

JP1 Version 12

**JP1/Extensible SNMP Agent Description,
Operator's Guide and Reference**

3021-3-E05(E)

Notices

■ Relevant program products

P-1J42-8ACL JP1/Extensible SNMP Agent 12-00 (for HP-UX (IPF))

P-9D42-8ACL JP1/Extensible SNMP Agent 12-00 (for Solaris)

P-1M42-8ACL JP1/Extensible SNMP Agent 12-00 (for AIX)

P-8142-8ACL JP1/Extensible SNMP Agent 12-00 (for Linux)

This software and documentation are based on software and documentation licensed from Hewlett-Packard Company.

■ Trademarks

HITACHI, JP1 are either trademarks or registered trademarks of Hitachi, Ltd. in Japan and other countries.

AMD, AMD Opteron, and combinations thereof, are trademarks of Advanced Micro Devices, Inc.

IBM, AIX are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide.

IBM, HACMP are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide.

IBM, PowerHA are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide.

Itanium is a trademark of Intel Corporation in the United States and other countries.

Linux^(R) is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Red Hat is a trademark or a registered trademark of Red Hat Inc. in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Veritas and the Veritas Logo are trademarks or registered trademarks of Veritas Technologies LLC or its affiliates in the U.S. and other countries.

Windows is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

Other company and product names mentioned in this document may be the trademarks of their respective owners.

The following program products contain some parts whose copyrights are reserved by Oracle and/or its affiliates:
P-9D42-8ACL.

The following program products contain some parts whose copyrights are reserved by UNIX System Laboratories, Inc.: P-9D42-8ACL.

HITACHI
Inspire the Next

Hitachi, Ltd.



■ Restrictions

Information in this document is subject to change without notice and does not represent a commitment on the part of Hitachi. The software described in this manual is furnished according to a license agreement with Hitachi. The license agreement contains all of the terms and conditions governing your use of the software and documentation, including all warranty rights, limitations of liability, and disclaimers of warranty.

Material contained in this document may describe Hitachi products not available or features not available in your country.

No part of this material may be reproduced in any form or by any means without permission in writing from the publisher.

■ Issued

Jan. 2019:3021-3-E05(E)

■ Copyright

All Rights Reserved. Copyright (C) 2019, Hitachi, Ltd.

Copyright (C) 1993-1998, Hewlett-Packard Company.

Copyright (C) 1989-2010, SNMP Research International, Incorporated.

Summary of amendments

The following table lists changes in this manual (3021-3-E05(E)) and product changes related to this manual.

Changes	Location
SNMP Agent no longer supports the following operating systems: <ul style="list-style-type: none">• Solaris 10• AIX V 6.1	--
Supported adding file system target definition file <code>/etc/SnmpAgent.d/esafstab.conf</code> .	<i>2.12.1(2), Adding file system target definition file (esafstab.conf)</i>
Added files <code>jp1esalog.sh.def</code> gets.	<i>jp1esalog.sh.def</i>

In addition to the above changes, minor editorial corrections were made.

Preface

This manual describes the agent functions of the following program products. It also explains how to use these functions.

- JP1/Extensible SNMP Agent

In this manual, the agent functions are referred to as the *SNMP Agent*.

■ Intended readers

This manual is intended for the following individuals:

- Network administrators who are in charge of administering a network using a product covered by this manual
- Network operators who are in charge of operating and/or maintaining a product covered by this manual

Readers of this manual must have:

- A basic knowledge of UNIX (HP-UX (IPF), Solaris, AIX, and Linux^(R))
- A basic knowledge of the SNMP protocol and management methods for TCP/IP networks that use SNMP

■ Conventions: Fonts and symbols

The following table explains the text formatting conventions used in this manual:

Text formatting	Convention
Bold	Bold characters indicate text in a window, other than the window title. Such text includes menus, menu options, buttons, radio box options, or explanatory labels. For example: <ul style="list-style-type: none">• From the File menu, choose Open.• Click the Cancel button.• In the Enter name entry box, type your name.
<i>Italic</i>	Italic characters indicate a placeholder for some actual text to be provided by the user or system. For example: <ul style="list-style-type: none">• Write the command as follows: <code>copy source-file target-file</code>• The following message appears: A file was not found. (file = <i>file-name</i>) Italic characters are also used for emphasis. For example: <ul style="list-style-type: none">• Do <i>not</i> delete the configuration file.
Monospace	Monospace characters indicate text that the user enters without change, or text (such as messages) output by the system. For example: <ul style="list-style-type: none">• At the prompt, enter <code>dir</code>.• Use the <code>send</code> command to send mail.• The following message is displayed: <code>The password is incorrect.</code>

The following table explains the symbols used in this manual:

Symbol	Convention
	In syntax explanations, a vertical bar separates multiple items, and has the meaning of OR. For example: A B C means A, or B, or C.
{ }	In syntax explanations, curly brackets indicate that only one of the enclosed items is to be selected. For example: { A B C } means only one of A, or B, or C.
[]	In syntax explanations, square brackets indicate that the enclosed item or items are optional. For example: [A] means that you can specify A or nothing. [B C] means that you can specify B, or C, or nothing.
. . .	In coding, an ellipsis (. . .) indicates that one or more lines of coding have been omitted. In syntax explanations, an ellipsis indicates that the immediately preceding item can be repeated as many times as necessary. For example: A, B, B, . . . means that, after you specify A, B, you can specify B as many times as necessary.
<>	Single angle brackets indicate the type of value the system assumes.
<<>>	Double angle brackets indicate the default value.
(())	Double parentheses indicate the range of values that can be specified.

■ Conventions: Version numbers

The version numbers of Hitachi program products are usually written as two sets of two digits each, separated by a hyphen. For example:

- Version 1.00 (or 1.0) is written as 01-00.
- Version 2.05 is written as 02-05.
- Version 2.50 (or 2.5) is written as 02-50.
- Version 12.25 is written as 12-25.

The version number might be shown on the spine of a manual as *Ver. 2.00*, but the same version number would be written in the program as *02-00*.

Contents

Notices 2

Summary of amendments 4

Preface 5

1 Introduction to SNMP Agent 13

- 1.1 About SNMP Agent 14
 - 1.1.1 System configuration of SNMP Agent 14
 - 1.1.2 Operating environment for SNMP Agent 15
- 1.2 Functions of SNMP Agent 16
 - 1.2.1 SNMP request processing 16
 - 1.2.2 Issuing SNMP traps 17
 - 1.2.3 Extended functions 19
- 1.3 SNMP Agent processes 21
 - 1.3.1 Processes that constitute SNMP Agent 21
 - 1.3.2 SNMP Agent startup processing 24
 - 1.3.3 Operations using SNMP Agent 24
- 1.4 SNMP Agent commands 25

2 Setting up an Environment for SNMP Agent 26

- 2.1 Procedures from installation to setup 27
- 2.2 Installing 28
 - 2.2.1 Preparations for installation 28
 - 2.2.2 Installing the commands used to acquire MIB values (for an OS other than HP-UX (IPF)) 28
 - 2.2.3 Installing SNMP Agent 29
 - 2.2.4 How to use Hitachi Program Product Installer 30
- 2.3 Performing an upgrade installation 32
 - 2.3.1 Backing up the customized definition files 32
 - 2.3.2 Setting the SNMP trap transmission port number (when performing overwrite installation on version 07-10 or earlier) 33
 - 2.3.3 Setting the log output options (when performing overwrite installation on version 07-10 or earlier) 33
 - 2.3.4 Specifying IPv6 trap destinations (when performing overwrite installation on version 09-00 or earlier) 36
- 2.4 Notes about installation 37
 - 2.4.1 Notes about installation (for HP-UX (IPF)) 37
 - 2.4.2 Notes about installation (for Solaris) 38
 - 2.4.3 Notes about installation (for AIX) 39
 - 2.4.4 Notes about installation (for Linux) 39

2.5	Uninstalling	41
2.5.1	How to uninstall SNMP Agent	41
2.5.2	Notes about uninstallation	41
2.6	Setting up the operating locale	43
2.7	Customizing the configuration file (/etc/SnmpAgent.d/snmpd.conf)	44
2.7.1	Specifying the system contact and system location	44
2.7.2	Specifying community names	45
2.7.3	How to register community names	45
2.7.4	How to specify community names	47
2.7.5	Sending authentication failure traps	49
2.7.6	Specifying trap destinations	49
2.7.7	Format of the configuration file	50
2.8	IPv6 settings	54
2.8.1	IPv6 transport and trap destination settings	54
2.8.2	IPv6 address format	56
2.9	Setting up the native agent adapter (for Solaris, AIX, and Linux)	57
2.9.1	Functions of the native agent adapter	57
2.9.2	Configuring the native agent (for Solaris and AIX)	59
2.9.3	How to configure the native agent adapter	59
2.9.4	Changing the communication protocol with the native agent	59
2.9.5	Notes about using the native agent adapter	60
2.10	Defining extended MIB objects	61
2.10.1	Defining MIB objects	62
2.10.2	Configuring an extended MIB definition file	63
2.10.3	Creating the shell commands to be executed during SNMP request	70
2.10.4	Creating the file to be processed during an SNMP request	74
2.10.5	Reconfiguring the subagent	75
2.10.6	Verifying the objects using manager commands	75
2.10.7	Setting up all SNMP Agent instances	75
2.10.8	Copying extended MIB objects to the manager	75
2.10.9	Integrating MIBs into the manager	75
2.10.10	Configuring more than one extended MIB definition file	76
2.10.11	Example of extended MIB object definition	79
2.11	Defining enterprise-specific traps	94
2.11.1	How to define enterprise-specific traps	94
2.11.2	How to use enterprise-specific traps	94
2.11.3	Sample script	94
2.12	Settings for operations in a cluster environment	96
2.12.1	Required settings for monitoring shared disks (for Linux)	96
2.12.2	Settings for suppressing an invalid shared disk capacity response (for AIX and Linux)	97
2.12.3	Settings for using PowerHA (HACMP)	98

- 2.13 Notes about the amount of free space in physical memory 100
- 2.14 Notes about swap space size 102
- 2.15 Notes about CPU information 103
- 2.16 Settings to prevent responses with information about file systems for which a response is not required (for Linux) 105
- 2.17 Notes about setup 106
- 2.17.1 Notes about setup (for AIX) 107
- 2.17.2 Notes about setup (for Linux) 108

3 Operating SNMP Agent 109

- 3.1 Starting SNMP Agent 110
 - 3.1.1 Customizing the startup options and defining environment variables for the processes 110
 - 3.1.2 Environment variable definition files provided by SNMP Agent 111
 - 3.1.3 Startup options that can be specified in the environment variable definition files 111
 - 3.1.4 Environment variables that can be specified for processes 112
 - 3.1.5 Files that are executed during system startup 113
- 3.2 Terminating SNMP Agent 115
 - 3.2.1 Notes about terminating processes individually 115
 - 3.2.2 Files that are executed during system shutdown 115
- 3.3 Starting and terminating the native agent 117
- 3.4 Changing the SNMP reception port on SNMP Agent 118
 - 3.4.1 Changing the SNMP reception port on SNMP Agent 118
 - 3.4.2 Changing the SNMP reception port on the native agent snmpd (for AIX) 119
- 3.5 Changing the maximum number of connected subagents 120
- 3.6 Backing up and restoring 121
 - 3.6.1 Backing up and restoring the configuration files 121
 - 3.6.2 Notes about full-backup and full-restoration 121
- 3.7 Notes about operations 123
 - 3.7.1 Notes about operations (for Solaris) 125
 - 3.7.2 Notes about operations (for AIX) 125
 - 3.7.3 Notes about operations (for Linux) 126
 - 3.7.4 Notes about renaming a host 126

4 MIB Objects 128

- 4.1 Standard MIB objects 129
 - 4.1.1 Organization of standard MIB objects 129
 - 4.1.2 Description of standard MIB objects 129
 - 4.1.3 Implementation of standard MIB objects 138
- 4.2 Hewlett-Packard enterprise-specific MIB objects 140
 - 4.2.1 Organization of Hewlett-Packard enterprise-specific MIB objects 140
 - 4.2.2 Description of Hewlett-Packard enterprise-specific MIB objects 140
 - 4.2.3 Implementation of Hewlett-Packard enterprise-specific MIB objects 148

- 4.3 Hitachi enterprise-specific MIB objects 154
- 4.3.1 Organization of Hitachi enterprise-specific MIB objects 154
- 4.3.2 Description of Hitachi enterprise-specific MIB objects 156
- 4.3.3 Implementation of Hitachi enterprise-specific MIB objects 188

5 Commands and Processes 214

- Commands 215
- Details of commands 216
- jp1esalog.sh.def 217
- snmpcheck 222
- snmpcmdchk 223
- snmpstart 224
- snmpstop 225
- snmptrap 226
- systemtrap 229
- trapsend 230
- Processes 234
- Detailed process descriptions 235
- snmpdm 236
- extsubagt 239
- hp_unixagt 241
- htc_monagt1 243
- htc_unixagt1 246
- htc_unixagt2 249
- htc_unixagt3 251
- htc_unixagt4 253
- naaagt 255
- trapdestagt 257

6 Definition Files 259

- About definition files 260
- Definition file description format 263
- Configuration file (snmpd.conf) 264
- Configuration file (snmpd.cnf) 267
- Configuration file (naa.cnf) 270
- Environment variable definition file (SnmpMaster) 273
- Environment variable definition file (SnmpNaa) 278
- Environment variable definition file (SnmpNative) 280
- Environment variable definition file (SnmpHpunix) 282
- Environment variable definition file (SnmpTrpDst) 285
- Environment variable definition file (SnmpHtcunix1) 287
- Environment variable definition file (SnmpHtcunix2) 289
- Environment variable definition file (SnmpHtcunix3) 291
- Environment variable definition file (SnmpHtcunix4) 293
- Environment variable definition file (SnmpHtcmonagt1) 295
- Environment variable definition file (SnmpExtAgt) 297

Operating locale definition file (esalocale.conf)	299
File system definition file (esafsys.conf)	300
Adding file system target definition file (esafstab.conf)	302
Disk definition file (esadisk.conf)	304

7 Troubleshooting 306

7.1	General troubleshooting procedure	307
7.2	Identifying the problem	308
7.3	Collecting logs	309
7.3.1	Log type	309
7.3.2	Log output destination	310
7.3.3	Number and size of the log files	311
7.3.4	Notes about logs	312
7.4	Collecting data	313
7.4.1	Acquiring a master agent send/receive packet dump	313
7.4.2	Acquiring a native agent adapter send/receive packet dump	314
7.4.3	Acquiring detailed trace information about the master agent	315
7.4.4	Collecting logs of unauthorized community names	317
7.5	Taking corrective action	318
7.5.1	Problems when SNMP Agent is starting	318
7.6	Problems when SNMP Agent is running	319
7.6.1	MIB values cannot be acquired	319
7.6.2	SNMP traps do not reach the manager	320
7.6.3	The SNMP Agent extended function cannot be used	320
7.7	Method for collecting log information	322

Appendixes 323

A	SNMP Agent Files	324
A.1	List of SNMP Agent files (HP-UX (IPF))	324
A.2	List of SNMP Agent files (Solaris)	327
A.3	List of SNMP Agent files (AIX)	331
A.4	List of SNMP Agent files (Linux)	336
B	Port Numbers	341
B.1	Port numbers used by SNMP Agent	341
B.2	Direction in which data passes through a firewall	341
C	List of Kernel Parameters	342
C.1	HP-UX (IPF)	342
C.2	Solaris	342
C.3	AIX	342
C.4	Linux	342
D	List of Prerequisite Patches and Processes (Services) for SNMP Agent	343
E	Version Changes	344

E.1	Revisions in version 12-00	344
E.2	Revisions in version 11-00	344
E.3	Changes from version 10-10 to version 10-50	345
E.4	Changes from version 10-00 to version 10-10	345
E.5	Revisions in version 10-00	346
E.6	Revisions in version 09-00	346
F	Reference Material for This Manual	348
F.1	Related publications	348
F.2	Conventions: Abbreviations for product names	348
F.3	Conventions: Acronyms	349
F.4	Conventions: File naming	349
F.5	Conventions: KB, MB, GB, and TB	351
G	Glossary	352

Index 354

1

Introduction to SNMP Agent

SNMP Agent runs on manager systems and on agent systems, and controls the system on which they run. This chapter outlines SNMP Agent.

1.1 About SNMP Agent

SNMP Agent is an agent that manages a TCP/IP network. SNMP Agent manages TCP/IP networks by using an Internet network management protocol called *SNMP* to exchange management information called *MIB* with managers.

This manual defines SNMP Agent types as follows:

- SNMP Agent
 - JP1/Extensible SNMP Agent
- Native agent
 - An SNMP agent provided by the OS
- SNMP agent of another company
 - An SNMP agent provided by another company

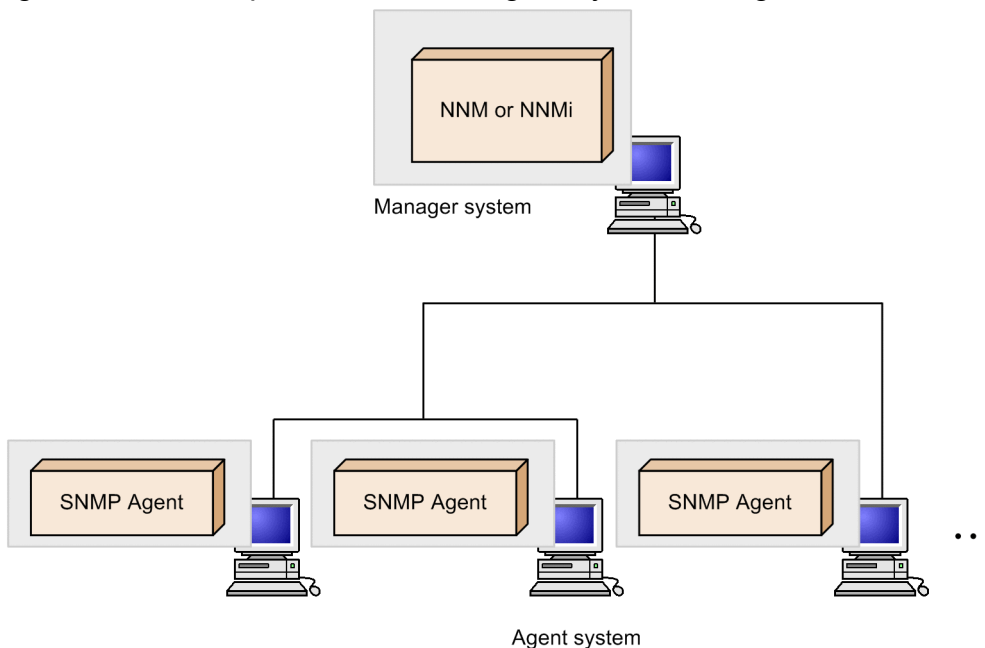
1.1.1 System configuration of SNMP Agent

SNMP Agent runs on a manager system or an agent system that constitutes a network.

In this manual, a machine on which NNM or NNMi is installed is called a *manager system*, and a machine on which SNMP Agent is installed is called an *agent system*.

The following figure shows an example of an SNMP Agent system configuration.

Figure 1–1: Example of an SNMP Agent system configuration



Note: NNM or NNMi may exist in the same system as the SNMP Agent.

1.1.2 Operating environment for SNMP Agent

The following table lists and describes the systems on which SNMP Agent can run and the supported operating systems.

Table 1–1: Systems on which SNMP Agent can run and the supported OS

Product name	Supported system	OS#
Extensible SNMP Agent	HA8500 series, HP Integrity server and its compatibles, BladeSymphony	HP-UX (IPF)
	SUN SPARC series, SUN Fire series, SUN Netra series, SUN Ultra series and their compatibles, SUN Blade series, PRIMEPOWER	Solaris
	EP8000 series, IBM Power Systems	AIX
	PC/AT compatible that can install Linux, BladeSymphony, HA8000 series, Oracle Database Appliance, Oracle Exadata Database Machine, Oracle Exalogic Elastic Cloud, Oracle Exalytics In-Memory Machine	Linux

#

SNMP Agent can be run on an OS in the 64-bit kernel mode; however, SNMP Agent itself still functions as a 32-bit application.

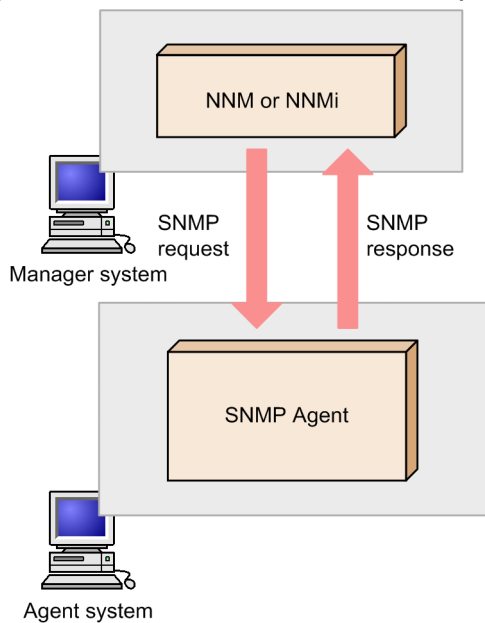
1.2 Functions of SNMP Agent

SNMP Agent has a function for responding to SNMP requests from the manager and a function for sending change events in the agent system (*SNMP traps*) to the manager. SNMP Agent also has extended functions that make it possible to define user-specific MIB objects (*extended MIB objects*) and SNMP traps (*enterprise-specific traps*).


1.2.1 SNMP request processing

An SNMP request is a request from the manager to access MIB values that are managed by SNMP Agent. The following figure shows an overview of SNMP request processing.

Figure 1–2: Overview of SNMP request processing



Legend:

 : Flow of messaging

There are three types of SNMP requests: SNMP GET (acquisition) requests, SNMP SET (setting) requests, and SNMP GET NEXT (acquisition) requests. When SNMP Agent receives an SNMP request, it parses the received SNMP request and acquires and sets values. After that, SNMP Agent creates a response message containing the processing result and sends it to the manager.

For SNMP request processing, SNMP Agent supports SNMPv1 and SNMPv2c. SNMP Agent can communicate using either IPv4 or IPv6.

The following describes the MIB objects that can be acquired and set by SNMP Agent.

(1) MIB objects managed by SNMP Agent

SNMP Agent implements not only standard MIB objects, but also Hewlett-Packard enterprise-specific MIB objects and Hitachi enterprise-specific MIB objects. For details about the MIB objects implemented by SNMP Agent, see [4. MIB Objects](#).

(2) MIB objects provided by native agents

A native agent is a standard agent that is provided by the system vendor.

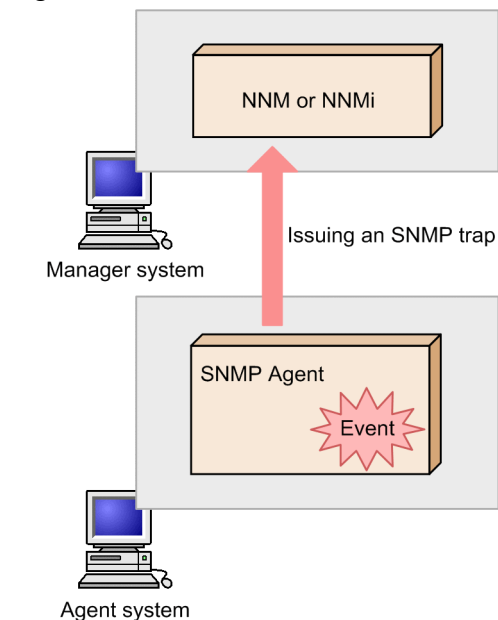
On HP-UX (IPF), SNMP Agent's master agent communicates directly with the native agent to acquire the MIB objects provided by the native agent.

