . . . 401
10.2.1 Connecting to the BS2000/OSD web agent . . . 401
10.2.2 Subtree functionality . . . 403
10.2.2.1 Subtree page of the web agent (DR-Web subtree page) . . . 403
10.2.2.2 Subtree URL - GetRequest functionality . . . 404
10.2.2.3 The row URL - selecting single table rows . . . 405
10.2.2.4 The raw-URL - representation of MIB information in “raw data” format . . . 406
10.2.2.5 The refresh URL - Automatic refresh of management information . . . 406
10.2.2.6 The set URL - SetRequest functionality . . . 407
10.2.3 Custom Page functionality . . . 410
10.2.3.1 Preconfigured Custom Pages . . . 410
10.2.3.2 DR-Web menu page . . . 411
10.2.3.3 Parameterizing the custom page . . . 414
10.2.4 Trap display in the web browser . . . 416
10.2.5 Using the web agent as a web server . . . 418
10.2.6 Customizing the DR-Web interface . . . 418
10.3 Configuring a custom page . . . 419
10.3.1 Creating the custom page . . . 419
10.3.2 Configuring the custom page using HTML-MIB . . . 424
10.3.2.1 Configuring the HTML-MIB tables . . . 424
10.3.2.2 Configuring the custom page in a configuration file . . . 426
10.3.2.3 Configuring the custom page using SNMP requests . . . 429
10.3.2.4 Configuring the custom page using the DR-Web interface . . . 429
10.4 DR-Web user configuration . . . 429

11 Trap server for Solaris and Reliant UNIX . . . 431
11.1 Files and directories . . . 432
11.2 Environment variables . . . 432
11.3 Trap server process *trpsrv* (daemon process) . . . 433
11.4 Command program *trpcmd* . . . 435
11.5 Trap send program *trpsnd* . . . 437
11.6 Trap send program *trpmsg* . . . 438
11.7 Trap receive program *trprcv* . . . 438

12 Configuration examples . . . 439
12.1 Basic monitoring . . . 441
Task . . . 441
Configuration . . . 441
Result . . . 442
12.2 Monitoring messages with the Console Monitor subagent . . . 444
Task . . . 444
Configuration . . . 444
Result . . . 446
12.3 Monitoring applications with the Application Monitor subagent . . . 448
Task . . . 448
Configuration . . . 448
Result . . . 450
12.4 Monitoring the systems using the Performance Monitor subagent . . . 452
Task . . . 452
Configuration . . . 452
Result . . . 454

13 Appendix: DCAM return codes . . . 457

Glossary . . . 463

Related publications . . . 469

Environmental protection . . . 479

Index . . . 481

SNMP Management V5.0
SNMP Management for BS2000/OSD

User Guide

Target group
This manual is intended for network administrators/operators and system administrators wishing to integrate BS2000 systems into an SNMP-based management strategy or to work with such systems.

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.

Edition: July 2000

File: snmp.pdf

Fujitsu Siemens computers GmbH
User Documentation
81730 Munich
Germany

Fax: (++49) 700 / 372 00000

e-mail: DOCetc@mchp.siemens.de
http://manuals.mchp.siemens.de

Comments
Suggestions
Corrections

Submitted by

Comments on SNMP Management V5.0
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.