On an OS other than HP-UX (IPF), you can use SNMP Agent's native agent adapter function to acquire the MIB objects provided by the native agent. For details about the native agent adapter function and settings, see [2.9 Setting up the native agent adapter \(for Solaris, AIX, and Linux\)](#).


1.2.2 Issuing SNMP traps

SNMP Agent sends a change event in the agent system as an SNMP trap to the manager. The following figure shows an overview of SNMP trap issuance.

Figure 1–3: Overview of SNMP trap issuance



Legend:

 : Flow of messaging

At startup, or when an invalid SNMP request is received, SNMP Agent creates an SNMP trap message and sends it to the manager. SNMP trap messages sent over IPv4 support SNMPv1. SNMP trap messages sent over IPv6 support SNMPv1 and SNMPv2c.



Note

The user can specify which SNMP traps are to be considered important and displayed as alarms at the manager. This enables the user to easily monitor the alarms displayed at the manager and take appropriate actions to maintain normal network operation.

The following describes SNMP traps.

(1) Standard trap numbers of SNMPv1 traps

The two types of SNMPv1 traps are generic traps and enterprise-specific traps. These SNMPv1 traps are described in RFC 1157 as follows:

Generic trap

A trap identified by a standard trap number from 0 to 5, as stipulated in RFC 1157

Enterprise-specific traps

A trap identified by a combination of the standard trap number 6, stipulated in RFC 1157, and a user-specific trap number

The following table shows the list of SNMPv1 trap types and standard trap numbers.

Table 1–2: List of SNMPv1 trap types and standard trap numbers

SNMPv1 trap type	Standard trap number	Description
Generic trap [#]	0	coldStart
	1	warmStart
	2	linkDown
	3	linkUp
	4	authenticationFailure
	5	egpNeighborLoss
Enterprise-specific traps	6	enterpriseSpecific

#

SNMP Agent issues only the `coldStart` generic trap and the `authenticationFailure` generic trap. It does not issue any other generic traps.

This subsection describes generic traps. For details about enterprise-specific traps, see [1.2.3\(2\) Definition of enterprise-specific traps](#).

(2) Agent address at the time of SNMP trap issuance

If the SNMPv1 trap is issued when IPv4 is being used, the agent address (the value stored in the `Agent Address` field of the SNMPv1 trap PDU) is the IPv4 address of the machine on which SNMP Agent is installed. You can obtain it by converting the host name to the IP address using the applicable system's OS functions.

If the SNMPv1 trap is issued when IPv6 is being used, the agent address is set to the following IPv4 address.

Table 1–3: Value of agent address for SNMPv1 trap when IPv6 is being used

Condition	Value
Default	0.0.0.0
Destination address is ::1	127.0.0.1
Destination address is an IPv4-mapped address, and <code>ip_proto</code> is not <code>ipv6</code> .	Determined by obtaining the host name using the function provided by the OS and converting it to an IPv4 address using the function provided by the OS.

If the value of the agent address is 0.0.0.0, the trap might not be handled correctly. For example, if the manager references this field and determines that the trap is from the unknown IP address 0.0.0.0, it will not be able to handle

the trap properly. In this case, one solution might be to issue an SNMPv2c trap. For details about the settings to use for issuing SNMPv2c traps over IPv6, see [2.8 IPv6 settings](#).

(3) Enterprise ID

The following table lists the enterprise IDs that are set in SNMPv1 traps.

Table 1–4: Enterprise IDs that are set in SNMP traps

OS on which SNMP Agent is running	Enterprise ID	Applicable system configuration
HP-UX (IPF)	.1.3.6.1.4.1.116.3.9.1.1	SNMP Agent and NNM coexist in the system.
	.1.3.6.1.4.1.116.3.9.1.3 [#]	SNMP Agent and NNM do not coexist in the system.
	.1.3.6.1.4.1.116.3.9.1.4	SNMP Agent and NNM coexist in the system.
Solaris	.1.3.6.1.4.1.116.3.8.1.1	SNMP Agent and NNM coexist in the system.
	.1.3.6.1.4.1.116.3.8.1.3 [#]	SNMP Agent and NNM do not coexist in the system.
	.1.3.6.1.4.1.116.3.8.1.4	SNMP Agent and NNM coexist in the system.
AIX	.1.3.6.1.4.1.116.3.13.1.3	SNMP Agent is installed.
Linux	.1.3.6.1.4.1.116.3.14.1.3 [#]	SNMP Agent is installed.

#

Even if SNMP Agent and NNMi coexist in the system, an enterprise ID ending with 3 is set.

The value of `sysObjectID` in the System group that is the standard MIB object of SNMP Agent is set as the enterprise ID in SNMPv1 traps.



Note

If you need to set events for the SNMP trap at the manager, use the enterprise ID that is set in the SNMP trap issued by SNMP Agent. For NNM events, SNMP trap information reported from SNMP Agent is set.

(4) Object IDs of SNMPv2c traps

The only SNMPv2c traps issued by SNMP Agent when IPv6 is being used are the `coldStart` trap and the `authenticationFailure` trap.

The following table lists the object IDs of the SNMPv2c traps.

Table 1–5: Object IDs of SNMPv2c traps

Object ID of SNMPv2c trap	Meaning
.1.3.6.1.6.3.1.1.5.1	<code>coldStart</code>
.1.3.6.1.6.3.1.1.5.5	<code>authenticationFailure</code>

1.2.3 Extended functions

This subsection describes the following extended functions of SNMP Agent:

- Definition of extended MIB objects
- Definition of enterprise-specific traps

(1) Definition of extended MIB objects

You can implement MIBs defined by hardware vendors or standardization organizations as extended MIB objects. To do so, you must define in a file the MIB object you want to implement as an extended MIB object, according to the ASN.1 encoding rules. You must also define the action that SNMP Agent performs upon receiving an SNMP request for this MIB object. Once you make these definitions, SNMP Agent, upon receiving an SNMP request from a manager, performs the specified action and notifies the manager of the result of the action.

You can use extended MIB objects for the following purposes:

- Managing information specific to your enterprise as MIB objects
- Starting and stopping applications specific to your enterprise

(2) Definition of enterprise-specific traps

SNMP Agent can report user-specific traps to the manager as enterprise-specific traps. The trap number of an enterprise-specific trap is defined by the trap number 6 and a user-specific trap number. For details about trap numbers, see [1.2.2\(1\) Standard trap numbers of SNMPv1 traps](#).

SNMP Agent provides the `snmptrap` and `trapsend` commands to send notifications. For example, when an important process stops, you can use the `snmptrap` command to send an enterprise-specific trap to the manager. For details about the `snmptrap` and `trapsend` commands, see [snmptrap](#) and [trapsend](#) in [Chapter 5. Commands and Processes](#).

1.3 SNMP Agent processes

This section describes the processes that constitute SNMP Agent. It also describes the processes that take place during SNMP Agent startup and operations.

1.3.1 Processes that constitute SNMP Agent

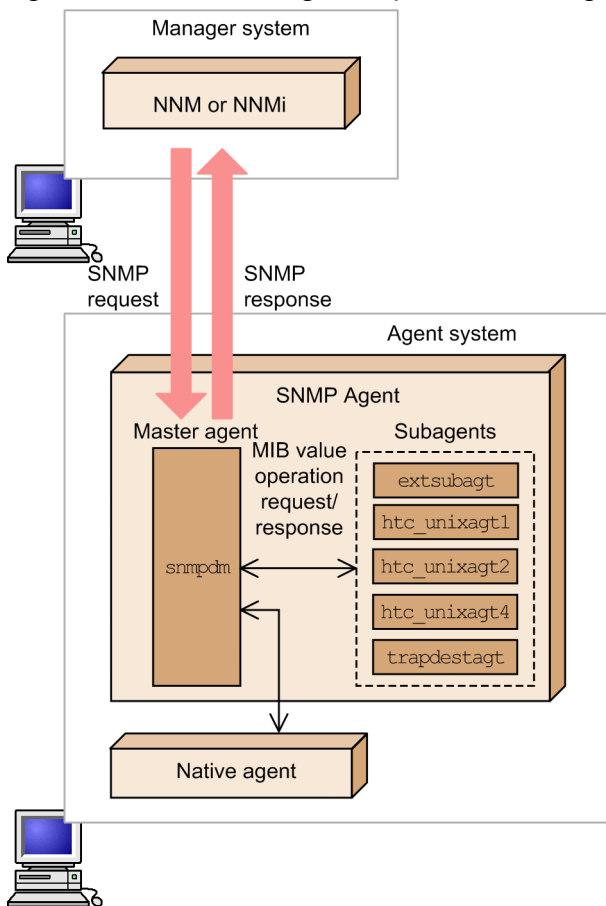
SNMP Agent consists of a master agent and subagents. When SNMP Agent runs on an OS other than HP-UX (IPF), it also uses an information collection daemon that periodically acquires information from the OS.

The processes that constitute SNMP Agent depend on which OS the system uses. The following describes SNMP Agent's process configuration for each OS.


For HP-UX (IPF)

The following shows SNMP Agent's process configuration for HP-UX (IPF).

Figure 1–4: SNMP Agent's process configuration (for HP-UX (IPF))



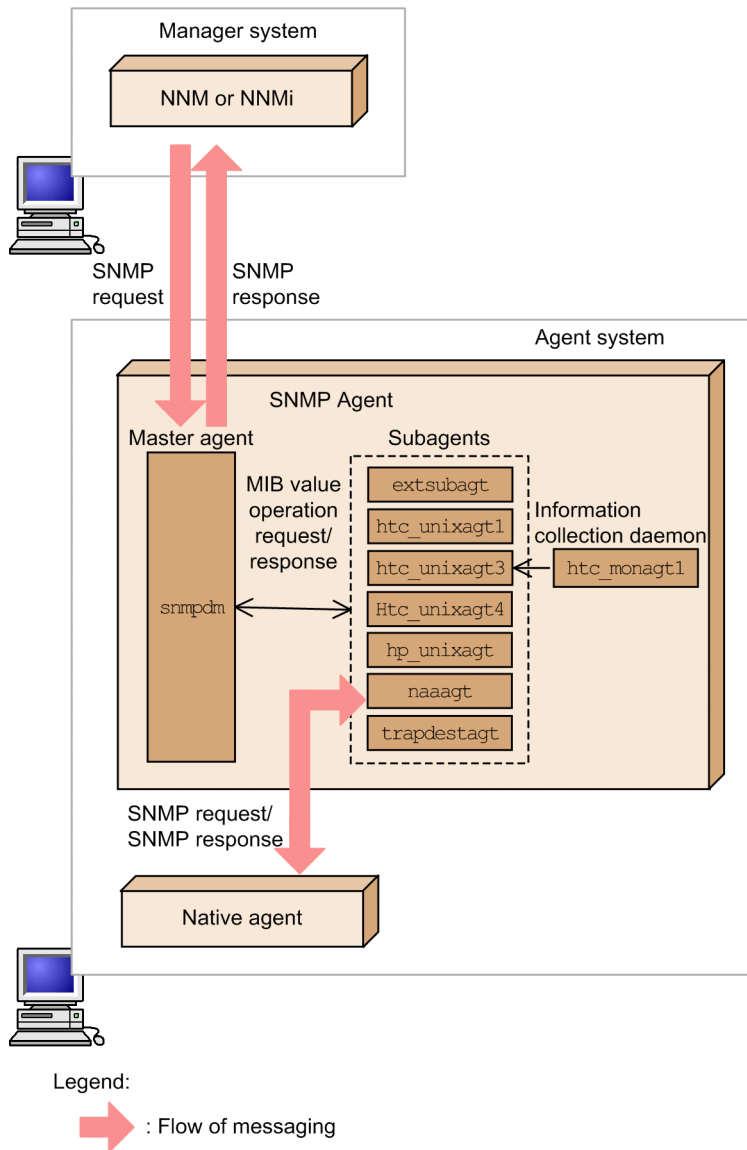
Legend:

 : Flow of messaging

For Solaris, AIX and Linux

The following shows SNMP Agent's process configuration for Solaris, AIX and Linux.

Figure 1–5: SNMP Agent's process configuration (for Solaris, AIX and Linux)



(1) Process performed at the master agent

The following process is performed at the master agent:

- `snmpdm`
 This process receives an SNMP request from the manager and sends the received message to the subagents. It also sends the subagents' responses to the manager.

(2) Processes performed at the subagents

The following processes are performed at the subagents:

- `extsubagt`
 Provides extended MIB objects
- `htc_unixagt1`
 Provides Hitachi enterprise-specific MIB objects. For details about the provided MIB objects, see *htc_unixagt1* in *Chapter 5. Commands and Processes*.

- `htc_unixagt2`
Provides Hitachi enterprise-specific MIB objects. For details about the provided MIB objects, see [htc_unixagt2](#) in *Chapter 5. Commands and Processes*.
- `htc_unixagt3`
Provides Hitachi enterprise-specific MIB objects. For details about the provided MIB objects, see [htc_unixagt3](#) in *Chapter 5. Commands and Processes*.
- `htc_unixagt4`
Provides Hitachi enterprise-specific MIB objects. For details about the provided MIB objects, see [htc_unixagt4](#) in *Chapter 5. Commands and Processes*.
- `hp_unixagt`
Provides Hewlett-Packard enterprise-specific MIB objects. For details about the provided MIB objects, see [hp_unixagt](#) in *Chapter 5. Commands and Processes*.
- `naaagt`
Provides the native agent adapter function.
- `trapdestagt`
Provides the trap group (`hp.nm.snmp.trapMIB`) of Hewlett-Packard enterprise-specific MIB objects.

The subagents' processes depend on the OS. The following table lists the subagents' processes that are provided by SNMP Agent and the supported OSs.

Table 1–6: Subagents' processes provided by SNMP Agent and the OSs

Subagent's process provided by SNMP Agent	OS			
	HP-UX (IPF)	Solaris	AIX	Linux
<code>extsubagt#1</code>	Y	Y	Y	Y
<code>htc_unixagt1</code>	Y	Y	Y	Y
<code>htc_unixagt2</code>	Y	N	N	N
<code>htc_unixagt3</code>	N	Y	Y	Y
<code>htc_unixagt4</code>	Y	Y	Y	Y
<code>hp_unixagt</code>	N	Y	Y	Y
<code>naaagt</code>	N	Y	Y	Y#2
<code>trapdestagt</code>	Y	Y	Y	Y

Legend:

Y: Provided

N: Not provided

#1

The `extsubagt` process is executed when the user configures an extended MIB definition file. Immediately after installation, there are no configured extended MIB definition files. Configure this file as required. For details about how to configure the extended MIB definition file, see [2.10.2 Configuring an extended MIB definition file](#). For details about how to configure multiple extended MIB definition files, see [2.10.10 Configuring more than one extended MIB definition file](#).

#2

When the native agent is starting, `naaagt` starts.

(3) Information collection daemon process

The following process functions as an information collection daemon:

- `htc_monagt1`
Provides CPU utilization-related information

Whether the `htc_monagt1` process is provided depends on the OS. The following table lists the OSs that support the `htc_monagt1` process.

Table 1–7: OSs that support the `htc_monagt1` process

Information collection daemon	OS			
	HP-UX (IPF)	Solaris	AIX	Linux
<code>htc_monagt1</code>	N	Y	Y	Y

Legend:

- Y: Provided
- N: Not provided

1.3.2 SNMP Agent startup processing

The SNMP Agent processes normally start automatically when the system starts, and they function as follows:

- At startup, the master agent reads the configuration file (`/etc/SnmpAgent.d/snmpd.conf`).
- Each subagent registers its own MIB with the master agent.

Each of the processes provided by SNMP Agent has startup options and a file for defining environment variables that enable you to customize processing. You can specify the process startup options in the command format. For details about the startup options, see *Processes* in *Chapter 5. Commands and Processes*.

1.3.3 Operations using SNMP Agent

The master agent receives all the SNMP requests that are sent by the manager. If the received SNMP request contains any MIB object that has been registered by a subagent, the master agent sends a MIB value operation request to that subagent. The subagent performs the specified MIB value operation and returns the execution result to the master agent as a MIB value operation response. The master agent then returns this execution result to the manager as an SNMP response.

Any errors that occur in the master agent or a subagent are logged in the `snmpd.logn` (*n*: log file number) file. Any errors that occur in the information collection daemon are output to the `htc_monagt1.log` file. You can specify the types of information to be logged in the log file. For details about acquiring log information, see *7.3 Collecting logs*.

1.4 SNMP Agent commands

SNMP Agent provides the following commands.

Table 1–8: List of commands provided by SNMP Agent

Classification	Command name	Function
Startup and termination of SNMP Agent	<code>snmpstart</code>	Starts SNMP Agent.
	<code>snmpstop</code>	Stops SNMP Agent.
Trap issuance	<code>snmptrap</code>	Issues an SNMP trap.
	<code>trapsend</code>	
	<code>systemtrap</code>	Issues a system trap.
Status list display	<code>snmpcheck</code>	Displays the operating status (running or not running) of the master agent and subagents.
	<code>snmpcmdchk</code>	Displays the installation status of the OS commands needed by SNMP Agent to generate MIB values.
Collection of failure information	<code>jplesalog.sh.def</code>	Collects system information from a machine on which a problem occurred.

Note:

In addition to the above commands, SNMP Agent provides commands for controlling SNMP Agent processes. For details about these processes, see [Chapter 5. Commands and Processes](#).

2

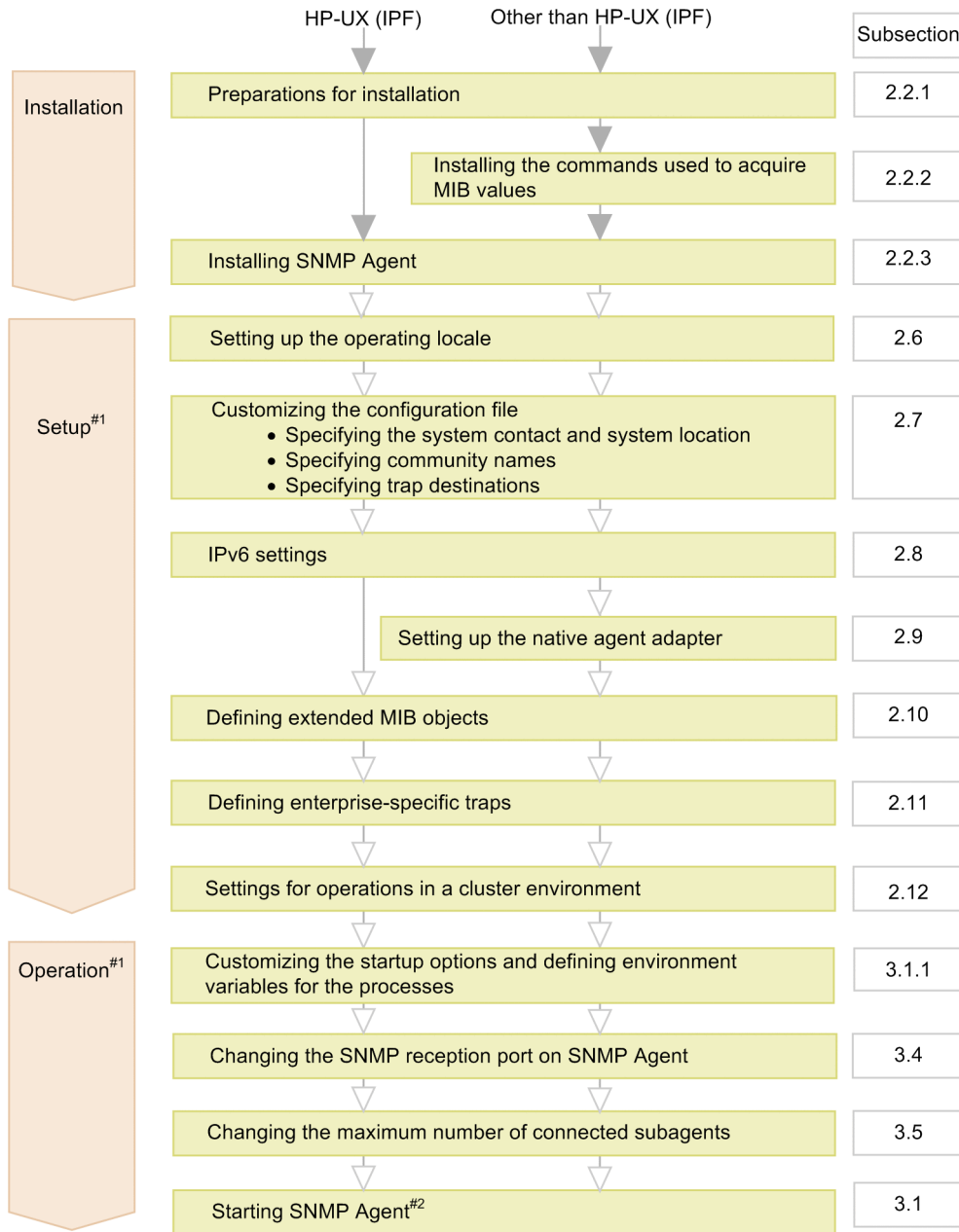
Setting up an Environment for SNMP Agent

This chapter describes SNMP Agent installation and customization of the environment settings.

2.1 Procedures from installation to setup

The following figure shows the procedures from SNMP Agent installation to setup.

Figure 2–1: Procedures from SNMP Agent installation to setup



Legend:

▼ : Required task

▽ : Optional task

#1: After SNMP Agent is installed, it runs with the default settings. Starting at the setup stage, customize SNMP Agent to match the environment you are using.

#2: SNMP Agent starts automatically when the system starts. If necessary, start SNMP Agent manually.

2.2 Installing

This subsection describes preparations for installation and the following installation procedures:

- Installing the commands used to acquire MIB values (for an OS other than HP-UX (IPF))
- Installing SNMP Agent

2.2.1 Preparations for installation

The following describes the preparations for installing SNMP Agent.

Procedure

1. Make sure that the prerequisite patches for SNMP Agent have been installed.

Problems can occur if the prerequisite patches are not installed, such as the return of an invalid MIB value as a response, or a MIB value acquisition error. For details about the prerequisite patches for SNMP Agent, see *D. List of Prerequisite Patches and Processes (Services) for SNMP Agent*.

2. Adjust the OS kernel parameter settings in order to allocate the resources that are required for SNMP Agent execution.

For details about the kernel parameters, see *C. List of Kernel Parameters*.

3. Make sure that the native agent has been configured.

A native agent is a required program for SNMP Agent. For details about installing a native agent, see *2.4.2 Notes about installation (for Solaris)* or *2.4.4 Notes about installation (for Linux)*.

2.2.2 Installing the commands used to acquire MIB values (for an OS other than HP-UX (IPF))

SNMP Agent uses OS commands to acquire some of the provided MIB values.

Install these commands before you install SNMP Agent. After you have installed SNMP Agent, make sure that these commands have been installed on the target machine. If the commands have not been installed, SNMP Agent will not be able to acquire MIB values or will return invalid MIB values.

To determine whether the commands have been installed, use the `snmpcmdchk` command.

Example:

The following example shows the results when running in Solaris.

```
#/opt/CM2/ESA/bin/snmpcmdchk
/usr/sbin/prtconf      installed.
/usr/bin/sar           installed.
/usr/sbin/swap         installed.
/usr/bin/pagesize     installed.
/usr/bin/mpstat        Not installed.
```

The following table lists the commands used by SNMP Agent to acquire MIB values for each OS.s

Table 2–1: Commands used by SNMP Agent to acquire MIB values

OS	Command used to acquire MIB values	Command usage
Solaris	/usr/bin/sar	sar 5 1 sar -r 5 sar -d 5 sar -d 300 1
	/usr/bin/pagesize	pagesize
	/usr/bin/mpstat	mpstat 300 2
	/usr/sbin/prtconf	prtconf
	/usr/sbin/swap	swap -s
AIX#1	/usr/bin/iostat	iostat -d
	/usr/sbin/lsdev	lsdev -Cc memory
	/usr/sbin/lsattr	lsattr -E
	/usr/sbin/lspcs	lspcs -a
	/usr/bin/ps	ps -e ps ug
	/usr/bin/uptime	uptime
	/usr/bin/vmstat	vmstat -f vmstat -s
	/usr/sbin/sar	sar -P ALL 300 1 sar -d 300 1
/usr/bin/svmon	svmon -G	
Linux#2	/usr/bin/vmstat	vmstat
	/bin/ps	ps -e
	/usr/bin/uptime	uptime
	/usr/bin/free	free
	/usr/bin/mpstat	mpstat -P ALL 300 1 mpstat 300 1

#1: Install the `svmon` command included in the fileset `bos.perf.tools`.

#2: The `mpstat` command is in the `sysstat` package when an SNMP agent is used in Linux.

2.2.3 Installing SNMP Agent

You can use either Hitachi Program Product Installer or JP1/Software Distribution to install SNMP Agent. If remote installation using JP1/Software Distribution fails, use Hitachi Program Product Installer to install SNMP Agent. For details about how to perform remote installation using JP1/Software Distribution, see the manual *Job Management Partner I/Software Distribution Manager Description and Administrator's Guide*.

This subsection describes how to install SNMP Agent using Hitachi Program Product Installer.

To install SNMP Agent, use the Hitachi Program Product Installer that is stored in the distribution media.

Use the following procedure to install SNMP Agent.

Procedure

1. Use `root` permissions to log in to the machine on which SNMP Agent is to be installed.
Before you use Hitachi Program Product Installer, either use `root` permissions to log in to the system or use the `su` command to change the user permissions to `root`.
2. Terminate all programs that are connected with SNMP Agent.
If JP1 product and other programs are connected with a SNMP Agent, stop those programs when performing overwrite installation.
If a program is running, the installation of SNMP Agent might get failed.
3. Run Hitachi Program Product Installer.
Install SNMP Agent according to Hitachi Program Product Installer's instructions. For details about how to use Hitachi Program Product Installer, see [2.2.4 How to use Hitachi Program Product Installer](#).

2.2.4 How to use Hitachi Program Product Installer

Hitachi Program Product Installer is stored in SNMP Agent distribution media. This subsection describes how to start Hitachi Program Product Installer and install SNMP Agent. For details about the Hitachi Program Product Installer, see the Release Notes in the distribution media.

(1) Starting Hitachi Program Product Installer

The following procedure shows how to start Hitachi Program Product Installer.

Procedure

1. Insert the provided SNMP Agent CD-ROM into the drive.
2. Mount the CD-ROM.
The mounting method depends on the OS, hardware, and environment in use. For details about the mounting method, see the OS documentation.
3. Install Hitachi Program Product Installer and then start it.
The directory and file names in the CD-ROM might be displayed differently depending on the system environment. Use the `ls` command to check the names and enter the correct file names.
4. Unmount the CD-ROM.
After installation is completed, unmount the CD-ROM. For details about the unmounting method, see the OS documentation.

(2) How to install SNMP Agent

This subsection describes how to install SNMP Agent using Hitachi Program Product Installer.

Execute the following command to start Hitachi Program Product Installer:

```
/etc/hitachi_setup
```

When Hitachi Program Product Installer starts, the initial window is displayed.

Figure 2–2: Example of the initial window of Hitachi Program Product Installer

```
L) List Installed Software.
I) Install Software.
D) Delete Software.
Q) Quit.

Select Procedure ==>

+-----+
CAUTION!
YOU SHALL INSTALL AND USE THE SOFTWARE PRODUCT LISTED IN THE
"List Installed Software." UNDER THE TERMS AND CONDITION OF
THE SOFTWARE LICENSE AGREEMENT ATTACHED TO SUCH SOFTWARE PRODUCT.
+-----+
```

In the initial window, enter **I** to display a list of the programs that can be installed. Move the cursor to the program you want to install and then press the space bar to select it. Enter **I** again to install SNMP Agent. After installation is completed, enter **Q** to return to the initial window.

(3) Removing SNMP Agent

Execute the following command to start Hitachi Program Product Installer:

```
/etc/hitachi_setup
```

The initial window of Hitachi Program Product Installer is displayed. For details about the initial window, see Figure 2-2.

In the initial window, enter **D** to display a list of the programs that can be removed. Move the cursor to the program you want to remove and then press the space bar to select it. Enter **D** again to remove the software. After the removal process is completed, enter **Q** to return to the initial window.

(4) Displaying the version information

Execute the following command to start Hitachi Program Product Installer:

```
/etc/hitachi_setup
```

The initial window of Hitachi Program Product Installer is displayed. For details about the initial window, see Figure 2-2.

In the initial window, enter **L** to display a list of the Hitachi products that have been installed.

2.3 Performing an upgrade installation

This subsection describes how to perform an upgrade installation of SNMP Agent.

2.3.1 Backing up the customized definition files

If you have directly customized the files provided by SNMP Agent, first back up the customized files and then perform an upgrade installation. For details about file backup, see [3.6 Backing up and restoring](#). During the upgrade installation of SNMP Agent, the files listed below are not overwritten, if they already exist. Therefore, there is no need to back up these files.

Table 2–2: List of files that are not overwritten during an upgrade installation

Type	Path name	File name	OS			
			HP-UX (IPF)	Solaris	AIX	Linux
Configuration files	/etc/SnmpAgent.d	snmpd.conf	N	N	N	N
	/etc/opt/OV/share/conf	snmpmib	N	N	--	N
	/usr/OV/conf	snmpmib	--	--	N	--
	/etc/opt/OV/share/conf	snmpmib.bin	N	N	--	N
	/usr/OV/conf	snmpmib.bin	--	--	N	--
	/etc/srconf/agt	naa.conf	--	N	N	N
	/etc/srconf/agt	snmpd.conf	N	N	N	N
	/etc/srconf/mgr	snmpinfo.dat	N	N	N	N
	/etc/srconf/mgr	mgr.conf	N	N	N	N
	/opt/CM2/ESA/ext	Files under the directory	N	N	--	N
	/usr/CM2/ESA/ext	Files under the directory	--	--	N	--
Environment variable definition files	/etc/rc.config.d	Files beginning with Snmp	--	N	--	--
	/opt/CM2/ESA/opt	Files under the directory	N	--	--	N
	/usr/CM2/ESA/opt	Files under the directory	--	--	N	--
Extended MIB definition file	/etc/SnmpAgent.d	snmpd.extend	N	N	N	N
Command	/opt/OV/bin	snmptrap	N	N	--	--
	/usr/OV/bin	snmptrap	--	--	N	--
File system definition file	/etc/SnmpAgent.d	esafilesys.conf	N	N	N	N

Type	Path name	File name	OS			
			HP-UX (IPF)	Solaris	AIX	Linux
Adding file system target definition file	/etc/SnmpAgent.d	esafstab.conf	--	--	--	N
Disk definition file	/etc/SnmpAgent.d	esadisk.conf	--	--	--	N
Operating locale definition file	/etc/SnmpAgent.d	esalocale.conf	N	N	N	N

Legend:

N: Not overwritten

--: Not applicable

2.3.2 Setting the SNMP trap transmission port number (when performing overwrite installation on version 07-10 or earlier)

If you are performing overwrite installation on SNMP Agent version 07-10 or earlier, and if you have specified a value other than 161/udp for the SNMP reception port, you must specify 162 for the SNMP trap transmission port number (SR_TRAP_TEST_PORT environment variable) after you have installed SNMP Agent.

Use the following procedure to set the SNMP trap transmission port number.

Procedure

1. Add the following two lines in the `SnmpMaster` file, which is used by the `snmpd` process:

```
SR_TRAP_TEST_PORT=162
export SR_TRAP_TEST_PORT
```

2. Use the `snmpstart` command to start SNMP Agent.

2.3.3 Setting the log output options (when performing overwrite installation on version 07-10 or earlier)

If you are performing overwrite installation on SNMP Agent version 07-10 or earlier, use one of the following methods to edit the environment variable definition file after you have finished overwrite installation, in order to acquire necessary log information:

- Directly edit the environment variable definition files.
- Copy the environment variable definition files from the backup installation files, and then edit those copies.

(1) Directly editing the environment variable definition files (for HP-UX (IPF))

Edit the environment variable definition files using the procedure described below. In the case of a comment line, remove the hash mark (#) at the beginning to enable the setting.

Procedure

1. Edit `SNMP_MASTER_OPTIONS` in the `SnmpMaster` file, which is used by the `snmpdm` process, as follows:

```
SNMP_MASTER_OPTIONS="-tcplocal -aperror -apwarn -apverbose -hexdump -
vbdump"
export SNMP_MASTER_OPTIONS
```

2. Edit `SNMP_EXTAGT_OPTIONS` in the `SnmpExtAgt` file, which is used by the `extsubagt` process, as follows:

```
SNMP_EXTAGT_OPTIONS="-aperror -apwarn -apverbose"
export SNMP_EXTAGT_OPTIONS
```

3. Edit `SNMP_HTCUNIX1_OPTIONS` in the `SnmpHtcunix1` file, which is used by the `htc_unixagt1` process, as follows:

```
SNMP_HTCUNIX1_OPTIONS="-aperror -apwarn -apverbose"
export SNMP_HTCUNIX1_OPTIONS
```

4. Edit `SNMP_HTCUNIX2_OPTIONS` in the `SnmpHtcunix2` file, which is used by the `htc_unixagt2` process, as follows:

```
SNMP_HTCUNIX2_OPTIONS="-aperror -apwarn -apverbose"
export SNMP_HTCUNIX2_OPTIONS
```

5. Edit `SNMP_TRAPDEST_OPTIONS` in the `SnmpTrpDst` file, which is used by the `trapdestagt` process, as follows:

```
SNMP_TRAPDEST_OPTIONS="-aperror -apwarn -apverbose"
export SNMP_TRAPDEST_OPTIONS
```

6. Use the `snmpstart` command to start SNMP Agent.

(2) Directly editing the environment variable definition files (for an OS other than HP-UX (IPF))

Procedure

1. Edit `SNMP_MASTER_OPTIONS` in the `SnmpMaster` file, which is used by the `snmpdm` process, as follows:

```
SNMP_MASTER_OPTIONS="-tcplocal -aperror -apwarn -apverbose -hexdump -
vbdump"
export SNMP_MASTER_OPTIONS
```

2. Edit `SNMP_EXTAGT_OPTIONS` in the `SnmpExtAgt` file, which is used by the `extsubagt` process, as follows:

```
SNMP_EXTAGT_OPTIONS="-aperror -apwarn -apverbose"  
export SNMP_EXTAGT_OPTIONS
```

3. Edit `SNMP_HTCUNIX1_OPTIONS` in the `SnmPhtcunix1` file, which is used by the `htc_unixagt1` process, as follows:

```
SNMP_HTCUNIX1_OPTIONS="-aperror -apwarn -apverbose"  
export SNMP_HTCUNIX1_OPTIONS
```

4. Edit `SNMP_HTCUNIX3_OPTIONS` in the `SnmPhtcunix3` file, which is used by the `htc_unixagt3` process, as follows:

```
SNMP_HTCUNIX3_OPTIONS="-aperror -apwarn -apverbose"  
export SNMP_HTCUNIX3_OPTIONS
```

5. Edit `SNMP_HPUNIX_OPTIONS` in the `SnmP hpunix` file, which is used by the `hp_unixagt` process, as follows:

```
SNMP_HPUNIX_OPTIONS="-aperror -apwarn -apverbose"  
export SNMP_HPUNIX_OPTIONS
```

6. Edit `SNMP_NAA_OPTIONS` in the `SnmPnaa` file, which is used by the `naaagt` process, as follows:

In Solaris and AIX:

```
SNMP_NAA_OPTIONS="-aperror -apwarn -apverbose -hexdump -vbdump"  
export SNMP_NAA_OPTIONS
```

In Linux:

Do not delete the space between `-vbdump` and the double quote(`"`).

```
SNMP_NAA_OPTIONS="-aperror -apwarn -apverbose -hexdump -vbdump  
"$SNMP_NAA_OPTIONS  
export SNMP_NAA_OPTIONS
```

7. Edit `SNMP_TRAPDEST_OPTIONS` in the `SnmPTrpDst` file, which is used by the `trapdestagt` process, as follows:

```
SNMP_TRAPDEST_OPTIONS="-aperror -apwarn -apverbose"  
export SNMP_TRAPDEST_OPTIONS
```

8. Use the `snmpstart` command to start SNMP Agent.

(3) Editing the environment variable definition files by using the backup installation files

Use the following procedure to edit the environment variable definition files.

Procedure

1. Back up the old version's environment variable definition files to a desired location.
Back up these files to a directory other than the directory in which the environment variable definition files are stored.
For details about backing up settings files, see [3.6 Backing up and restoring](#).

2. Copy the environment variable definition files from the backup installation files to the path specified for the environment variable definition files.
3. Compare the copies of the environment variable definition files created in step 2 with the environment variable definition files backed up in step 1. If there is any difference, harmonize the environment variable definition files that you copied in step 2 with the environment variable definition files that you backed up in step 1.
4. Use the `snmpstart` command to start SNMP Agent.

For details about the path to the environment variable definition files and the installation backup environment variable definition files, see [A. SNMP Agent Files](#).

2.3.4 Specifying IPv6 trap destinations (when performing overwrite installation on version 09-00 or earlier)

If you are performing overwrite installation on SNMP Agent version 09-00 or earlier, so that SNMP Agent can issue SNMP traps over IPv6, edit the configuration file `snmpd.cnf` as follows after performing overwrite installation:

Procedure

1. Back up the configuration file `snmpd.cnf` to the location of your choice.
2. Copy the configuration file `snmpd.cnf` from the backup installation files to the configuration file path (`/etc/srconf/agt`).
3. Compare the copies of the configuration file `snmpd.cnf` copied in step 2 with the configuration file `snmpd.cnf` backed up in step 1. If there is any difference, synchronize the copied configuration file `snmpd.cnf` with the backed up configuration file `snmpd.cnf`.
4. Set the IPv6 trap destinations.
5. Use the `snmpstart` command to start SNMP Agent.
For Solaris and AIX, if you do not want to stop the native agent, execute the `snmpstart` command with the `-n` option.

For details about the path to the configuration file `snmpd.cnf`, and the path to the backup installation files copy of the configuration file `snmpd.cnf`, see [A. SNMP Agent Files](#).

For details about setting IPv6 trap destinations, see [2.8 IPv6 settings](#).

2.4 Notes about installation

This section provides notes about SNMP Agent installation that are applicable to all OSs. For OS-specific notes, see the relevant subsections.

- Before you install SNMP Agent, install the prerequisite programs.
If you install SNMP Agent and NNM on the same system, first install NNM and then install SNMP Agent.

Important

If you upgrade your OS by overwriting or if you upgrade AIX by migration installation, uninstall SNMP Agent beforehand. After you have upgraded the OS, re-install the application area. If you have customized SNMP Agent, you need to specify the settings again.

2.4.1 Notes about installation (for HP-UX (IPF))

This subsection provides HP-UX (IPF)-specific notes about installation when SNMP Agent for HP-UX (IPF) system is installed. For notes common to all OSs, see [2.4 Notes about installation](#).

- When SNMP Agent is installed, some of the OS files are changed.
SNMP Agent uses the 161/udp port as the SNMP packet reception port. Only one process can connect to this 161/udp port. If both the native agent and SNMP Agent are started, the native agent is set to not start automatically during OS startup so that SNMP Agent can always connect to the 161/udp port. If SNMP Agent is uninstalled, the original names are restored. The following shows the file names before and after the change.

Before change	After change
/sbin/rc1.d/K440SnmpMaster	/sbin/rc1.d/xK440SnmpMaster
/sbin/rc1.d/K435SnmpHpunix	/sbin/rc1.d/xK435SnmpHpunix
/sbin/rc1.d/K435SnmpMib2	/sbin/rc1.d/xK435SnmpMib2
/sbin/rc1.d/K435SnmpTrpDst	/sbin/rc1.d/xK435SnmpTrpDst
/sbin/rc1.d/K435SnmpIpv6	/sbin/rc1.d/xK435SnmpIpv6
/sbin/rc2.d/S560SnmpMaster	/sbin/rc2.d/xS560SnmpMaster
/sbin/rc2.d/S565SnmpHpunix	/sbin/rc2.d/xS565SnmpHpunix
/sbin/rc2.d/S565SnmpMib2	/sbin/rc2.d/xS565SnmpMib2
/sbin/rc2.d/S565SnmpTrpDst	/sbin/rc2.d/xS565SnmpTrpDst
/sbin/rc2.d/S565SnmpIpv6	/sbin/rc2.d/xS565SnmpIpv6

If snmp 161/udp is not defined in the `/etc/services` file, it is added to the `/etc/services` file.

- If you install SNMP Agent on the same host as for NNM, make sure that the required directories exist.
If you install SNMP Agent on the same host as for NNM, the directories shown below must exist when SNMP Agent is installed. Take precautions especially in the case of a cluster system.
 - `/var/opt/OV/share`
 - `/etc/opt/OV/share`

SNMP Agent creates files under these directories during installation. If these directories do not exist, installation fails.

- During SNMP Agent installation, the following files are replaced with files provided by SNMP Agent:

- `/etc/srconf/agt/snmpd.cnf`
- `/etc/srconf/mgr/mgr.cnf`
- `/etc/srconf/mgr/snmpinfo.dat`

If any of these files already exist when SNMP Agent is installed, they are saved in the SNMP Agent installation directory during installation of SNMP Agent (under `/opt/CM2/ESA/newconfig`). These saved files are restored if SNMP Agent is uninstalled.

2.4.2 Notes about installation (for Solaris)

This subsection provides Solaris-specific notes about installation when SNMP Agent for a Solaris system is installed. For installation notes common to all OSs, see [2.4 Notes about installation](#).

- If you install SNMP Agent on the same host as for NNM, the directories listed below must exist. Take precautions especially in the case of a cluster system.

- `/var/opt/OV/share`
- `/etc/opt/OV/share`

SNMP Agent creates files under these directories during installation. If these directories do not exist, installation fails.

- Make sure that the Solaris native agent has been installed. If it is not installed, install it.

If the Solaris native agent has been installed, SNMP Agent acquires MIB-II's interfaces, `at`, `ip`, `icmp`, `tcp`, and `udp` group information from the Solaris native agent. If the Solaris native agent has not been installed, SNMP Agent does not respond regarding MIB-II's interfaces, `at`, `ip`, `icmp`, `tcp`, and `udp` group information, or it returns invalid information.

Normally, the Solaris native agent is installed when Solaris is installed. If you are not sure whether the native agent was installed in the currently running system, check as necessary.

The native agent consists of the following packages:

- `system/management/snmp/net-snmp`
- `system/management/snmp/net-snmp/addons`

To determine whether these packages have been installed, use the following command:

- `/usr/bin/pkg info system/management/snmp/net-snmp`
- `/usr/bin/pkg info system/management/snmp/net-snmp/addons`

- When SNMP Agent is installed, the following OS files are changed:

`/lib/svc/method/svc-net-snmp`

If `snmp 161/udp` is not defined in the `/etc/services` file, this information is added to the `/etc/services` file.

- When SNMP Agent is installed, the start/stop script of the following Solaris native agent is changed:

The OS file (`/lib/svc/method/svc-net-snmp`) can be overwritten. If the `/lib/svc/method/svc-net-snmp` file has been customized, or if the `snmpd` process will not start up correctly after SNMP Agent is installed, edit the file `/lib/svc/method/svc-net-snmp` as needed. Note that the `snmpd` process option `udp:8161` is required for the native agent adapter to work correctly, so check that it has not been deleted. Also, if SNMP Agent is deleted, the `/lib/svc/method/svc-net-snmp` file will be restored to the state it was in prior to installation of SNMP Agent. In this case, too, edit the `/lib/svc/method/svc-net-snmp` file as necessary.

2.4.3 Notes about installation (for AIX)

This subsection provides AIX-specific notes about installation when SNMP Agent for AIX systems is installed. For installation notes common to all OSs, see [2.4 Notes about installation](#).

- When SNMP Agent is installed, the following files are changed:
 - `/etc/rc.tcpip`
 - `/etc/inittab`
 - `/etc/rc.shutdown`

If `snmp 161/udp` is not defined in the `/etc/services` file, it is added to the `/etc/services` file.

2.4.4 Notes about installation (for Linux)

This subsection provides Linux-specific notes about installation when SNMP Agent for Linux systems is installed. For installation notes common to all OSs, see [2.4 Notes about installation](#).

- Some OS files are changed when an SNMP agent is installed. If `snmp 161/udp` is not defined in the file `/etc/services`, it is added to the file `/etc/services`.
- Confirm that the Linux native agent is installed.

If the Linux native agent is not installed, install it as follows:

1. To confirm whether the native agent is installed, execute the following command as a superuser:

```
#rpm -qa | grep net-snmp
```

If the results below are shown, the native agent is installed. If the results below are not shown, then the native agent is not installed. If that is the case, install the native agent by using the `rpm` command.

In RHEL 6, CentOS 6, and Oracle Linux 6:

```
net-snmp-x.x.x.x
net-snmp-libs-x.x.x-x
net-snmp-utils-x.x.x-x
```

In RHEL 7, CentOS 7, and Oracle Linux 7:

```
net-snmp-x.x.x.x
net-snmp-libs-x.x.x-x
net-snmp-utils-x.x.x-x
```

In SUSE Linux 12:

```
net-snmp-x.x.x.x
```

2. In RHEL 6, CentOS 6, and Oracle Linux 6, check the OS auto-start files.

Confirm that the OS auto-start files in `/etc/rc.d/rc3.d`, `/etc/rc.d/rc4.d`, and `/etc/rc.d/rc5.d` have symbolic links attached to them from the file `/etc/rc.d/init.d/snmpd`. Symbolic links are usually attached to the `S50snmpd` file in the above directories. If no symbolic links exist, configure auto-start setting so that `snmpd` starts at run levels 2, 3 and 5 when OS starts.

3. In RHEL 7, CentOS 7, Oracle Linux 7, SUSE Linux 12, execute the following command to confirm whether `snmpd` service is enabled.

```
systemctl is-enabled snmpd.service
```

If "enabled" is displayed as a result, `snmpd` service is enabled. If `snmpd` service is not enabled, execute the following command to enable `snmpd` service.

```
systemctl enable snmpd.service
```

4. In RHEL 6, 7, CentOS 6, 7, and Oracle Linux 6, 7, Change the configuration of the native agent.

Under the default setting, the native agent responds to only MIBs for the system group. Change the file `/etc/snmp/snmpd.conf` as follows, so that all MIB groups are responded to.

<Before change>

```
view systemview included .1.3.6.1.2.1.1
```

```
view systemview included .1.3.6.1.2.1.25.1.1
```

<After change>

```
view systemview included .1.3
```

5. Start the SNMP agent.

To start the SNMP agent, reboot the machine or execute the following procedure as a superuser:

In RHEL 6, CentOS 6, and Oracle Linux 6:

```
/opt/CM2/ESA/bin/snmpstop (#)
```

```
/etc/rc.d/init.d/snmpd restart
```

```
/opt/CM2/ESA/bin/snmpstart
```

In RHEL 7, CentOS 7, Oracle Linux 7, and SUSE Linux 12:

```
/opt/CM2/ESA/bin/snmpstop (#)
```

```
systemctl stop snmpd.service
```

```
systemctl start snmpd.service
```

```
/opt/CM2/ESA/bin/snmpstart
```

#: If the SNMP agent is running, stop it.

- Define the node in the Linux file `/etc/hosts`.

The SNMP agent asynchronously sends events that have occurred to the manager by using SNMP trap messages. Each message contains the IP address of the host from which it is sent. Each IP address corresponds with a particular host name.

When Linux is installed, the IP address of the node in the file `/etc/hosts` might be `xxx.0.0.1`. The following is an example of how to define a host name (`linux01`) in the file `/etc/hosts`:

```
127.0.0.1    linux01 localhost.localdomain localhost
```

In the definition of `/etc/hosts` in the above example, the IP address corresponding to `linux01` is `127.0.0.1`.

In the `/etc/hosts` file, specify the local IP address, instead of `127.0.0.1`, for the local node.

The following is an example of such a definition in the file `/etc/hosts`:

```
127.0.0.1          localhost.localdomain localhost
172.16.49.18      linux01
```


2.5 Uninstalling

This subsection describes how to uninstall SNMP Agent and provides notes.

2.5.1 How to uninstall SNMP Agent

Use the following procedure to uninstall SNMP Agent.

Procedure

1. Terminate the program.
2. Back up the user files.
During uninstallation of SNMP Agent, definition files and log files are also deleted by directory. Back them up if necessary.
3. Execute Hitachi Program Product Installer.
Uninstall SNMP Agent according to the instructions of Hitachi Program Product Installer. For details about how to execute Hitachi Program Product Installer, see [2.2.4 How to use Hitachi Program Product Installer](#).

2.5.2 Notes about uninstallation

Notes follow about uninstalling SNMP Agent.

- Deleting unneeded files after uninstallation
When SNMP Agent is uninstalled, the files listed below are not deleted. If these files are not needed, delete them after you have uninstalled SNMP Agent.
 - `/etc/SnmpAgent.d/snmpd.conf`
 - `/etc/SnmpAgent.d/snmpd.extend`
 - `/etc/SnmpAgent.d/esafilesys.conf`
 - `/etc/SnmpAgent.d/esafilesys.conf.err`
 - `/etc/SnmpAgent.d/esafstab.conf` (for Linux)
 - `/etc/SnmpAgent.d/esafstab.conf.err` (for Linux)
 - `/etc/SnmpAgent.d/esadisk.conf` (for Linux)
 - `/etc/SnmpAgent.d/esadisk.conf.err` (for Linux)
 - `/etc/SnmpAgent.d/esalocale.conf`
 - User-specified extended MIB definition file
 - User-specified log files
 - Files under `/opt/CM2/ESA/ext` (for an OS other than AIX)
 - Files under `/usr/CM2/ESA/ext` (for AIX)
 - `/tmp/esa.log`
 - Files under `/tmp/.AgentSockets/`

- `/etc/snmpd.conf` (symbolic link)

2.6 Setting up the operating locale

SNMP Agent runs by internally specifying C for the LANG environment variable. Therefore, when you set up the system language environment, if you want to specify a value other than C for a locale environment variable that is higher than the LANG environment variable, set up the operating locale for SNMP Agent before you use it. In addition, when performing the new installation of later version of 11-00, setting is set by default. So, the setting is unnecessary.

The following describes the setup procedure.

Procedure

1. Stop Extensible SNMP Agent.

Execute `/opt/CM2/ESA/bin/snmpstop`.

2. Edit the file `/etc/SnmpAgent.d/esalocale.conf` as follows.

Before change:

```
#LC_ALL=C
#export LC_ALL
LANG=C
export LANG
```

After change:

```
LC_ALL=C
export LC_ALL
LANG=C
export LANG
```

3. Perform the following step only if you are using this product by overwriting version 09-00-01 or earlier:

Add the single line shown below to the beginning of the environment variable definition file (the file that begins with `Snmp` under the `/opt/CM2/ESA/opt` directory).

If the following line has already been added, proceed to step 4:

The line to be added to the beginning of the file:

```
. /etc/SnmpAgent.d/esalocale.conf
```

Important

The dot at the beginning of the line to be added must be entered, as well as the single-byte space following the dot.

4. Start Extensible SNMP Agent.

- Execute `/opt/CM2/ESA/bin/snmpstart`.

Important

When you set up the system language environment, if you specified a value other than C for a locale environment variable that is higher than the LANG environment variable, but no operating locale for SNMP Agent is set up, SNMP Agent might not run correctly. (For example, it might not be able to acquire MIB values.)

2.7 Customizing the configuration file (/etc/SnmpAgent.d/snmpd.conf)

The configuration file (/etc/SnmpAgent.d/snmpd.conf) is used to define an environment for SNMP Agent. If this file does not exist, SNMP Agent will not run.

In the configuration file (/etc/SnmpAgent.d/snmpd.conf), specify the following information:

- System contact and system location
- Community name
- Trap destinations

SNMP Agent provides the configuration file (/etc/SnmpAgent.d/snmpd.conf). This file already contains definition information. Customize the file, if necessary.

For details about the contents of the configuration file (/etc/SnmpAgent.d/snmpd.conf), see [2.7.7 Format of the configuration file](#).

2.7.1 Specifying the system contact and system location

The *system contact* is the name of the system's administrative contact, or information on how to contact the system administrator. The *system location* is a description of the physical location of the system.

You can set the system contact and system location with either of the following methods:

- Using the configuration file (/etc/SnmpAgent.d/snmpd.conf)
- Using the options of the snmpd process

If both methods are used, the system uses the option values of the snmpd process.

(1) Using the configuration file (/etc/SnmpAgent.d/snmpd.conf)

Use the following procedure to edit the configuration file (/etc/SnmpAgent.d/snmpd.conf) and to set the system contact and system location.

Procedure

1. Search the configuration file for the following two lines:

```
#contact:      # enter contact person for agent
#location:     # enter location of agent
```

These lines are located near the end of the configuration file.

2. Delete the hash mark (#) preceding the `contact:` label and delete the comment (preceded by the hash mark (#)). Similarly, delete the hash mark (#) preceding the `location:` label and delete the comment.
3. After the `contact:` label, enter the name of the person in charge of the system, expressed as a string of ASCII characters. The maximum length of the system contact is 255 characters. In this character string, include information about how to contact the person.
4. After the `location:` label, enter the location of the system, expressed as a string of ASCII characters.

The maximum length of the system location is 255 characters.

Example:

```
contact: Bob Jones (Phone 555-2000)
location: 1st Floor near Mens Room
```

(2) Using the options of the `snmpdm` process

Specify the system contact and system location in the options and then start the `snmpdm` process. For details about the `snmpdm` process, see *snmpdm* in *Chapter 5. Commands and Processes*.

Example:

The following shows an example for AIX:

```
/usr/sbin/snmpdm -C system-contact -L system-location
```

2.7.2 Specifying community names

A *community name* is a password required in order to access MIB values using the SNMP protocol. Community names have a low security level and are openly used in the network.

SNMP Agent community names can be used as follows:

- Specify a *get* community name to suppress referencing of the MIB values for SNMP requests from an unauthorized manager.
- Specify a *set* community name to suppress updating of the MIB values for SNMP requests from an unauthorized manager.

To specify the community names of the manager and SNMP Agent, use the configuration file (`/etc/SnmpAgent.d/snmpd.conf`).

(1) Types of community names

There are two types of community names: *get* community names and *set* community names. By using these community names appropriately, you can determine the type(s) of SNMP requests to which SNMP Agent will respond.

get community name

This is a password for `GetRequests`.

If you use a *get* community name, SNMP Agent will respond only to `GetRequests`.

set community name

This is a password for both `GetRequest` and `SetRequests`.

If you use a *set* community name, SNMP Agent will respond to both `GetRequests` and `SetRequests`.

You can register multiple community names.

2.7.3 How to register community names

When SNMP Agent is installed, `public` is set in both *get* and *set* community names.

To change the default *get* community name:

1. Register a *get* community name.
2. Register a *set* community name.
3. Store the community names in the manager.
4. Specify the community names in the native agent (applicable to Solaris).

(1) Registering a *get* community name

The following procedure shows how to register a *get* community name.

Procedure

1. Search the configuration file (`/etc/SnmpAgent.d/snmpd.conf`) for the following line:

```
get-community-name:    public
```

For details about the `snmpd.conf` file, see *Configuration file (snmpd.conf)* in *Chapter 6. Definition Files*.

The `get-community-name: label` is located near the end of the file.

2. Change the *get* community name.

`public` is set by default. To change this value, delete `public` and enter a desired *get* community name for SNMP Agent, expressed as a string of ASCII characters. To specify multiple *get* community names, add as many lines as needed.

For details about how to specify community names, see *2.7.4 How to specify community names*.

Example:

```
get-community-name: public
get-community-name: private
```

(2) Registering a *set* community name

The following procedure shows how to register a *set* community name.

Procedure

1. Search the configuration file (`/etc/SnmpAgent.d/snmpd.conf`) for the following line:

```
#set-community-name:    # enter community name
```

For details about the `snmpd.conf` file, see *Configuration file (snmpd.conf)* in *Chapter 6. Definition Files*.

The `#set-community-name: label` is located near the end of the file.

2. Add a *set* community name.

This name is provided as a comment line by default. Delete the hash mark (`#`) preceding the `set-community-name: label` and the comment (consisting of the second hash mark (`#`) and what follows it), and then enter a desired *set* community name for SNMP Agent, expressed as a string of ASCII characters. To specify multiple *set* community names, add as many lines as needed.

For details about how to specify community names, see *2.7.4 How to specify community names*.

Example:

```
set-community-name: private
set-community-name: point
```

! Important

- Specifying community names

If you use the same name for both *get* and *set* community names, specify the name only in the `set-community-name: label`. To use different names for *get* and *set* community names, specify appropriate names in the `get-community-name:` and `set-community-name:` labels.

- Suppressing the transmission of authentication failure traps

To suppress the transmission of authentication failure traps after you have set community names for SNMP Agent, set 2 in `snmpEnableAuthenTraps` in the configuration file (`/etc/srconf/agt/snmpd.cnf`), and then restart SNMP Agent.

For details about the transmission of authentication failure traps, see [2.7.5 Sending authentication failure traps](#).

(3) Storing community names in the manager

Store community names in the manager so that the manager's applications can access MIBs using the correct community names of each SNMP Agent.

(4) Specifying the community name of the native agent (for Solaris)

If SNMP Agent is running on a Solaris system, the community name of the native agent must be set.

The following procedure shows how to specify the community name of the native agent.

Procedure

1. Make sure that the community name of the native agent has been specified.

Make sure that the following line is specified in the configuration file (`/etc/net-snmp/snmp/snmpd.conf`):

```
rocommunity public
```

2. If these lines are not defined, add the community name of the native agent.
Add the line from step 1.

2.7.4 How to specify community names

This subsection describes how to specify community names.

(1) get community name

Specify a *get* community name in the following line in the configuration file (`/etc/SnmpAgent.d/snmpd.conf`):

```
get-community-name: get-community-name options#
```

#: For details about options, see [2.7.4\(3\) Options](#).

- If you specify a *get* community name, a request from any name other than the specified community name will result in an authentication failure.
- If no *get* community name is specified, SNMP Agent does not respond to a *GetRequest*. However, if a *set* community name is specified, SNMP Agent responds to a *GetRequest* that uses the *set* community name.
- If you specify a *get* community name, make sure that there is a single-byte space immediately after the colon (:).
- If you set multiple *get* community names in the configuration file (`/etc/SnmpAgent.d/snmpd.conf`), SNMP Agent can respond to multiple *get* community names.

(2) set community name

Specify a *set* community name in the following line in the configuration file (`/etc/SnmpAgent.d/snmpd.conf`):

```
set-community-name: set-community-name options#
```

#: For details about options, see [2.7.4\(3\) Options](#).

- If no *set* community name is specified, SNMP Agent does not respond to a *SetRequest*.
- If you specify a *set* community name, make sure that there is a single-byte space immediately after the colon (:).
- To enable the manager to set MIB values, you must specify a *set* community name. The manager uses the registered *set* community name to set MIB values.
- You can configure SNMP Agent to respond to multiple *set* community names.

Important

- To specify the same name for both *GetRequests* and *SetRequests*, specify only the `set-community-name: label`.
- Because it is reserved for use by SNMP Agent, you cannot specify the community name `sendtrap` as the `get-community-name: label` or `set-community-name: label` in the configuration file (`/etc/SnmpAgent.d/snmpd.conf`).

(3) Options

The available options are `IP:` and `VIEW:`

If you omit both options, the community name permits access requests from any IP address. In addition, you can access any MIB supported by SNMP Agent.

`IP:`

The community name specified in the SNMP request restricts the IP addresses that can access MIBs. Specify each IP address that can access MIBs separated by a space. No host name is allowed. Place at least one space between the community name and `IP:` and between `IP:` and the IP address.

Example:

```
get-community-name: public IP: 172.16.45.17 172.16.45.18
```

If the community name specified in an SNMP request is `public`, SNMP Agent will respond to the SNMP request as long as the request comes from `172.16.45.17` or `172.16.45.18`.

VIEW:

The specified community name restricts accessible MIBs. Specify object IDs representing accessible subtrees (1.3.6.1.2.1 for `mib-2`, for example), separated by a space. If you add a hyphen (-) before an object ID, the subtree represented by the object ID will be inaccessible. Place at least one space between the community name and `VIEW:`, and between `VIEW:` and the object ID. Also, place one space before a hyphen (-).

Example:

```
get-community-name: public VIEW: 1.3.6.1.2.1 -1.3.6.1.2.1.1
```

If the community name specified in an SNMP request is `public`, SNMP Agent will permit access to MIBs under 1.3.6.1.2.1 with the exception of 1.3.6.1.2.1.1.

Important

Specifying both `IP:` and `VIEW:`

If you specify both `IP:` and `VIEW:`, specify `IP:` before `VIEW:`. Specify both on one line and do not place a linefeed between the two. You can also specify `IP:` and `VIEW:` for `set-community-name`.

2.7.5 Sending authentication failure traps

An authentication failure results if the manager sends an incorrect or invalid community name to SNMP Agent. When SNMP Agent receives an incorrect or invalid community name, it sends an authentication failure trap to the manager.

If no `get` community name is specified for `GetRequests`, SNMP Agent does not respond to any community name, but sends an authentication failure trap.

2.7.6 Specifying trap destinations

SNMP Agent has trap destinations. A *trap destination* determines the destination of SNMP traps and identifies a manager system that is to receive SNMP Agent's traps. If there are multiple managers that manage an SNMP Agent instance, that SNMP Agent instance has multiple trap destinations. Trap destinations are configured with IPv4 addresses. For details about setting IPv6 trap destinations, see [2.8 IPv6 settings](#).

This subsection describes trap destination settings when the manager product that manages SNMP Agent is NNM and when the trap destinations are NNMi and any manager.

(1) For NNM

NNM allows the `netmon` process to add SNMP Agent trap destinations automatically using SNMP. However, when you use HP NNM version 6 or later as the SNMP Agent manager, no trap destination is set because SNMP Agent's `set` community name does not match the NNM's default `set` community name. If you want the `netmon` process to add SNMP Agent trap destinations automatically using SNMP, make sure that the `set` community name of SNMP Agent matches the `set` community name of HP NNM. For details about how to register community names, see [2.7.2 Specifying community names](#).

If a node is removed from the management target or is deleted from the map, the `netmon` process of NNM automatically deletes that node from the list of SNMP Agent trap destinations in the configuration file (`/etc/SnmpAgent.d/snmpd.conf`).

(2) For NNMi and any manager

If you want to set up SNMP Agent to send traps to NNMi and any manager, you must set the trap destinations. You can achieve this by editing SNMP Agent trap destinations in the configuration file (`/etc/SnmpAgent.d/snmpd.conf`).

The following procedure shows how to set up SNMP Agent to send traps to NNMi and any manager.

Procedure

1. Search the configuration file (`/etc/SnmpAgent.d/snmpd.conf`) for the following line:

```
#trap-dest: # enter trap destination
```

The `#trap-dest:` label is located near the end of the file.

2. Delete the hash mark (`#`) preceding the `trap-dest:` label and the comment (consisting of the second hash mark (`#`) and what follows it).
3. After the `trap-dest:` label, enter the host name or IP address of the manager to which SNMP Agent is to send traps.

Example:

```
trap-dest: 15.2.113.223
```

4. To add more trap destinations, add as many `trap-dest:` lines as needed.

2.7.7 Format of the configuration file

The following shows the contents of the provided configuration file (`/etc/SnmpAgent.d/snmpd.conf`).

```
#
# Use this file (snmpd.conf) to configure the following SNMP
# Agent parameters. The valid configuration keywords are:
#
#   get-community-name:
#   set-community-name:
#   trap-dest:
#   contact:
#   location:
#
# get-community-name: <name> IP: <ip_address_list> VIEW:
#   <view_list>
#
#   The agent will only respond to get requests using <name> as
#   the community name. Embedded blanks are not allowed.
#   If a community name is not specified in either this file or
#   snmpd.cnf, the agent not respond to any get requests.
#   More than one community name can be configured for the agent
#   by adding a separate entry for each name to be allowed. For
#   example,
#
#       get-community-name: secret
#
```

```

# restricts access to only those requests using
# community "secret"; and,
#
#     get-community-name: secret
#     get-community-name: private
#
# restricts access to only those requests using either
# community "secret" or "private".
#
# The IP: and VIEW: qualifiers are optional. When either or
# both qualifiers are omitted, the community name is allowed
# for any requesting IP address and provides access the entire
# MIB supported by the agent, respectively.
#
# The IP: qualifier further restricts use of the community name
# to only those requests that originate from one of the listed
# IP addresses. Host names are not supported. For example,
#
#     get-community-name: operator IP: 15.2.112.90 15.2.114.101
#
# only allows access using community "operator" from IP address
# 15.2.112.90 or 15.2.114.101.
#
# The VIEW: qualifier further restricts access using the
# community name to the sub-set of the agent's supported MIB
# identified by the space list of "MIB view sub-trees".
# A view sub-tree may be either the object identifier
# (1.3.6.1.2.1.1) or object name (system) of the MIB sub-tree
# to be included. The '-' character may be used to exclude an
# oid/name from the view. For example,
#     get-community-name: operator VIEW: 1.3.6.1.2.1 1.3.6.1.
4.1.11 -1.3.6.1.2.1.1
#
# allows access using community "operator" to all MIB objects
# under "mib-2" except those objects under "system", plus all
# objects under the "hp" sub-tree.
#
# For example,
#
#     get-community-name: operator IP: 15.2.112.90 15.2.114.101
#     VIEW: 1.3.6.1.2.1 1.3.6.1.4.1
#
# combines the access restrictions described in the previous
# examples for community "operator".
#
# set-community-name: <name> IP: <ip_address_list> VIEW:
# <view_list>
#
# The agent will only process get or set requests using <name>
# as the community name. Embedded blanks are not allowed.
# If a set community name is not configured in either this file
# or snmpd.cnf, set requests are not allowed by the agent. More
# than one set community name can be configured by adding a
# separate entry for each name to be allowed. For example,
#
#     set-community-name: control
#
# enables the agent to process set requests using the community

```

```

# name "control".
#
# The IP: and VIEW: qualifiers are optional. They provide the
# same access restrictions for the set community name as
# described above for get-community-name. For example,
#
#     set-community-name: technician VIEW: system
#     set-community-name: administrator IP: 15.2.112.90
#
# allows set request processing using community "technician" to
# only those objects under the "system" group
# (e.g., sysContact);and allows set requests processing
# on any object using community "administrator", but only from
# IP address 15.2.112.90.
#
# trap-dest: <trap destination>
#
# Specifies the system name where traps will be sent. The
# system name is usually the hostname or IP address of the
# management station. More than one trap destination can be
# configured by adding a separate entry for each
# destination. For example,
#
#     trap-dest: manager1
#     trap-dest: 15.2.113.223
#
# will cause traps to be sent to the systems named "manager1,"
# and "15.2.113.223".
#
# contact: <contact string>
#
# Specifies the value of the MIB-II sysContact object. This
# information usually includes the name of person responsible
# for the agent system, plus information on how to contact this
# person. For example, if Bob Jones is responsible person, and
# his phone number is 555-2000, enter:
#
#     contact: Bob Jones (Phone 555-2000)
#
# NOTE: the value specified in this file overrides snmpd.cnf
#
# location: <location string>
#
# Specifies the value of the MIB-II sysLocation object. For
# example, if the agent system is on the first floor near the
# men's room, enter:
#
#     location: 1st Floor near Men's Room
#
# NOTE: the value specified in this file overrides snmpd.cnf
#
##
#
get-community-name:      public
#set-community-name:    # enter community name
#contact:               # enter contact person for agent

```

```
#location: # enter location of agent  
#trap-dest: # enter trap destination
```

2.8 IPv6 settings

If you are using IPv6 transport, SNMP Agent must be set up as a dual-stack environment for both IPv4 and IPv6. The default setting for SNMP Agent is to use SNMP request reception ports for both IPv4 and IPv6.

2.8.1 IPv6 transport and trap destination settings

This section describes the settings for using IPv6 transport and IPv6 trap destinations.

Procedure

1. If SNMP Agent is running, execute the command `/opt/CM2/ESA/bin/snmpstop` while logged on as a superuser.
This stops SNMP Agent.
2. Specify the `-ip_proto` argument using the startup options environment variable `SNMP_MASTER_OPTIONS` in the environment variable definition file `SnmpMaster`.

By default, the `-ip_proto` argument is not defined in the environment variable definition file `SnmpMaster` when SNMP is installed. For details about the `-ip_proto` argument, see *snmpdm* in [Chapter 5. Commands and Processes](#). The following example shows the settings to make the SNMP manager and SNMP Agent use IPv6 transport only.

```
SNMP_MASTER_OPTIONS="-ip_proto ipv6 -tcplocal -aperror -apwarn -apverbose  
-hexdump -vbdump"  
export SNMP_MASTER_OPTIONS
```

3. Back up `/etc/srconf/agt/snmpd.cnf` to a location of your choice.
4. Edit `snmpTargetAddrEntry` in `snmpd.cnf`.
For details, see [2.8.1\(1\) Customizing the configuration file \(/etc/srconf/agt/snmpd.cnf\)](#).
5. Execute the command `/opt/CM2/ESA/bin/snmpstart` while logged on as a superuser.
SNMP Agent is started with the new settings applied.

(1) Customizing the configuration file (/etc/srconf/agt/snmpd.cnf)

Trap destinations are set in `snmpTargetAddrEntry` in the `snmpd.cnf` file. The following default configuration template is set in `snmpd.cnf` when the product is installed:

```
#snmpTargetAddrEntry <CONFIG_NAME> transportDomainUdpIpv6 \  
#   [<IPv6_ADDRESS>]:0 0 0 TrapConf <v1TrapParams|v2cTrapParams> readOnly \  
#   [ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff]:0 2048
```

Copy the configuration template, and then enable the settings by removing the hash mark (#) at the beginning of each line. Edit the area inside the angle brackets (< >) to set the trap destination and remove the angle brackets. If you want to add more trap destinations, add more `snmpTargetAddrEntry` definitions.

Table 2–3: snmpd.cnf settings

Setting	Description
<CONFIG_NAME>	Specify any name as the configuration name. If multiple trap destinations are set, give each a unique configuration name. The configuration name cannot exceed 32 characters, and can contain only alphanumeric characters, including underscores.
<IPv6_ADDRESS>#	Specify an IPv6 address for the trap destination. A host name cannot be specified.
<v1TrapParams v2cTrapParams>	Specify the protocol version for the SNMP trap. For SNMPv1 traps Specify v1TrapParams. For SNMPv2c traps Specify v2cTrapParams.

#

If you specify an IPv6 address with a scope ID, write the `transportDomainUdpIpv6` portion of the `snmpTargetAddrEntry` definition line as `transportDomainUdpIpv6z`.

(2) Examples of settings in the configuration file (/etc/srconf/agt/snmpd.cnf)

Below is a sample configuration for sending SNMPv1 traps to interface number 1 at IP address `fec0::1111:2222:3333:4444:5555`. The configuration name is `Trapsend_SNMPv1_IPv6`. Note that when a *%scope-ID* is specified, you must write `transportDomainUdpIpv6z`.

```
snmpTargetAddrEntry Trapsend_SNMPv1_IPv6 transportDomainUdpIpv6z \
    [fec0::1111:2222:3333:4444:5555%1]:0 0 0 TrapConf v1TrapParams readOnly \
    [ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff]:0 2048
```

Below is a sample configuration for sending SNMPv2c traps to IP address `fec0::1111:2222:3333:4444:5555`, with no *%scope-ID* specified. The configuration name is `Trapsend_SNMPv2c_IPv6`.

```
snmpTargetAddrEntry Trapsend_SNMPv2c_IPv6 transportDomainUdpIpv6 \
    [fec0::1111:2222:3333:4444:5555]:0 0 0 TrapConf v2cTrapParams readOnly \
    [ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff]:0 2048
```

Below are sample configurations for sending SNMPv2c traps to IP address `fec0::1111:2222:3333:4444:5555` and IP address `fec0::aaaa:bbbb:cccc:dddd:eeee`, with no *%scope-ID* specified. The configuration names are `NNM_1` and `NNM_2`. Note that when multiple trap destinations are set, each must be given a unique configuration name.

```
snmpTargetAddrEntry NNM_1 transportDomainUdpIpv6 \
    [fec0::1111:2222:3333:4444:5555]:0 0 0 TrapConf v2cTrapParams readOnly \
    [ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff]:0 2048
snmpTargetAddrEntry NNM_2 transportDomainUdpIpv6 \
    [fec0::aaaa:bbbb:cccc:dddd:eeee]:0 0 0 TrapConf v2cTrapParams readOnly \
    [ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff]:0 2048
```

For both SNMPv1 traps and SNMPv2c traps, the IPv6 address must be enclosed in square brackets ([]). The square brackets cannot be omitted.

2.8.2 IPv6 address format

This section describes the IPv6 address format that can be used by SNMP Agent.

- In each individual block separated by a colon (:), leading zeros can be omitted.

```
fec0:0001:2020:0033:4000:0500:0060:0077  
→ fec0:1:2020:33:4000:500:60:77
```

- Contiguous blocks of zeros can be compressed to a single unit, which can be represented by :: (double colon).

```
fec0:0:0:0:100:0:0:22  
→ fec0::100:0:0:22
```

- The final 32 bits of the address can be written in the IPv4 address format (dotted decimal notation).

```
::ffff:0b16:212c  
→ ::ffff:11.22.33.44
```

- You can attach a scope ID using a percent sign (%) at the end of the address. The scope ID can be expressed as an interface number (numeric characters) or interface name.

```
fec0::1111:2222:3333:4444:5555%4  
→ The scope ID is 4.  
  
fec0::1111:2222:3333:4444:5555%eth0  
→ The scope ID is eth0.
```

The numeric value representing an IPv6 address in the above format can contain numeric characters (0 to 9), as well as alphabetic characters (A to F and a to f). Alphabetic characters are not case sensitive.

2.9 Setting up the native agent adapter (for Solaris, AIX, and Linux)

This section describes the functions of the native agent adapter and how to configure the native agent and native agent adapter.

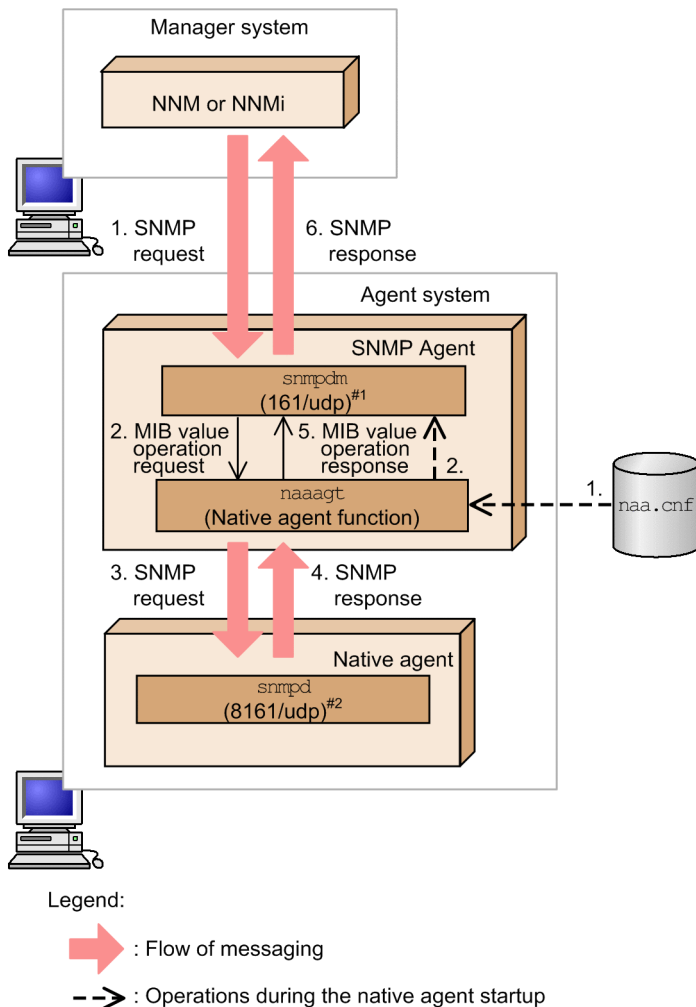
2.9.1 Functions of the native agent adapter

The *native agent adapter* provides an adapter function that is used to connect a native agent and an SNMP Agent instance. Using this function, you can acquire the MIB objects provided by the native agent and the MIB objects that do not belong to the `system` or `snmp` group under `mib-2` via SNMP. SNMPv1 and SNMPv2c communications between the native agent adapter and native agent are addressed to the IPv4 loopback address. SNMP requests from the SNMP manager to SNMP Agent work the same way when IPv6 transport is being used.

The `naa.conf` configuration file determines which MIB groups are to be acquired from the native agent. For details, see *Configuration file (naa.conf)* in *Chapter 6. Definition Files*.

The following figure shows how the native agent adapter operates.

Figure 2–3: Operations of the native agent adapter



#1: The master agent binds one of the following UDP ports as the SNMP request reception port and starts:

- For Solaris and AIX: UDP port 161

- For Linux: UDP port 22161

#2: The native agent binds one of the following UDP ports as the SNMP request reception port and starts:

- For Solaris and AIX: UDP port 8161
- For Linux: UDP port 161

The following describes operations when the native agent adapter starts and when SNMP requests are issued from NNM or NNMi. The numbers assigned to the operations correspond to those in Figure 2-3.

(1) Operations when the native agent adapter starts (indicated by the broken-line arrows)

Procedure

1. The native agent adapter reads the `naa.cnf` definition file when it starts.
2. The native agent adapter registers in the master agent the MIB objects specified in the `naa.cnf` definition file as the MIB objects that are to be processed by the native agent adapter.

(2) Operations when SNMP requests are issued from NNM or NNMi (indicated by the large filled arrows)

Procedure

1. NNM or NNMi sends an SNMP request for the MIB object that has been defined in the `naa.cnf` definition file.
2. The master agent distributes the MIB value operation request to the native agent adapter.
3. The native agent adapter re-creates the SNMP request as an SNMP packet and sends it to the UDP port (for Solaris and AIX: 8161, for Linux: 161).
The request sent to the UDP port is received by the native agent.
4. The native agent sends the SNMP response to the native agent adapter.
For the native agent, the native agent adapter functions as an SNMP manager.
5. The native agent adapter returns a MIB value operation response to the master agent.
6. The master agent returns an SNMP response to NNM or NNMi.

The following table lists the native agent adapter's target native agents.

Table 2–4: Native agent adapter's target native agents

OS	Target native agent
Solaris	<code>/usr/sbin/snmpd</code>
AIX	<code>/usr/sbin/snmpd</code>
Linux [#]	<code>/usr/sbin/snmpd</code>

[#]: In Linux, you can select how to install the native agent. Install the native agent either by using the CD-ROM, or install it while installing OS.

2.9.2 Configuring the native agent (for Solaris and AIX)

Before you can run the native agent and SNMP Agent simultaneously for the purpose of retrieving native agent MIBs via the native agent adapter, you must first configure the native agent so that it will listen for SNMP requests through port 8161. Normally, this configuration is set automatically when SNMP Agent is installed.

For details about configuring the native agent, see *2.17.1 Notes about setup (for AIX)*.

2.9.3 How to configure the native agent adapter

To configure the native agent adapter, edit the `naa.cnf` configuration file (`/etc/srconf/agt/naa.cnf`). After you have finished editing the file, start SNMP Agent to apply the settings. For details about the `naa.cnf` definition file, see *Configuration file (naa.cnf)* in *Chapter 6. Definition Files*.

2.9.4 Changing the communication protocol with the native agent

To use SNMPv2c as the communication protocol with the native agent, follow the procedure explained below to change the protocol from SNMPv1 to SNMPv2c. The same procedure can be used to change the protocol from SNMPv2c back to SNMPv1.

Procedure

1. If SNMP Agent is running, execute the `/opt/CM2/ESA/bin/snmpstop` command while logged on as a superuser.
2. Use the `SNMP_NAA_OPTIONS` environment variable to specify the startup option in the `SnpNaa` environment variable definition file:
 - To change to SNMPv1, specify the option `-v1` (or specify nothing).
 - To change to SNMPv2c, specify the option `-v2c`.
3. Execute the `/opt/CM2/ESA/bin/snmpstart` command while logged on as a superuser.

The following examples illustrate how startup options are specified in the `SnpNaa` environment variable definition file. The second example uses the `SNMP_NAA_OPTIONS` environment variable to set the option to `-v2c`.

In Solaris and AIX:

- Initial values in the `SNMP_NAA_OPTIONS` environment variable:

```
SNMP_NAA_OPTIONS="-aperror -apwarn -apverbose -hexdump -vbdump"
export SNMP_NAA_OPTIONS
```
- Example of specifying the `-v2c` option in the `SNMP_NAA_OPTIONS` environment variable:

```
SNMP_NAA_OPTIONS="-v2c -aperror -apwarn -apverbose -hexdump -vbdump"
export SNMP_NAA_OPTIONS
```

In Linux:

- Initial values in the `SNMP_NAA_OPTIONS` environment variable:

```
SNMP_NAA_OPTIONS="-aperror -apwarn -apverbose -hexdump -vbdump"
"$SNMP_NAA_OPTIONS"
```

```
export SNMP_NAA_OPTIONS
```

- Example of specifying the `-v2c` option in the `SNMP_NAA_OPTIONS` environment variable:

Do not delete the space between `-vbdump` and the double quote("").

```
SNMP_NAA_OPTIONS="-v2c -aperror -apwarn -apverbose -hexdump -vbdump  
"$SNMP_NAA_OPTIONS  
export SNMP_NAA_OPTIONS
```

2.9.5 Notes about using the native agent adapter

This subsection provides notes about using the native agent adapter.

(1) Restarting the native agent

If you want to use the native agent adapter in order to restart an individual native agent, you must restart the native agent so that it binds UDP port 8161 and that port is configured to listen for SNMP requests. In this case, instead of using the native agent start and stop commands, use the `snmpstop` command with no arguments and `snmpstart` command with no arguments. The `snmpstop` command with no arguments will stop the native agent, while the `snmpstart` command with no arguments will start the native agent.

(2) Operation performed when the `naa.cnf` configuration file is deleted

If you delete the `naa.cnf` configuration file and start the native agent adapter, the native agent adapter only retrieves MIB-II information from the native agent. However, we do not recommend that you use this method, even if you want to retrieve only MIB-II information from the native agent. Specify which MIB objects to retrieve in the `naa.cnf` configuration file.

(3) How to change the `naaagt` SNMP packet transmission port

If 8161/udp is already used by another program, change the `naaagt` SNMP packet transmission port number and SNMP reception port number to port numbers that are not used by the system that includes that program. Use the `-port` option to specify the `naaagt` SNMP packet transmission port number. Use the following method to change the native agent SNMP reception port. After the change, restart SNMP Agent using the `snmpstart` command with no arguments.

For AIX

See [3.4 Changing the SNMP reception port on SNMP Agent](#).

For Solaris

Change the following line in the file `/lib/svc/method/svc-net-snmp`:

```
/usr/sbin/snmpd udp:8161
```

(4) Notes about `naa.cnf` configuration file specifications (for Solaris)

The path names and definition specifications are different for the `naa.cnf` configuration file referenced by the `naaagt` process of the Solaris edition of SNMP Agent, and the one that is referenced by the `naaagt` process provided by NNM. The `naa.cnf` configuration file provided by NNM cannot be used as is by SNMP Agent. For details about the specification of the `naa.cnf` definition file of SNMP Agent, see [Configuration file \(naa.cnf\)](#) in [Chapter 6. Definition Files](#).

2.10 Defining extended MIB objects

To define extended MIB objects, use one of the following extended MIB definition files:

- `/etc/SnmpAgent.d/snmpd.extend` file
- Extended MIB definition file under the `/opt/CM2/ESA/ext` directory. For the extension of the extended MIB definition file, specify `def`.

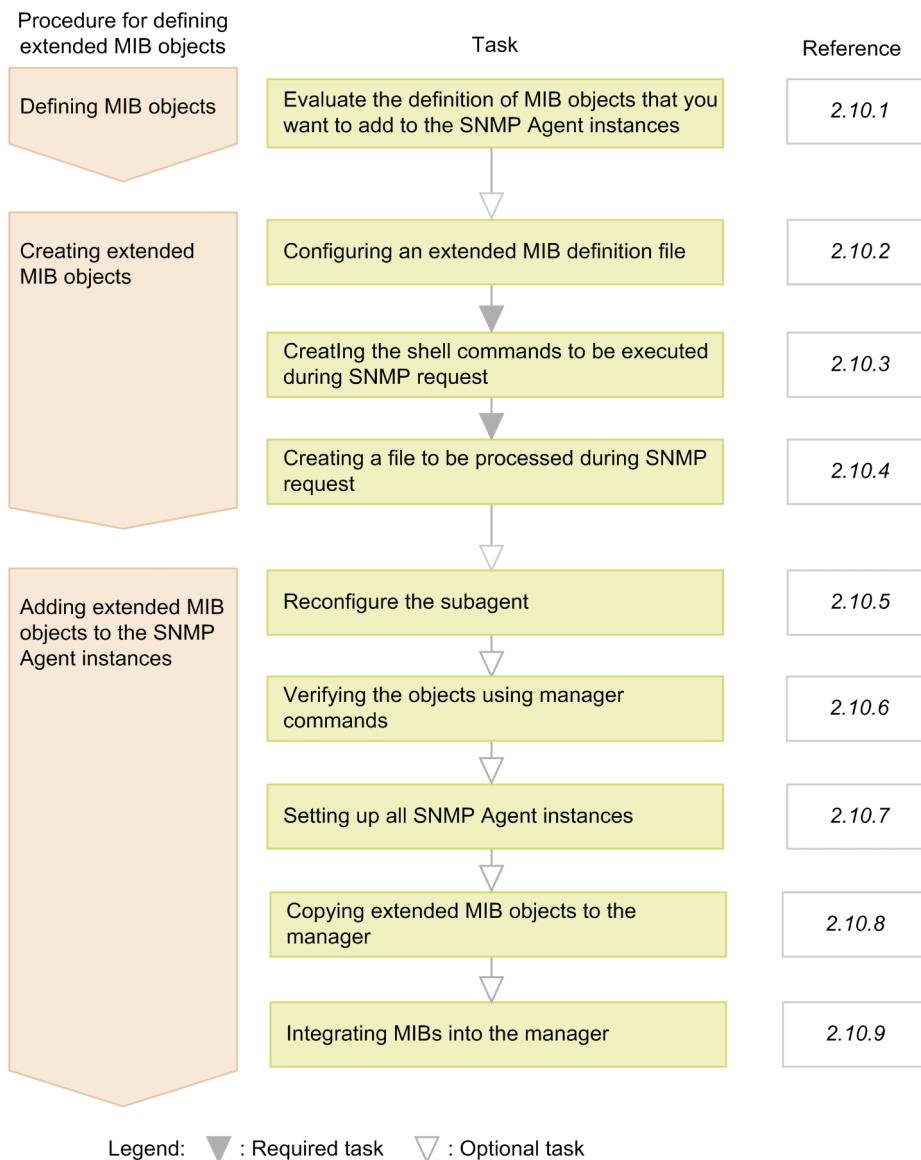
This manual describes how to define extended MIB objects using `/etc/SnmpAgent.d/snmpd.extend` as the name of the extended MIB definition file. If you want to store the extended MIB definition file under the `/opt/CM2/ESA/ext` directory, the procedure is the same as the one for adding more than one extended MIB definition file. For details, see [2.10.10 Configuring more than one extended MIB definition file](#).

To enable the manager to access the MIB objects defined in SNMP Agent, you must specify MIB modules according to the rules specified in the following RFCs:

- RFC 1155: Structure and Identification of Management Information for TCP/IP-based Internets
- RFC 1212: Concise MIB Definitions

The following figure shows the procedure for defining extended MIB objects and the related subsections.

Figure 2–4: Procedure for defining extended MIB objects and the related subsections



You can set multiple extended MIB definition files in SNMP Agent. For details about how to define multiple extended MIB definition files, see [2.10.10 Configuring more than one extended MIB definition file](#).

SNMP Agent provides the `/opt/OV/prg_samples/eagent/snmpd.extend` file as a sample extended MIB definition. For examples of MIB object definition and creation of an extended MIB definition file (`/etc/SnmpAgent.d/snmpd.extend`), see [2.10.11 Example of extended MIB object definition](#).

2.10.1 Defining MIB objects

You can define one or more MIB objects in a MIB module, and group and define them as one or more subtrees. For each subtree, you can define a maximum of 200 nodes.

To define MIB objects:

1. List all the MIB objects that you want to add to SNMP Agent.
2. Determine the logical configuration of the MIB objects.

Organize your MIB objects into logical groups. For example, the MIB objects `sysDescr`, `sysObjectID`, `sysUpTime`, `sysContact`, `sysName`, `sysLocation`, and `sysServices` all belong to the `systems` group. For more examples, see the MIB modules in the `/var/opt/OV/share/snmp_mibs` directory.

3. Define all the nodes in each subtree.

Define all the nodes in each subtree. Keep in mind that nodes can be children of other nodes.

When you define nodes, observe the following rules defined by ASN.1:

- Any number of letters, numeric characters, and hyphens are allowed.
- Begin with a lower case letter.
- Do not end with a hyphen.
- Do not use multiple hyphens consecutively.
- Do not use underscores.

The following rules also apply:

- End the counter with the character `s`.
- Make each node name unique.

4. Define the leaf nodes in the subtree.

Define the actual object (leaf node in the subtree).

When you define the actual object, assign a unique name. The following are common conventions for determining object names:

- Start all object names in a group with a prefix that can be derived from the group name. For example, the objects in the `systems` group all begin with the prefix `sys`.
- Capitalize the character after the prefix. For example, `Contact` is capitalized in the object name `sysContact`.

5. Determine where to place the object in the MIB tree.

Determine the location of the object in the MIB tree.

To ensure that your object IDs are unique, add your MIBs under your own company (enterprise) name in the `enterprises` subtree.

To use a unique enterprise ID, you can register your enterprise ID with the following:

```
Internet Assigned Numbers Authority
URL:http://www.iana.org/
Email:iana-mib@iana.org
```

The benefit of registering your enterprise ID with the Internet Assigned Numbers Authority (IANA) is that you can control your own MIB and avoid conflict with other MIBs.

2.10.2 Configuring an extended MIB definition file

Log in to the system as a root user and define extended MIB objects in the extended MIB definition file (`/etc/SnmpAgent.d/snmpd.extend`).

The `/etc/SnmpAgent.d/snmpd.extend` file is the MIB module that extends the MIB on SNMP Agent to include the objects defined by the user.

The `/etc/SnmpAgent.d/snmpd.extend` file is designed to use the macro template defined in *RFC 1212: Concise MIB Definitions*. Therefore, when you create the `/etc/SnmpAgent.d/snmpd.extend` file, follow the Abstract

Syntax Notation One (ASN.1) format described in RFC 1212. For details about MIB modules for Internet-standard MIB-II, Hewlett-Packard enterprise-specific MIB, or Hitachi enterprise-specific MIB, see the following files:

- /var/opt/OV/share/snmp_mibs/eagent/rfc1213-MIB-II
- /var/opt/OV/share/snmp_mibs/eagent/hp-unix
- /var/opt/OV/share/snmp_mibs/eagent/hitachi-cometAgt
- /var/opt/OV/share/snmp_mibs/eagent/hitachi-cometAgt-aix
- /var/opt/OV/share/snmp_mibs/eagent/hitachi-cometAgt-linux
- /var/opt/OV/share/snmp_mibs/eagent/hitachi-cometAgt-solaris

There are table and non-table formats for the extended MIB objects:

Non-table format

MIB objects have a unique value.

Table format

MIB objects are arranged in a table that consists of multiple MIB object columns and multiple entry rows. You assign a unique ID to each MIB object column to identify individual entries. As a result, each MIB object has as many MIB values as there are entries.

The following explains how to create extended MIB objects in non-table and table formats in the `/etc/SnmpAgent.d/snmpd.extend` file.

(1) Extended MIB objects in non-table format

The figure below shows the macro template that you can use when defining MIB objects in non-table format. You must fill in the fields shown in italics.

```
1. ModuleName DEFINITION ::= BEGIN
2. --comment
3. enterpriseName OBJECT IDENTIFIER ::= {objectID}
4. nodeName OBJECT IDENTIFIER ::= {objectID}
5. Object OBJECT-TYPE
6. SYNTAX data_type
7. ACCESS access_level
8. STATUS condition
9. DESCRIPTION
10. "comment"
11. READ-COMMAND: read_command
12. READ-COMMAND-TIMEOUT: command_completion_wait_time
13. WRITE-COMMAND: write_command
14. WRITE-COMMAND-TIMEOUT: command_completion_wait_time"
   ::= {parent_node_subidentifier}
   END
```

1. *ModuleName*

Specify the name of your MIB module.

2. *comment*

Write a comment as required. Note that any comment must be preceded by `--`.

3. *enterpriseName*

Specify the enterprise ID that you registered with the Internet Assigned Numbers Authority.

objectID

Specify an object identifier corresponding to the enterprise ID that you specified for *enterpriseName*. For example, Hitachi's enterprise ID is `hitachi` and its corresponding object ID is `{enterprises 116}`.

4. *nodeName*

Specify the node name of your MIB object. You can have multiple node name entries and make any node a child of another node.

objectID

Specify the name of the parent node of *nodeName* and the object identifier of *nodeName*.

5. *Object*

Specify the label of the object.

6. *data_type*

Specify a data type for *Object*. Table 2-6 lists the data types supported by SNMP Agent.

Table 2–5: Data types

Data type	Explanation
INTEGER ^{#1}	A simple type consisting of positive and negative integers, including zero. Do not use the value zero as an enumerated type.
OCTET STRING	A simple type taking zero or more octets, each octet being an ordered sequence of eight bits.
OBJECT IDENTIFIER	A type denoting an authoritatively named object. An example is 1.3.6.1.2.1.1.0
NULL	A simple type consisting of a single value, also called null. This type can only be used with defining an object associated with a command.
NetworkAddress	A type representing an IP address.
Counter ^{#2}	A type representing a non-negative integer which calculates change and increases until it reaches a maximum value. When it reaches the maximum value, it wraps around and starts increasing again from zero.
Counter64	A Counter type whose maximum is 2 ⁶⁴ -1.
Gauge ^{#2}	A type representing a non-negative integer which can increase or decrease, but which latches at a maximum value.
TimeTicks ^{#2}	A type representing a non-negative integer which indicates the time in hundredths of a second that elapsed since a certain point in time.
Opaque	A type representing an arbitrary encoding.
DisplayString	A type representing textual information taken from the NVT ASCII character set.
PhysAddress	A type representing a media address. Most media addresses are in binary representation. For example, an Internet address is represented as a string of six octets.

#1: The maximum value is 2³¹ - 1.

#2: The maximum value is 2³² - 1.

7. *access_level*

Specify the level of access allowed. Valid values are:

read-only

Means you can perform *GetRequests* but not *SetRequests* on the object.

read-write

Means you can perform both *GetRequests* and *SetRequests* on the object.

8. *condition*

Specify the required implementation condition. Valid values are mandatory, optional, obsolete, and deprecated. Except in special cases, specify mandatory.

9. *comment*

Write a comment on the object after the label of the object.

10. *read_command*

Specify the name of the command that you want SNMP Agent to execute when it receives a `GetRequest` or `GetNextRequest`. The *read_command* parameter follows the `READ-COMMAND` label. When specifying *read_command*, use the full path name. If the *access_level* value following the `ACCESS` label is `read-only` or `read-write`, you must not omit *read_command*. The output destination must be either the standard output or standard error.

11. *command_completion_wait_time* ~ <1- or 2-digit number>((1 to 90))<<3>>

Specify the number of seconds you want SNMP Agent to wait for the completion of *read_command*. The *command_completion_wait_time* parameter follows the `READ-COMMAND-TIMEOUT` label. If *read_command* is not completed within *command_completion_wait_time* seconds, SNMP Agent forces *read_command* to terminate and returns a `noSuchName` error for SNMPv1 or `noSuchInstance` for SNMPv2c to the manager. The *command_completion_wait_time* parameter is optional. If you do not specify *command_completion_wait_time*, the agent waits for 3 seconds.

12. *write_command*

Specify the name of the command that you want SNMP Agent to execute when it receives a `SetRequest`. The *write_command* parameter follows the `WRITE-COMMAND` label. When specifying *write_command*, use the full path name. If the *access_level* value following the `ACCESS` label is `read-write`, you must not omit *write_command*.

13. *command_completion_wait_time* ~ <1- or 2-digit number>((-1, 1 to 90)) <<3>>

Specify the number of seconds you want SNMP Agent to wait for the completion of *write_command*. The *command_completion_wait_time* parameter follows the `WRITE-COMMAND-TIMEOUT` label. If you specify `-1`, SNMP Agent returns a response without waiting for the completion of *write_command*. If *write_command* is not completed within *command_completion_wait_time* seconds, SNMP Agent forces *write_command* to terminate and returns a `genErr` for SNMPv1 or `commitFailed` for SNMPv2c to the manager. The *command_completion_wait_time* parameter is optional. If you do not specify *command_completion_wait_time*, the agent waits for 3 seconds. SNMP Agent processes one command at a time. Furthermore, the agent waits for a response from each command before processing the next command. This parameter has no bearing on the access level or file read/write permissions.

14. *parent_node*

Specify the name of the parent node. *parent_node* must be a node name you have already defined.

subidentifier

Specify the number that uniquely identifies the *parent_node* object.

(2) Extended MIB objects in table format

To define an SNMP table, you must code the `SYNTAX SEQUENCE OF` and `INDEX` clauses. The figure below shows the macro template that you can use when defining MIB objects in table format. You must fill in the fields shown in italics.

This macro template is based on a MIB table that consists of objects 1 and 2. Note that the fields identical to those in non-table format are not explained below.

```

ModuleName DEFINITIONS ::= BEGIN
--comment
enterpriseName      OBJECT IDENTIFIER ::= {objectID}
nodeName            OBJECT IDENTIFIER ::= {objectID}
1. table_name OBJECT-TYPE
2. SYNTAX SEQUENCE OF entry_data_type
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"comment"
3. FILE-COMMAND : file_command
4. FILE-COMMAND-FREQUENCY : file_command_execution_interval
5. PIPE-IN-NAME : pipe_in_name
6. PIPE-OUT-NAME : pipe_out_name
7. PIPE-FREQUENCY : pipe_writing_interval
8. APPEND-COMMUNITY-NAME : {true|false}
9. FILE-NAME : file_name
::= {parent_node subidentifier}
10. entry_name OBJECT-TYPE
SYNTAX entry_data_type
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
"comment"
INDEX {object_1}
::= {table_name 1}
entry_data_type ::=
SEQUENCE{
object_1 data_type_1,
object_2 data_type_2
}
11. object_1 OBJECT-TYPE
SYNTAX data_type_1
ACCESS access_level
STATUS condition
DESCRIPTION
"comment"
::= {entry_name 1}
object_2 OBJECT-TYPE
SYNTAX data_type_2
ACCESS access_level
STATUS condition
DESCRIPTION
"comment"
::= {entry_name 2}
END

```

1. *table_name*

Specify a label for your MIB table.

2. *entry_data_type*

Specify a data type for the entries of your MIB table. Usually, you will use the first character (in uppercase) of the MIB table entry label. The *entry_data_type* parameter specifies the data type of columns in the MIB table. In this example, objects 1 and 2 are columns of the MIB table.

3. *file_command*

When the agent receives an SNMP GetRequest, GetNextRequest, or SetRequest, the agent executes the *file_command* before either reading or creating the *file_name*. When the agent receives an SNMP SetRequest, the agent also executes the *file_command* after the *file_name* has been created. The *file_command* parameter follows the FILE-COMMAND label. When specifying *file_command*, use the full path name.

You can specify a monitoring time of 90 seconds or less during which SNMP Agent will wait for a command response, using the *-fcmdguard* option of the *extsubagt* command. The default monitoring time is 10 seconds. If no response returns during this period, SNMP Agent will execute *kill* on this command and perform as follows.

For SNMPv1

SNMP Agent returns a `genErr` error to the manager.

For SNMPv2c

If SNMP Agent has received `get_request`, it returns a `noSuchName` error to the manager.#

If SNMP Agent has received `get_next_request`, it returns an `EndOfMibView` error to the manager.#

If SNMP Agent has received `set_request`, it returns a `genErr` error to the manager.

#: A value in the `VarBind` list. The status is normal.

You can set the `extsubagt` options so that they will always be effective when the operating system starts or when the `snmpstart` command is executed. For details about the option settings for the `extsubagt` process, see [3.1 Starting SNMP Agent](#).

4. `file_command_execution_interval` ~ <number>((0 to 2147483647))<<10>>

When the agent receives an SNMP `GetRequest` or `GetNextRequest`, the agent executes the `file_command` if the agent last executed the `file_command` more than `file_command_execution_interval` seconds ago.

The `file_command_execution_interval` parameter follows the `FILE-COMMAND-FREQUENCY` label. The unit is the second. If the received request is a `SetRequest`, SNMP Agent does not check the time that elapsed since `file_command` was last executed. The `file_command_execution_interval` parameter is valid only if you specified `file_command` after the `FILE-COMMAND` label. The `file_command_execution_interval` parameter is optional. The default value is 10 seconds.

5. `pipe_in_name`

Specify the name of the file that the UNIX process uses to notify SNMP Agent of the completion of processing after writing data into the file specified after the `PIPE-OUT-NAME` label. The `pipe_in_name` parameter follows the `PIPE-IN-NAME` label. When specifying `pipe_in_name`, use the full path name.

You can specify a monitoring time of 90 seconds or less from the time data is written to `pipe_out_name` to the time processing results are written, using the `-pipeguard` option of the `extsubagt` command. The default monitoring time is 20 seconds.

If the monitoring time extends past a certain period, it might cause the next request to be written before the processing results of the current request are written. For cases when such a request-response mismatch occurs, an identification number can be added to the beginning of the data to be written to `pipe_in_name` so that responses can be easily matched against requests. You can also specify this identification number by using the `-invokeid` option of the `extsubagt` command.

The following table shows the data to be passed to `pipe_in_name`.

Table 2–6: Data to be passed to `pipe_in_name`

Data item	Value to be passed
Identification number	Identifies the request for the processing result. This identification number is added only when <code>extsubagt</code> 's option (<code>-invokeid</code>) is specified. The format is <code>xxxxxxxx.yyzzzzzz</code> (where <code>xxxxxxxx</code> : absolute time in seconds, <code>yyzzzzzz</code> : microseconds).
Result code	0

SNMP Agent behaves as follows if the contents of `pipe_in_name` are not 0 (4-byte numeric value) or if processing results are not written into `pipe_in_name` within the specified monitoring time.

For SNMPv1

SNMP Agent returns a `genErr` error to the manager.

For SNMPv2c

If SNMP Agent has received `get_request`, it returns a `noSuchName` error to the manager.#

If SNMP Agent has received *get_next_request*, it returns an `EndOfMibView` error to the manager.#

If SNMP Agent has received *set_request*, it returns a `genErr` error to the manager.

#: A value in the `VarBind` list. The status is normal.

You can set the `extsubagt` options so that they will always be effective when the operating system starts or when the `snmpstart` command is executed. For details about the option settings for the `extsubagt` process, see [3.1 Starting SNMP Agent](#).

6. *pipe_out_name*

Specify the name of the file into which SNMP Agent writes the data to be passed to the UNIX process. When SNMP Agent receives a `GetRequest`, `GetNextRequest`, or `SetRequest`, SNMP Agent writes the data into this file before reading the *file_name* file specified after the `FILE-NAME` label. If the received request is a `SetRequest`, SNMP Agent writes data into the *pipe_out_name* file before and after reading the *file_name* file. The *pipe_out_name* parameter follows the `PIPE-OUT-NAME` label. When specifying *pipe_out_name*, use the full path name. The *pipe_out_name* parameter is optional. Note, however, that you must use the `PIPE-OUT-NAME` and `PIPE-IN-NAME` labels in a pair. As in the case with *pipe_in_name*, you can add an identification number to the beginning of the data to be written to *pipe_out_name*. Specify this identification number using the `-invokeid` option of the `extsubagt` command.

The table below shows the data to be passed to *pipe_out_name*. These data items are separated by a space. The data is of the character string type. A `\0` is appended to the end of the data.

Table 2–7: Data to be passed to *pipe_out_name*

Data item	Explanation
Identification number	Identifies a request. This identification number is added only when it is specified using the <code>-invokeid</code> option of the <code>extsubagt</code> command. The format is <code>xxxxxxxx.yyzzzzzz</code> (where <code>xxxxxxxx</code> is the absolute time in seconds and <code>yyzzzzzz</code> represents the fragment in microseconds).
Manager's IP address	Manager's IP address in the Internet dot notation
Community name	Community name specified in the request
Object identifier	Object identifier specified in the request. The object identifier is in the dot notation.
Object syntax	The syntax of the object being requested is passed as one of the following values: <code>Integer</code> , <code>OctetString</code> , <code>ObjectIdentifier</code> , <code>Null</code> , <code>NetworkAddress</code> , <code>IpAddress</code> , <code>Counter</code> , <code>Counter64</code> , <code>Gauge</code> , <code>TimeTicks</code> , <code>Opaque</code> , <code>DisplayString</code> , or <code>PhysAddress</code> .
PDU type	Type of the request. Valid values are <code>GetRequest</code> , <code>GetNextRequest</code> , <code>SetRequest</code> , and <code>PostSetRequest</code> . <code>PostSetRequest</code> is the value that SNMP Agent passes to the UNIX process after reading the <i>file_name</i> file.
Instance name	Suffix specified in the request
Set value	Only when the request type is <code>SetRequest</code> , SNMP Agent passes the <code>Set</code> value in the <code>SetRequest</code> to the UNIX process.

You can set the options of the `extsubagt` process to be always enabled at startup or during `snmpstart` command execution. For details about the option settings for the process, see [3.1 Starting SNMP Agent](#).

7. *pipe_writing_interval* ~ <number>((0 to 2147483647))<<10>>

When SNMP Agent receives a `GetRequest` or `GetNextRequest`, the agent writes to the *pipe_out_name* if the agent last wrote to the *pipe_out_name* more than *pipe_writing_interval* seconds ago. The *pipe_writing_interval* parameter follows the `PIPE-FREQUENCY` label. The unit is the second. If the received request is a `SetRequest`, SNMP Agent does not check the time that elapsed since *pipe_out_name* was last written. The *pipe_writing_interval* parameter is valid only if you specified both the `PIPE-IN-NAME` and `PIPE-OUT-NAME` labels. The default value is 10 seconds.

8. APPEND-COMMUNITY-NAME: { *true* | *false* }

If you specify *true*, when SNMP Agent receives a request (`GetRequest`, `GetNextRequest`, or `SetRequest`), it names the file that it reads or writes by appending the community name specified in the request to the file name *file_name* specified in the `FILE-NAME` label. If you specify *false*, SNMP Agent reads or writes the file whose file name is *file_name* (that is, the community name is not appended). You can omit the entire `APPEND-COMMUNITY-NAME: { true | false }` line. If you omit it, *false* is assumed.

9. *file_name*

Specify the name of the file that SNMP Agent reads or writes when it receives a `GetRequest`, `GetNextRequest`, or `SetRequest`. The *file_name* parameter follows the `FILE-NAME` label. When specifying *file_name*, use the full path name. You must not omit the *file_name* parameter.

Before execution of *file_command*, or before a write to *pipe_out_name*, SNMP Agent checks whether the specified file exists. If the file does not exist, an error occurs and the MIB value cannot be acquired. SNMP Agent checks for existence of the file, but does not check the contents of the file.

10. *entry_name*

Specify the label of a MIB table entry. The data type of the object is determined by *entry_data_type*. After the `INDEX` label in the definition of the object, you specify a MIB object whose MIB value uniquely identifies the entry row within the MIB table column. In this example, *object_1* is specified.

11. *object_1*

The following provides notes on defining MIB objects as columns in a MIB table.

(3) Notes

Note that the `/etc/SnmpAgent.d/snmpd.extend` file differs from the RFCs in the following areas:

- The `imports` and `exports` clauses are not required in the `/etc/SnmpAgent.d/snmpd.extend` file and will be ignored if added.
- The `DESCRIPTION` clause is required. Use this field to define the commands you want to execute. By adding a description of the command in the `DESCRIPTION` field, you will be able to check the details of commands that are executed against requests from the manager.
- If you specify a label two or more times in the `DESCRIPTION` clause, the second and subsequent labels are interpreted as the values of the first label.
- You must specify the labels in the `DESCRIPTION` clause in the order in which they appear in the definition macro template. If they are out of order, a definition statement analysis error will result.
- When you define extended MIB objects in table format by using the extended MIB object definition function, define them using no more than 255 columns for the table.
- Object names and entry names that you define must be no longer than 59 characters.

2.10.3 Creating the shell commands to be executed during SNMP request

You must create shell commands that you specified in the `/etc/SnmpAgent.d/snmpd.extend` file. The following explains how to create them.

The shell commands are UNIX shell scripts or programs. The `/opt/OV/prg_samples/eagent` directory contains an example of creating the `/etc/SnmpAgent.d/snmpd.extend` file. This example contains sample shell commands.

Note the following when determining command names:

- You specify these commands in the DESCRIPTION clause in the `/etc/SnmpAgent.d/snmpd.extend` file.
- The maximum command size is 5120 characters.
- A command can span multiple lines. End each line with a backslash (`\`).
- Commands that are defined using READ-COMMAND, WRITE-COMMAND, or FILE-COMMAND can be executed by a root user. Assign a file attribute to these commands to enable the commands to be executed by a root user.

The following procedure shows how to create a command.

Procedure

1. Log in to the system on which you want to execute the command.
2. Write a shell script or program.
3. Verify the operation of the shell command.
4. Check the exit code.
5. Check the arguments of the shell command.

The procedure for creating commands is as follows:

(1) Logging in to the system

Log in to the system on which you want to execute the commands.

(2) Writing a script or program

The following explains how to write a shell script or program.

- Arguments
You can configure SNMP Agent to pass some arguments to your command. Table 2-9 lists the available arguments. These arguments can be in any order.

Table 2–8: Arguments

Argument	Value to be passed
<code>\$i</code>	The management station's IP address. The address is in internet dot notation.
<code>\$c</code>	The community name used in the request.
<code>\$o</code>	The OBJECT IDENTIFIER used in the request. The OBJECT IDENTIFIER is in dot notation.
<code>\$s</code>	The SYNTAX of the object. One of the following values will be passed: Integer, OctetString, ObjectIdentifier, Null, NetworkAddress, IPAddress, Counter, Counter64, Gauge, TimeTicks, Opaque, DisplayString, or PhysAddress.
<code>\$r</code>	The request issued by the management station. One of the following values will be passed: GetRequest, GetNextRequest, SetRequest, or PostSetRequest. The PostSetRequest will be passed to the <i>file_command</i> after the <i>file_name</i> has been created.
<code>\$I</code>	The instance used in the request.
<code>\$*</code>	This is the same as specifying <code>\$i \$c \$o \$s \$r \$I</code> .

Argument	Value to be passed
\$\$	To pass a \$ character as an argument value to the command, specify \$\$\$. One of the two \$ characters is passed as the argument value.
Set value	Only when the request is a SetRequest or PostSetRequest, the Set value in the SetRequest is passed to the command. The Set value is passed as the last item of the argument list.

Suppose that you are defining MIB objects in non-table format so that SNMP Agent executes the `/opt/OV/prg_samples/eagent/num_widgets` command. To make SNMP Agent pass the manager's IP address, community name, and object identifier as arguments to the command, code the following line in the DESCRIPTION clause in the `/etc/SnmpAgent.d/snmpd.extend` file:

```
READ-COMMAND: /opt/OV/prg_samples/eagent/num_widgets $i $c $o
```

Suppose that you are defining MIB objects in table format so that SNMP Agent executes the `/opt/OV/prg_samples/eagent/update_inetd` command before reading the file specified after the FILE-NAME label. To make SNMP Agent pass the request type as an argument to the command, code the following line in the DESCRIPTION clause in the `/etc/SnmpAgent.d/snmpd.extend` file:

```
FILE-COMMAND: /opt/OV/prg_samples/eagent/update_inetd $r
```

If you do not specify arguments, SNMP Agent will not pass any arguments to *read_command* or *file_command*. For *write_command* or *file_command*, SNMP Agent passes only one Set value in the request if you do not specify arguments for the command.

- Arguments that you must specify to use the same command for both get and set operations
When specifying one command for both the READ-COMMAND and WRITE-COMMAND lines, you must specify the appropriate arguments in order to distinguish between GetRequest and SetRequest. You must specify the following arguments:

READ.REQ for a get operation

WRITE.REQ for a set operation

For example, to use the `/usr/bin/my_command` command for both READ-COMMAND and WRITE-COMMAND, enter the following:

```
READ-COMMAND
    /usr/bin/my_command READ.REQ
WRITE-COMMAND
    /use/bin/my_command WRITE.REQ
```

- Search path
When specifying a command name, use the full path name.
- Return values
Output the return values from the command specified on the READ-COMMAND label to standard out or standard error.
- Execution
The commands you created appear to be executed by `/bin/sh`. You can also specify shell commands such as `exit`, `read`, `if`, and `for`.
- Exit codes
Any shell command written with a READ-COMMAND or WRITE-COMMAND label must be terminated with a 0. If this type of command is terminated with a non-zero exit code, SNMP Agent behaves as follows:

For SNMPv1

If SNMP Agent has received a `get-request`, it returns a `noSuchName` error to the manager.

If SNMP Agent has received a `get-next-request`, it searches for the next object.

If SNMP Agent has received a `set-request`, it returns a `genErr` error to the manager.

For SNMPv2c

If SNMP Agent has received a `get-request`, it returns a `noSuchInstance` error to the manager.

If SNMP Agent has received a `get-next-request`, it searches for the next object.

If SNMP Agent has received a `set-request`, it returns a `commitFailed` error to the manager.

The command specified after the `FILE-COMMAND` label must always terminate with exit code 0. If the exit code is not 0, SNMP Agent behaves as follows:

For SNMPv1

If SNMP Agent has received a `get-request`, it returns a `noSuchName` error to the manager.

If SNMP Agent has received a `get-next-request`, it searches for the next object.

If SNMP Agent has received a `set-request`, it returns a `genErr` error to the manager.

For SNMPv2c

If SNMP Agent has received a `get-request`, it returns a `noSuchInstance` error to the manager.

If SNMP Agent has received a `get-next-request`, it searches for the next object.

If SNMP Agent has received a non-table version of `SetRequest` or `PostSetRequest`, a type of `set-request`, it returns a `commitFailed` error to the manager.

If SNMP Agent has received a table version of `SetRequest`, a type of `set-request`, it returns a `genErr` error to the manager.

If SNMP Agent has received a table version of `PostSetRequest`, a type of `set-request`, it returns a `commitFailed` error to the manager.

All the behaviors shown above also apply when SNMP Agent's attempt to execute a user-defined shell command resulted in an error such that the command was not found.

(3) Verifying the operation of your shell command

Execute the shell command to make sure that it executes successfully.

To check whether the `fmailListMsg`-related command in the `/etc/SnmpAgent.d/snmpd.extend` file executes successfully, execute the following command:

```
/usr/bin/mailq
```

If the execution is successful, the command outputs a mail message list to standard output.

(4) Checking the exit code

Execute the following command and check the exit code. If the exit code is 0, your shell command has normally terminated.

```
echo $?
```

(5) Checking the arguments of your shell command

If your shell command has arguments, verify the arguments.

The following example checks the definition and arguments in the `/etc/SnmpAgent.d/snmpd.extend` file:

Definition in the `/etc/SnmpAgent.d/snmpd.extend` file

```
READ-COMMAND:/opt/OV/prg_samples/eagent/num_widgets $i $c $o $s
```

Checking the arguments

```
num_widgets 15.2.3.149 public 1.3.6.4.4242.3.1 Gauge
```

2.10.4 Creating the file to be processed during an SNMP request

If your extended MIB objects are in table format, you must create a `file_name` file. The following explains the method and provides notes on the procedure.

Note the following when specifying `file_name`:

- You specify the `file_name` in the DESCRIPTION clause in the `/etc/SnmpAgent.d/snmpd.extend` file.
- You must specify `file_name` using the full path name.
- The maximum `file_name` size is 5,120 characters.
- A `file_name` can span multiple lines. Optionally, end each line with a backslash (`\`).

The procedure for creating the file is as follows:

1. Each row of the table ends in a new-line. A row can continue over a new-line by adding a backslash (`\`) at the end of the line. In this way, you can use more than one line to code each table row.

For example, SNMP Agent interprets the file shown below as a table that consists of one row and five columns.

```
Column1 "Column # 2" \  
        "Column # 3" Column4 Column5
```

2. Separate columns in each row by a space, or enclose each column in double quotation marks (`"`). If a column value to be enclosed in double quotation marks contains a double quotation mark (`"`) or backslash (`\`), specify

`\"` to represent `"`

`\\` to represent `\`

For example, if the file contains

```
"This is an \"example\" of a column with \" style quotes"
```

The agent would return the following to the management station.

```
This is an "example" of a column with \" style quotes
```

3. If the SYNTAX value of the MIB object is `PhysAddress`, `OCTET STRING`, or `Opaque`, SNMP Agent regards any character string beginning with `0x` as a hexadecimal number. For example, if the file contains `0x0800093519D0`, SNMP Agent returns `(0800093519D0)16` to the manager.

4. If the value in the first column is `#`, SNMP Agent regards the value as a comment and skips it.

2.10.5 Reconfiguring the subagent

At startup, SNMP Agent checks the contents of `/etc/SnmpAgent.d/snmpd.extend`. If there is no error, SNMP Agent adds the objects.

Make sure that there is no syntax error in `/etc/SnmpAgent.d/snmpd.extend`, and then use the `/opt/CM2/ESA/bin/snmpstart` command to start SNMP Agent. You can check the syntax of `/etc/SnmpAgent.d/snmpd.extend` by using the `extsubagt` process with the `-p -apall` option specified. Note that you must execute the `extsubagt` process while SNMP Agent is stopped by the `snmpstop` command.

2.10.6 Verifying the objects using manager commands

Verify that SNMP Agent responds to the added objects by using the command that is provided for object verification via manager commands. For details about the commands, see the documentation on object verification via individual manager commands.

2.10.7 Setting up all SNMP Agent instances

Use one of the following methods to set up the SNMP Agent instances:

- Copy `/etc/SnmpAgent.d/snmpd.extend` to all SNMP Agent instances.
After copying, you must reconfigure the SNMP Agent instances.
- Create an `/etc/SnmpAgent.d/snmpd.extend` file for each SNMP Agent instance.
If you want different SNMP Agent instances to manage different objects, you can create a separate `/etc/SnmpAgent.d/snmpd.extend` file for each SNMP Agent instance. In this case, make sure that each object ID in use is unique. The description related to each object ID must be consistent for all the SNMP Agent instances. Repeat the procedure shown in Figure 2-4 to create an `/etc/SnmpAgent.d/snmpd.extend` file for each SNMP Agent instance.

2.10.8 Copying extended MIB objects to the manager

Before the manager can access the new MIB objects you added to the SNMP Agent instances, you must copy the `/etc/SnmpAgent.d/snmpd.extend` file to the manager.

For NNM, you can copy the file to the default directory for MIB modules. This helps you when referencing the MIB module file. The default directory is `/var/opt/OV/share/snmp_mibs`.

2.10.9 Integrating MIBs into the manager

After copying the MIB module to a manager, you must integrate the new MIB objects into the manager's MIB. For details about how to integrate MIBs, see the documentation provided by each manager.

To integrate MIBs into NNM:

1. Load new MIB objects onto the manager's MIB.

Before NNM can access the new MIB objects, the MIB module defining these objects must be loaded on NNM's MIB. To load the module, use the NNM command `xnmloadmib` or **Load/Unload MIBs: SNMP**.

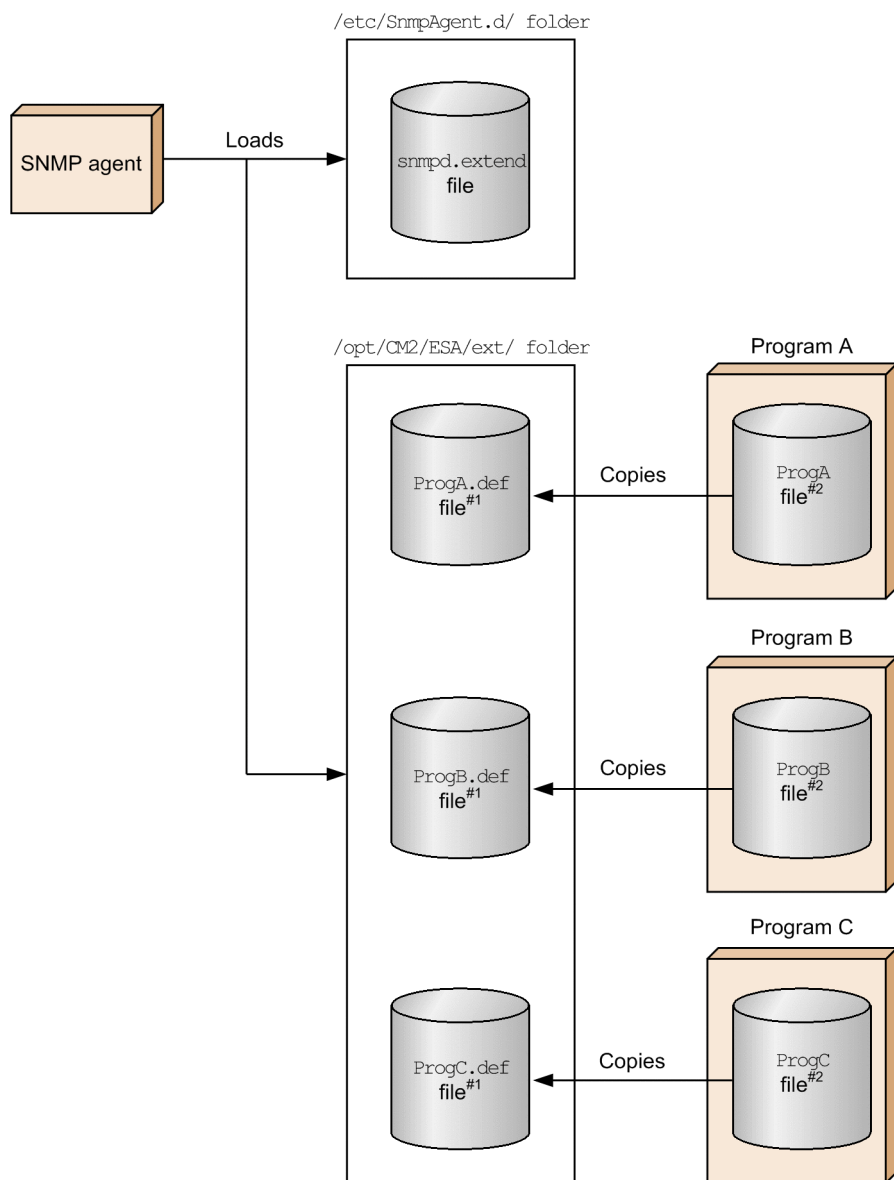
- For NNM, to manage new objects, use **MIB Browser: SNMP**, **Data Collection & Thresholds: SNMP**, **MIB Application Builder: SNMP**, and applications created with **MIB Application Builder: SNMP**.

2.10.10 Configuring more than one extended MIB definition file

When you want to add extended MIB definition files to SNMP Agent, you must create new extended MIB definition files.

The following figure gives an overview of the procedure for creating an extended MIB definition file per program and setting the created extended MIB definition files in SNMP Agent.

Figure 2–5: Overview of creating an extended MIB definition file for each program



#1: A file created by copying an extended MIB definition file and changing its extension to `.def`.

#2: An extended MIB definition file created for each program.

(1) How to add the extended MIB definition file

This subsection describes how to add the created extended MIB definition file to SNMP Agent.

- Storing the extended MIB definition file

Either copy the extended MIB definition file to the `/opt/CM2/ESA/ext` directory and change the file extension to `.def` or create a symbolic link.

The maximum length of the file name is 12 characters including the extension (*1-to-8-characters.extension*).

Set the attributes of the file in such a manner that the `extsubagt` process (owner: `bin`, group: `bin`) can read them.

To set the start options for the extended MIB objects that are to be added, create an option definition file under the `/opt/CM2/ESA/ext` directory. For details about how to create an option definition file, see [2.10.10\(3\) How to set the startup options definition file for an extended MIB object](#).

- Reading the extended MIB objects

In order to provide the extended MIB objects, you must start the `extsubagt` process that is specified in the extended MIB definition file.

You can start the `extsubagt` process specified in the extended MIB definition file either by stopping SNMP Agent or without stopping SNMP Agent. Each method is explained below.

<Adding the extended MIB definition file by stopping SNMP Agent>

To add the extended MIB definition file by stopping SNMP Agent:

1. Execute the `/opt/CM2/ESA/bin/snmpstop` command as a superuser.
SNMP Agent stops.
2. Execute the `/opt/CM2/ESA/bin/snmpstart` command as a superuser.
SNMP Agent starts, and then the `extsubagt` process specified in the extended MIB definition file starts.

<Adding the extended MIB definition file without stopping SNMP Agent>

1. Execute the `/opt/CM2/ESA/bin/snmpstart -e` command as a superuser.
The `extsubagt` process specified in the extended MIB definition file starts.

(2) Checking whether extsubagt has started for each extended MIB definition file

Use the `snmpcheck` command to check whether `extsubagt`, specified in the defined extended MIB definition file, has started.

The following figure shows how to check whether all the `extsubagt` programs have normally started when the `/etc/SnmpAgent.d/snmpd.extend` file has been set, in addition to the extended MIB definition files (`ProgA.def`, `ProgB.def`, and `ProgC.def`) under the `/opt/CM2/ESA/ext` directory.

Figure 2–6: Checking whether extsubagt has started for each extended MIB definition file

```
#!/opt/CM2/ESA/bin/snmpcheck
snmpdm    running pid=29771
mib2agt   running pid=29779
hp_unixagt    running pid=29787
trapdestagt  running pid=29789
extsubagt    running pid=29795  file = /etc/SnmpAgent.d/
snmpd.extend
extsubagt    running pid=3593  file = /opt/CM2/ESA/ext/ProgA.def
extsubagt    running pid=3594  file = /opt/CM2/ESA/ext/ProgB.def
extsubagt    running pid=3596  file = /opt/CM2/ESA/ext/ProgC.def
htc_unixagt1  running pid=29791
htc_unixagt2  running pid=29793
```

(3) How to set the startup options definition file for an extended MIB object

The following procedure shows how to specify startup options for an extended MIB object. These options are enabled at the start of the OS or SNMP Agent.

Procedure

1. Create a startup options definition file with the extension `opt` in the `/opt/CM2/ESA/ext` folder.
Make sure that the attributes of the file allow the `extsubagt` process (owner: `bin`, group: `bin`) to read the file.
The following are the names of the extended MIB definition file and startup options definition file that are to be created:
 - Name of the extended MIB definition file: `/opt/CM2/ESA/ext/ProgA.def`
 - Name of the startup options definition file: `/opt/CM2/ESA/ext/ProgA.opt`
2. Set the options you want to enable when executing the extended MIB object.

The figure below shows how to set options when specifying the following settings:

- Command response interval specified in `FILE_COMMAND`: 20 seconds
- Pipe response monitoring interval specified in `PIPE_IN_NAME` and `PIPE_OUT_NAME`: 25 seconds
- Whether to use the ID in the data match judgment for the data sent or received through a pipe: `Yes`

Figure 2–7: Example of setting the options to be enabled during execution of an extended MIB object

```
SNMP_EXTAGT_OPTIONS="-fcmdguard 20 -pipeguard 25 -invokeid"
export SNMP_EXTAGT_OPTIONS
```

Perform the following procedure to change the options for an active extended MIB object without stopping SNMP Agent:

Procedure

1. Execute the `snmpcheck` command.
2. Check the process ID of the extended MIB object for which you want to change the options.
3. Execute `kill -9 process-ID-obtained-in-step-2` as a superuser.

The extended MIB object stops.

4. Edit the startup options definition file.
5. Execute `/opt/CM2/ESA/bin/snmpstart -e` as a superuser.
SNMP Agent starts.

(4) Notes

- The files that are placed under the `/opt/CM2/ESA/ext` directory, but that do not have `.def` as an extension, are not read as extended MIB definition files.
- The contents of an extended MIB definition file are checked when SNMP Agent starts. If the file has an error, SNMP Agent does not start. Therefore, you must perform syntax checking before startup using the following process:

In HP-UX (IPF):

```
/opt/CM2/ESA/bin/extsubagt -e extended-MIB-definition-file -p -apall
```

In Solaris, AIX and Linux:

```
/usr/sbin/extsubagt -e extended-MIB-definition-file -p -apall
```

- You can define a maximum of 100 extended MIB definition files.

2.10.11 Example of extended MIB object definition

Now, let us define extended MIB objects. The procedure for defining extended MIB objects consists of the following steps:

- Defining MIB objects
- Logging in to the system (`flintagent` in this example)
- Creating an `/etc/SnmpAgent.d/snmpd.extend` file

(1) Defining MIB objects

Suppose, for example, that you work for the Flintstones Company. Your goal is to define MIB objects so that you can:

1. list users who are using the system
2. reference the contents of memory on each machine
3. manage mail queues
4. manage the number of widgets produced per hour on an unattended system
5. keep track of the LP scheduler
6. reference the default printer
7. reference the user IDs, the amount of disk space used by each user, and the users' e-mail address list
8. list root processes
9. modify the `inetd(1M)`'s configuration file

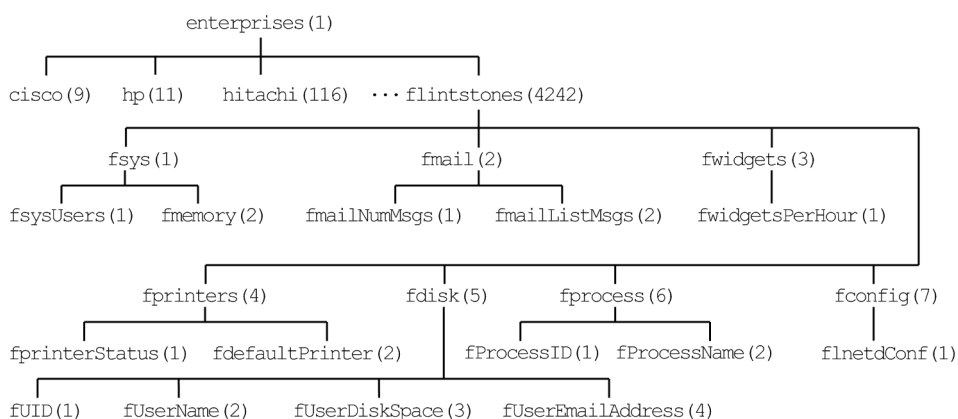
For details about defining the above MIB objects, see the description of MIB object definition in [2.10.11\(3\) Creating an `/etc/SnmpAgent.d/snmpd.extend` file](#).

Your agent system is flintagent with the default community name public. The *set* community name is secret. To ensure that your object IDs are unique, you decide to define your MIB objects in the flintstones (4242) subtree. The MIB tree for this example has the following layers. The following shows the structure of this MIB tree.

```
iso (1) ::= { 1 }
org (3) ::= { 3 }
dod (6) ::= { 6 }
internet (1) OBJECT IDENTIFIER ::= { dod 1 }
private (4) OBJECT IDENTIFIER ::= { internet 4 }
enterprises (1) OBJECT IDENTIFIER ::= { private 1 }
flintstones (4242) OBJECT IDENTIFIER ::= { enterprises 4242 }
fsys (1) OBJECT IDENTIFIER ::= { flintstones 1 }
fmail (2) OBJECT IDENTIFIER ::= { flintstones 2 }
fwidgets (3) OBJECT IDENTIFIER ::= { flintstones 3 }
fprinters (4) OBJECT IDENTIFIER ::= { flintstones 4 }
fdisk(5) OBJECT IDENTIFIER ::= { flintstones 5 }
fprocess(6) OBJECT IDENTIFIER ::= { flintstones 6 }
fconfig(7) OBJECT IDENTIFIER ::= { flintstones 7 }
```

The following figure illustrates the structure of the MIB tree.

Figure 2–8: Structure of the MIB tree



Each leaf node has one object identifier as shown below.

<i>Leaf node</i>	<i>Object identifier</i>
fsysUsers	1.3.6.1.4.1.4242.1.1.0
fmemory	1.3.6.1.4.1.4242.1.2.0
fmailNumMsgs	1.3.6.1.4.1.4242.2.1.0
fmailListMsgs	1.3.6.1.4.1.4242.2.2.0
fwidgetsPerHour	1.3.6.1.4.1.4242.3.1.0
fprintersStatus	1.3.6.1.4.1.4242.4.1.0
fdefaultPrinter	1.3.6.1.4.1.4242.4.2.0
fUID	1.3.6.1.4.1.4242.5.1.1.1.fUID#1
fUserName	1.3.6.1.4.1.4242.5.1.1.2.fUID
fUserDiskSpace	1.3.6.1.4.1.4242.5.1.1.3.fUID
fUserEmailAddress	1.3.6.1.4.1.4242.5.1.1.4.fUID
fProcessID	1.3.6.1.4.1.4242.6.1.1.1.fProcessID#2
fProcessName	1.3.6.1.4.1.4242.6.1.1.2.fProcessID
fInetdConf	1.3.6.1.4.1.4242.7.1.0

#1: FUID value

#2: fProcessID value

(2) Log in as a root user to the flintagent system

Log in to the flintagent system as a root user.

(3) Creating an /etc/SnmpAgent.d/snmpd.extend file

This subsection presents an example of creating an /etc/SnmpAgent.d/snmpd.extend file.

The contents of the sample file /etc/SnmpAgent.d/snmpd.extend are provided in the sample extended MIB definition file /opt/OV/prg_samples/eagent/snmpd.extend.

```
FLINTSTONES-MIB DEFINITIONS ::= BEGIN

    --
    -- Test MIB- used for testing snmpd.ea(extensible agent)
    --
    --
    internet      OBJECT IDENTIFIER ::= { iso(1) org(3) dod(6) internet(1) }
    enterprises   OBJECT IDENTIFIER ::= { internet private(4) 1 }
    flintstones   OBJECT IDENTIFIER ::= { enterprises 4242 }
    fsys          OBJECT IDENTIFIER ::= { flintstones 1 }
    fmail         OBJECT IDENTIFIER ::= { flintstones 2 }
    fwidgets     OBJECT IDENTIFIER ::= { flintstones 3 }
    fprinters    OBJECT IDENTIFIER ::= { flintstones 4 }
    fdisk        OBJECT IDENTIFIER ::= { flintstones 5 }
    fprocess     OBJECT IDENTIFIER ::= { flintstones 6 }
    fconfig      OBJECT IDENTIFIER ::= { flintstones 7 }

    fsysUsers OBJECT-TYPE
        SYNTAX DisplayString
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "List of users on the flintstone machine
            READ-COMMAND: /usr/bin/users; exit 0
            READ-COMMAND-TIMEOUT: 5"
        ::= { fsys 1 }

    fmemory OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
            "Amount of memory (in megabytes) on system
            APPEND-COMMUNITY-NAME: true
            FILE-NAME: /opt/OV/prg_samples/eagent/memory"
        ::= { fsys 2 }

    fmailNumMsgs OBJECT-TYPE
        SYNTAX INTEGER
        ACCESS read-only
        STATUS mandatory
```

```

DESCRIPTION
    "Message count on the mail queue.
    READ-COMMAND: /usr/bin/mailq | fgrep -v Mail
    | wc -l"
 ::= { fmail 1 }

fmailListMsgs OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "List of messages on the mail queue.
        READ-COMMAND: /usr/bin/mailq
        READ-COMMAND-TIMEOUT: 10"
 ::= { fmail 2 }

fwidgetsPerHour OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Number of widgets produced per hour.
        READ-COMMAND: /opt/OV/prg_samples/eagent/num_widgets $i
        $c $o $s
        READ-COMMAND-TIMEOUT: 2
        WRITE-COMMAND:
/opt/OV/prg_samples/eagent/change_num_widgets $*
        WRITE-COMMAND-TIMEOUT: 10"
 ::= { fwidgets 1 }

fprintersStatus OBJECT-TYPE
    SYNTAX INTEGER {
        up(1),
        down(2)
    }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Status of printer scheduler.
        READ-COMMAND: ps -ef|grep lpsched | grep -v
        grep |wc | awk '{ if ($1 == 0) print 2;
        else print 1 }'"
 ::= { fprinters 1 }

fdefaultPrinter OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Default printer
        FILE-NAME: /usr/spool/lp/default"
 ::= { fprinters 2 }

fUserDiskTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FuserDiskEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION

```

```

        "List of users and the number of kilobytes in their
        home directory.
        FILE-NAME:
/opt/OV/prg_samples/eagent/user_disk_space"
        ::= { fdisk 1 }

fUserDiskEntry OBJECT-TYPE
    ACCESS not-accessible
    SYNTAX FuserDiskEntry
    STATUS mandatory
    DESCRIPTION
        "This macro documents the column that uniquely
        describes each row."
        INDEX { fUID }
        ::= { fUserDiskTable 1 }

FUserDiskEntry ::=
    SEQUENCE {
        fUID INTEGER,
        fUserName DisplayString,
        fUserDiskSpace INTEGER,
        fUserEmailAddress DisplayString
    }

fUID OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "User's unique ID"
        ::= { fUserDiskEntry 1 }

fUserName OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "User login name"
        ::= { fUserDiskEntry 2 }

fUserDiskSpace OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Amount of disk space (in kilobytes) used by the
        user."
        ::= { fUserDiskEntry 3 }

fUserEmailAddress OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Email address for the user"
        ::= { fUserDiskEntry 4 }

fUserRootProcessTable OBJECT-TYPE

```

```

SYNTAX SEQUENCE OF FuserRootProcessEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "List of root processes. Do not execute command
    more than every 60 seconds.
    FILE-COMMAND:
/opt/OV/prg_samples/eagent/get_processes
    FILE-COMMAND-FREQUENCY: 60
    FILE-NAME:
/opt/OV/prg_samples/eagent/root_processes"
 ::= { fprocess 1 }

fUserRootProcessEntry OBJECT-TYPE
SYNTAX FuserRootProcessEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
    "This macro documents the column that uniquely
    describes each row."
INDEX { fProcessID }
 ::= { fUserRootProcessTable 1 }

FUserRootProcessEntry ::=
SEQUENCE {
    fProcessID INTEGER,
    fProcessName DisplayString
}

fProcessID OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Process ID"
 ::= { fUserRootProcessEntry 1 }

fProcessName OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "Name of process"
 ::= { fUserRootProcessEntry 2 }

fInetdConf OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "The configuration file for inetd
    FILE-COMMAND:
/opt/OV/prg_samples/eagent/update_inetd $r
    FILE-COMMAND-FREQUENCY: 7200
    FILE-NAME: /etc/inetd.conf"
 ::= { fconfig 1 }

```

```
END
```

This subsection describes each MIB object definition that is included in the example of creating an `/etc/SnmpAgent.d/snmpd.extend` file.

Procedure

1. List users who are using the system

You can configure SNMP Agent to execute a command when it receives a `GetRequest` from a manager. To do so, specify the command to be executed after the `READ-COMMAND` label. For example, the following code fragment makes SNMP Agent execute the `/usr/bin/users` command to obtain a list of users on the system as a MIB value:

```
fsysUsers OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "List of users on the flintstone machine
        READ-COMMAND: /usr/bin/users; exit 0
        READ-COMMAND-TIMEOUT: 5"
    ::= { fsys 1 }
```

When SNMP Agent receives from a manager a `GetRequest` with object identifier `flintstones.fsys.fsysUsers.0` specified, SNMP Agent executes the `/usr/bin/users` command (specified after the `READ-COMMAND` label) and returns the result to the manager.

By default, SNMP Agent returns a `genErr` error to the manager if the command specified after the `READ-COMMAND` label does not return a response within 3 seconds. You can change this command completion wait time to any value by using the `READ-COMMAND-TIMEOUT` label. If, for example, you want to change the command completion wait time to 5 seconds, write the following `READ-COMMAND-TIMEOUT` line:

```
READ-COMMAND-TIMEOUT: 5
```

2. Reference the contents of memory on each machine

You can use the Extensible SNMP Agent to create a proxy. The agent can respond to objects on behalf of another system, device, or application.

For example, if you want the agent to respond with the amount of memory for three systems that do not support SNMP, the agent can act as a proxy for those other systems. The three systems that do not support SNMP are named `larry`, `curly`, and `moe`. The following three files contain the amount of memory on each system:

```
/opt/OV/prg_samples/eagent/memory.larry
/opt/OV/prg_samples/eagent/memory.curly
/opt/OV/prg_samples/eagent/memory.moe
```

To implement this proxy, you would define the following object:

```
fmemory OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Amount of memory (in megabytes) on system
        APPEND-COMMUNITY-NAME: true
        FILE-NAME:
```

```
        /opt/OV/prg_samples/eagent/memory"
 ::= { fsys 2 }
```

The agent will respond on behalf of `larry`, `curly`, and `moe` for the object `flintstones.fsys.fmemory`. The community name in the request indicates the system of interest. Suppose that SNMP Agent receives a `GetRequest` for the object with object identifier `flintstones.fsys.fmemory.0` and that the community name specified in this request is `moe`. In this case, SNMP Agent reads `/opt/OV/prg_samples/eagent/memory.moe` and returns the value in the file to the manager. As shown in the code sample, you must specify `APPEND-COMMUNITY-NAME: true` to use SNMP Agent as a proxy. If you specify `APPEND-COMMUNITY-NAME: true`, SNMP Agent reads or writes the file whose file name is `file_name` followed by the community name specified in the request (in this example, the resulting file name is `/opt/OV/prg_samples/eagent/memory.community-name`). You can also set SNMP Agent so that it acts as a proxy for an object (for example, a MIB object) that already exists on SNMP Agent. The example below shows how to make proxy machines `larry`, `curly`, and `moe` return their own `sysDescr` value. Each proxy machine has its own `sysDescr` value in the following file:

If SNMP Agent receives from a manager a `GetRequest` for the object with object identifier `system.sysDescr.0` and the community name specified in this request is `public`, SNMP Agent returns the `sysDescr` value of the MIB object.

```
/opt/OV/prg_samples/eagent/sysDescr.larry
/opt/OV/prg_samples/eagent/sysDescr.curly
/opt/OV/prg_samples/eagent/sysDescr.moe
```

To implement this proxy, you would define the following object:

```
sysDescr OBJECT-TYPE
    SYNTAX DisplayString (SIZE(0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "A textual description of the entity. This value
        should include the full name and version
        identification of the system's hardware type,
        software operating-system, and networking
        software. It is mandatory that this only contain
        printable ASCII characters.
        APPEND-COMMUNITY-NAME : true
        FILE-NAME:
            /opt/OV/prg_samples/eagent/sysDescr"
 ::= { system 1 }
```

If SNMP agent receives a `GetRequest` for the object ID `system.sysDescr.0` with community name `moe` from the manager, SNMP Agent reads `/opt/OV/prg_samples/eagent/sysDescr.moe` and returns the contents of the file to the manager. If SNMP Agent receives a `GetRequest` for the object ID `system.sysDescr.0` with community name `public` from the manager, SNMP Agent returns the `sysDescr` value of the MIB object.

3. Manage the mail queue

The example below shows how to specify the `READ-COMMAND` line to manage the mail queue.

```
fmailNumMsgs OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
```

```

                "Message count on the mail queue.
                READ-COMMAND: /usr/bin/mailq
                | fgrep -v Mail |wc -l"
 ::= { fmail 1 }

```

```

fmailListMsgs OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "List of messages on the mail queue.
        READ-COMMAND: /usr/bin/mailq
        READ-COMMAND-TIMEOUT: 10"
 ::= { fmail 2 }

```

4. Control the number of widgets that the system produces per hour while unattended

You can configure SNMP Agent to execute a command when it receives a SetRequest from a manager. To do so, specify the command to be executed after the WRITE-COMMAND label.

For example, to use the /opt/OV/prg_samples/eagent/change_num_widgets command to change the number of widgets that the system produces per hour while unattended, specify the following:

```

fwidgetsPerHour OBJECT-TYPE
    SYNTAX Gauge
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Number of widgets produced per hour.
        READ-COMMAND: /opt/OV/prg_samples/eagent
        /num_widgets $i $c $o $s
        READ-COMMAND-TIMEOUT: 2
        WRITE-COMMAND: /opt/OV/prg_samples/eagent
        /change_num_widgets $*
        WRITE-COMMAND-TIMEOUT: 10"
 ::= { fwidgets 1 }

```

You can pass arguments to the commands you specified after the READ-COMMAND and WRITE-COMMAND label. To do so, specify the arguments after the command name. For the meanings of the arguments, see the argument descriptions in [2.10.3\(2\) Writing a script or program](#).

5. Keep track of the LP scheduler

The example below shows how to specify the READ-COMMAND line to monitor the status of the LP scheduler.

```

fprintersStatus OBJECT-TYPE
    SYNTAX INTEGER {
        up(1),
        down(2)
    }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Status of printer scheduler.
        READ-COMMAND: ps -ef|grep lpsched
        | grep -v grep | wc |
        awk '{ if ($1 == 0) print 2; else print
        1}'"
 ::= { fprinters 1 }

```

6. Reference the default printer

You can configure SNMP Agent to create an extended MIB object using a file as source of the MIB value. To do so, specify the name of the file after the `FILE-NAME` label.

To create an object that reads the `/usr/spool/lp/default` file to reference the default printer, specify the following:

```
fdefaultPrinter OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Default printer
        FILE-NAME: /usr/spool/lp/default"
    ::= { fprinters 2 }
```

When SNMP Agent receives from a manager a `GetRequest` for the object with object identifier `flintstones.fprinters.fdefaultPrinter.0`, SNMP Agent reads the `/usr/spool/lp/default` file specified after the `FILE-NAME` label and returns the value in the file to the manager.

7. Reference the user IDs, the amount of disk space used by each user, and the users' e-mail address list

You can add MIB objects in table format to SNMP Agent by specifying a specific value in a file as the MIB value. An example is given below.

A list of user IDs, the amount of disk space used by each user, and their e-mail addresses are stored in the `/opt/OV/prg_samples/eagent/user_disk_space` file. The following is an example of this data:

#	User ID	User Name	Disk Space	Email Address
100		zach	120	zach@server1
201		alice	65	alice@server2
320		john	2	john@server3
119		craig	500	root@server1
217		steve	75	steve@server1
83		bob	111	bob@bobby

This table has four columns and six rows. Every table defined using the Extensible SNMP Agent must have a column or a set of columns that uniquely define the row. In some models, this column would be called a key. In this example, the first column is unique. The `User ID` is unique on this system. If the `User Names` are unique, the second column could be used as the key.

```
fUserDiskTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FuserDiskEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "List of users and the number of kilobytes
        in their home directory.
        FILE-NAME: /opt/OV/prg_samples/eagent/
        user_disk_space"
    ::= { fdisk 1 }

fUserDiskEntry OBJECT-TYPE
    SYNTAX FuserDiskEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "This macro documents the column that
        uniquely describes each row."
```



```

        INDEX { fUID }
        ::= { fUserDiskTable 1 }
FUserDiskEntry ::=
    SEQUENCE {
        fUID INTEGER,
        fUserName DisplayString,
        fUserDiskSpace INTEGER,
        fUserEmailAddress DisplayString
    }

fUID OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "User's unique ID"
    ::= { fUserDiskEntry 1 }
fUserName OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "User login name"
    ::= { fUserDiskEntry 2 }

fUserDiskSpace OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Amount of disk space (in kilobytes) used by
        the user."
    ::= { fUserDiskEntry 3 }

fUserEmailAddress OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Email address for the user"
    ::= { fUserDiskEntry 4 }

```

The first OBJECT-TYPE macro, `fUserDiskTable`, describes the file name associated with the object. The second OBJECT-TYPE macro, `fUserDiskEntry`, describes the column that uniquely identifies a row. The next entry, `FUserDiskEntry`, is for documentation purposes. This entry lists the columns in the table.

The last four OBJECT-TYPE macros define individual columns.

If the agent receives a `GetNextRequest` for `fUserDiskTable.fUserDiskEntry.fUID`, the agent will read the entire file `/opt/OV/prg_samples/eagent/user_disk_space`. The agent then sorts the table based on the object specified in the `INDEX` clause. The sorted table will then look like this:

83	bob	111	bob@bobby
100	zach	120	zach@server1
119	craig	500	root@server1
201	alice	65	alice@server2
217	steve	75	steve@server1
320	john	2	john@server3

As a result, SNMP Agent returns the first value in the table. This is the value in the first column of the first row. The manager receives 83 as the MIB value of the object ID `flintstones.fdisk.fUserDiskTable.fUserDiskEntry.fUID.83`. Next, if SNMP Agent receives a `GetNextRequest` for the object ID `flintstones.fdisk.fUserDiskTable.fUserDiskEntry.fUID.83`, SNMP Agent checks to see if the file has been updated. If the file has been updated, SNMP Agent reloads the `/opt/OV/prg_samples/eagent/user_disk_space` file and then returns the User ID in row 2 as the MIB value. If SNMP Agent receives a `GetNextRequest` for the object ID `flintstones.fdisk.fUserDiskTable.fUserDiskEntry.fUID.320`, SNMP Agent returns the first value in column 2 because there is no more User ID row. The object ID is `flintstones.fdisk.fUserDiskTable.fUserDiskEntry.fUserName.83` and the MIB value is bob.

If SNMP Agent receives a `GetRequest` for the object ID `flintstones.fdisk.fUserDiskTable.fUserDiskEntry.fUserEmailAddress.217`, SNMP Agent searches row 217 of the Email Address column and returns `steve@server1` as the MIB value.

To change the e-mail address of alice from `alice@server2` to `alice@mailier`, issue a `SetRequest` specifying `flintstones.fdisk.fUserDiskTable.fUserDiskEntry.fUserEmailAddress.201` as the object ID and `alice@mailier` as the MIB value. SNMP Agent replaces the value in the table in the `/opt/OV/prg_samples/eagent/user_disk_space` file with the specified value.

8. List root processes

You can configure SNMP Agent to execute a UNIX command before reading a file. To do so, specify the UNIX command after the `FILE-COMMAND` label in the `DESCRIPTION` clause in the `/etc/SnmpAgent.d/snmpd.extend` file. The example below shows how to specify the `FILE-COMMAND` line to output a list of root processes.

```
fUserRootProcessTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FuserRootProcessEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "List of root processes. Do not execute
        command more than every 60 seconds.
        FILE-COMMAND: /opt/OV/prg_samples/eagent
        FILE-COMMAND-FREQUENCY: 60
        FILE-NAME: /opt/OV/prg_samples/eagent
        /root_processes"
    ::= { fprocess 1 }

fUserRootProcessEntry OBJECT-TYPE
    SYNTAX FuserRootProcessEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "This macro documents the column that
        uniquely describes each row."
    INDEX { fProcessID }
    ::= { fUserRootProcessTable 1 }

FUserRootProcessEntry ::=
    SEQUENCE {
        fProcessID INTEGER,
        fProcessName DisplayString
    }
```

```

fProcessID OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Process ID"
 ::= { fUserRootProcessEntry 1 }

fProcessName OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Name of process"
 ::= { fUserRootProcessEntry 2 }

```

When SNMP Agent receives from a manager the GetRequest shown below, SNMP Agent first executes the `/opt/OV/prg_samples/eagent/get_process` command specified after the FILE-COMMAND label.

```

fprocess.fuserRootProcessTable.fUserRootProcessEntry.fProcessID
fprocess.fuserRootProcessTable.fUserRootProcessEntry.fProcessName

```

The `/opt/OV/prg_samples/eagent/get_process` command creates an `/opt/OV/prg_samples/eagent/root_process` file that contains the IDs and names of the root processes. Upon the completion of command execution, SNMP Agent reads the `/opt/OV/prg_samples/eagent/root_process` file, sorts the table, and returns the value in the first column of the first row to the manager.

By default, SNMP Agent does not execute the `/opt/OV/prg_samples/eagent/get_process` command in response to a GetRequest from a manager if the time that elapsed since SNMP Agent last executed the `/opt/OV/prg_samples/eagent/get_process` command is 10 seconds or fewer. SNMP Agent returns to the manager the value in the file read during the previous execution. You can change this default interval to any value using the FILE-COMMAND-FREQUENCY label. For example, to make SNMP Agent execute the command hourly, specify the FILE-COMMAND-FREQUENCY line as follows:

```

FILE-COMMAND-FREQUENCY: 3600

```

You can configure SNMP Agent to communicate with a UNIX process instead of the UNIX command specified after the FILE-COMMAND label. To do so, write the PIPE-IN-NAME and PIPE-OUT-NAME lines of the DESCRIPTION clause in the `/etc/SnmpAgent.d/snmpd.extend` file. SNMP Agent writes data into the file specified in the PIPE-OUT-NAME line before reading the file specified in the FILE-NAME line. The UNIX process reads the data, changes the value in the file specified in the FILE-NAME line, and notifies SNMP Agent of the completion using the file specified in the PIPE-IN-NAME line. Like the example for specifying the FILE-COMMAND label, the following example shows how to specify the PIPE-IN-NAME and PIPE-OUT-NAME labels to output a list of root processes. Note that any definitions identical to those in the FILE-COMMAND example are omitted.

```

fUserRootProcessTable OBJECT-TYPE
    SYNTAX SEQUENCE OF FuserRootProcessEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "List of root processes.
        PIPE-IN-NAME: /tmp/fifo_in
        PIPE-OUT-NAME: /tmp/fifo_out
        FILE-NAME: /opt/OV/prg_samples/eagent/

```

```
root_processes"
 ::= { fprocess 1 }
```

When SNMP Agent receives from a manager a GetNextRequest for the object shown below, SNMP Agent first writes data into the `/tmp/fifo_out` file.

```
fprocess.fUserRootProcessTable.fUserRootProcessEntry.fProcessID
fprocess.fUserRootProcessTable.fUserRootProcessEntry.fProcessName
```

For details about the data, see the description of the data that is passed to `pipe_out_name` in [2.10.2\(2\) Extended MIB objects in table format](#).

The UNIX process reads this request message and creates an `/opt/OV/prg_samples/eagent/root_processes` file. The created file contains the IDs and names of the root processes. The UNIX process then writes 0 into the `/tmp/fifo_in` file and notifies SNMP Agent that it has successfully created the file. SNMP Agent reads the `/tmp/fifo_in` file and checks whether 0 is set. If 0 is set, the agent reads the `/opt/OV/prg_samples/eagent/root_processes` file, sorts the table in the file, and returns to the manager the value in the first column of the first row of the table.

By default, SNMP Agent does not write any data into the file specified after the `PIPE-OUT-NAME` label in response to a GetRequest or GetNextRequest from a manager if the time that elapsed since SNMP Agent last wrote the file is 10 seconds or fewer. SNMP Agent returns to the manager the value in the file it read previously. You can change this default interval to any value using the `PIPE-FREQUENCY` label.

If SNMP Agent receives a SetRequest from a manager, Agent writes data into the file specified after the `PIPE-OUT-NAME` label before and after reading the file specified in the `FILE-NAME` line. The UNIX process receives data from the file specified after the `PIPE-OUT-NAME` label, performs the specified action, and writes 0 into the file specified after the `PIPE-IN-NAME` label.

9. Modify the `inetd(1M)`'s configuration file

If you specify a UNIX command after the `FILE-COMMAND` label, SNMP Agent acts as follows: Upon receiving a SetRequest from a manager, SNMP Agent executes the specified UNIX command before and after modifying the file specified after the `FILE-NAME` label. The sample code below uses this feature to modify the `inetd(1M)` configuration file.

```
fInetdConf OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The configuration file for inetd
        FILE-COMMAND: /opt/OV/prg_samples/eagent
        /update_inetd $r
        FILE-COMMAND-FREQUENCY: 7200
        FILE-NAME: /etc/inetd.conf"
 ::= { fconfig 1 }
```

Suppose that SNMP Agent receives from a manager a SetRequest for the object with object identifier `flintstones.fconfig.fInetdConf.0` and that the MIB value specified in the request is a value in the `inetd.conf` configuration file. In this case, SNMP Agent performs the following procedure:

- Execute the `/opt/OV/prg_samples/eagent/update_inetd SetRequest` command.
- Modify the Set value in the `/etc/inetd.conf` file to the Set value specified in the SetRequest.
- Execute the `/opt/OV/prg_samples/eagent/update_inetd PostSetRequest` command.

The `/opt/OV/prg_samples/eagent/update_inetd` command then checks the first argument. If the first argument is a `PostSetRequest`, the `/opt/OV/prg_samples/eagent/update_inetd` command executes the `/etc/inetd -c` command. This causes `/etc/inetd` to re-read the configuration file.

2.11 Defining enterprise-specific traps

This section describes how to use the `snmptrap` command to send SNMP traps from SNMP Agent to the manager. The topics discussed here are as follows:

- How to define enterprise-specific traps
- How to use enterprise-specific traps
- Sample script

To configure your agent to send enterprise-specific traps, you must first understand what traps are. For details about traps, see [1.2.2 Issuing SNMP traps](#) and *RFC 1157: A Simple Network Management Protocol (SNMP)*.

2.11.1 How to define enterprise-specific traps

To define your own enterprise-specific trap, you need to uniquely identify your trap. You do so by combining the generic enterpriseSpecific trap 6 with your own specific trap number. The maximum specific trap number is $2^{32}-1$. This combination tells the manager what kind of trap it is.

Optionally, you can pass along data.

2.11.2 How to use enterprise-specific traps

The agent sends the traps using the `snmptrap` command. For example, you can configure your agent to send traps by executing the `snmptrap` command from a shell script.

As a manager, you have two alternatives when monitoring the status of an agent. You can

- Continuously poll the agent from the manager to get information.
- Send a trap from the agent to the manager.

Polling creates a lot of traffic on the network and, if an event occurs shortly after polling has taken place, the manager might not find out about an event for an extended period of time. The key benefits of using the `snmptrap` command are that you can decrease the amount of SNMP traffic on the network and that you can find out about an event sooner.

If you have NNM, you can customize your environment by using the `snmptrap` command in conjunction with the **Event Configuration** operation. For example, assume that you have written a script on an agent that executes the `snmptrap` command when a particular process on the agent goes down. You can then use the **Event Configuration** operation from the NNM station to take an action when the manager receives that particular trap from the agent.

2.11.3 Sample script

Assume that you are responsible for managing the printers on your network. Your goal is to write a script that executes the `snmptrap` command when the printer scheduler goes down. Here is an example script.

```
#!/bin/sh
#
#
```

```

# This script checks to see if the printer scheduler
# (lpshed) is running. This check is performed every
# hour. If the scheduler is not running, the agent
# sends an SNMP trap to the management station.

#
# If a management station receives a trap from a system with
# enterprise equal to .1.3.6.1.4.1.4242, generic-trap equal
# to 6, and the specific trap equal to 2, the management station
# knows that the printer scheduler for that agent-addr is down.

# The agent sends how many hours the lp scheduler has been
# down with the trap.
#

AGENT_ADDRESS=15.6.71.223

MGMT_STATION=flcndmak

hours=0
while true
do
    sleep 3600
    pid='ps -ef | grep lpsched | grep -v grep |wc -l'
    if [$pid -eq 0]
    then
        hours='expr $hours + 1'
        snmptrap -cpublic $MGMT_STATION .1.3.6.1.4.1.4242
        $AGENT_ADDRESS 6 2 0 \
            .1.3.6.1.4.1.4242.4.2.0 Integer $hours
    else
        hours=0
    fi
done

```

Important

In AIX, if the `snmptrap` or `systemtrap` command is executed as an extension of a shell script or program started from `cron` or `/etc/inittab`, the command might fail with the following message:

```
snmptrap:cannot set locale($LANG="Ja_JP")
```

If this message is output, in the `LC_ALL` environment variable, set the language you want to use.

The following shows an example of setting `C` as the language code for the `B` shell.

```

LC_ALL=C
export LC_ALL
snmptrap flcndmak .1.3.6.1.4.1.4242 15.6.71.223 6 2 0

```

2.12 Settings for operations in a cluster environment

SNMP Agent supports operations in a cluster environment.

The operations management server manages the primary and secondary servers as individual servers (nodes). Therefore, SNMP Agent keeps both the primary and secondary servers active. This means that no failover setting is required.

To run SNMP Agent in a cluster system, you must specify the following settings:

2.12.1 Required settings for monitoring shared disks (for Linux)

When the SNMP agent is installed in a cluster system, file system information about the shared disk might not be periodically acquired from the manager. This problem occurs because the target file system is not in the file `/etc/fstab`. The SNMP agent uses OS system calls to acquire file system information. These OS system calls acquire information for only the file systems in the file `/etc/fstab` in default setting. The solution to this problem is to add the information about the file system on the shared disk to the file `/etc/fstab`, or to configure `/etc/SnmpAgent.d/esafstab.conf`.

(1) Configuring the `/etc/fstab`

When the shared disk is monitored, the SNMP agent recognizes only the first field (block special devices) and the second field (mount points) among the various fields in the information about the file system on the shared disk in the file `/etc/fstab`.

This product does not recognize any other fields. Therefore, for details about how to set values for the other fields and how to check the values set for the other fields related to the shared disk in the file `/etc/fstab`, see the cluster software and OS documentation.

For example, if the cluster software name is HA monitor, the mount point of the shared disk is `/mnt/test`, and as a requirement for managing the shared disk, the shared disk is set to not be automatically mounted when the OS starts, the example setting in the file `/etc/fstab` is as follows.

<Setting example>

```
/dev/sdb1 /mnt/test ext3 defaults,noauto 0 0
```

The above setting is an example for HA monitor cluster software. Before performing this setting, see the manuals and Release Notes for HA monitor.

If you use some other cluster software, perform the appropriate settings in the file `/etc/fstab`, according to the requirements of that cluster software and OS environment.

(2) Configuring `/etc/SnmpAgent.d/esafstab.conf`

The SNMP agent gets fileSystem64 group information from not only `/etc/fstab` but also `/etc/SnmpAgent.d/esafstab.conf` when configurations are added to `/etc/SnmpAgent.d/esafstab.conf`. You can add the first field (block special device) and the second field (mount point) of `/etc/fstab` to `/etc/SnmpAgent.d/esafstab.conf`. For details about the `/etc/SnmpAgent.d/esafstab.conf` file, see [Adding file system target definition file\(esafstab.conf\)](#) in [Chapter 6. Definition Files](#).

Perform the following operations as a superuser.

Procedure

1. Open the `/etc/SnmpAgent.d/esafstab.conf` file in a text editor.
2. Add the following line.

Example

A mount point of a shared disk is `/mnt/test`, and the special block device is `/dev/sdb1`.

```
/dev/sdb1 /mnt/test
```

3. Restart the SNMP agent.

The SNMP agent reads the `/etc/SnmpAgent.d/esafstab.conf` when starting. At this time, if there are syntax errors in `/etc/SnmpAgent.d/esafstab`, it ignores the lines and starts. The SNMP agent outputs syntax errors to `/etc/SnmpAgent.d/esafstab.conf.err`.

2.12.2 Settings for suppressing an invalid shared disk capacity response (for AIX and Linux)

If SNMP Agent receives a MIB acquisition request while the shared disk is not mounted, it returns an invalid shared disk capacity as the response. To suppress this response, you must specify `/etc/SnmpAgent.d/esafsys.conf`. The following shows a coding example of `/etc/SnmpAgent.d/esafsys.conf`.

Example

This example specifies `/mnt/test` as the shared disk and `esatest` as the file.

```
check: /mnt/test esatest
```

In AIX and Linux, if SNMP Agent acquires a shared disk's file system information from a node where no logical host exists in a cluster system, it might return invalid information. This is because the target file system is not mounted. You can check whether the file system targeted by SNMP Agent is mounted, and if no file system is mounted, you can prevent file system information from being returned.

For details about file systems, see [4.2.2\(2\) fileSystem group](#) and [4.3.2\(20\) fileSystem64 group](#).

Perform the following operations as a superuser.

Procedure

1. Open the `/etc/SnmpAgent.d/esafsys.conf` file in a text editor.
2. Add the following line to the last line of the `/etc/SnmpAgent.d/esafsys.conf` file:
`check: shared-disk-file-system-path-name name-of-applicable-file-located-immediately-under-shared-disk`

Example

To monitor shared disks `/shdisk1` and `/shdisk2`:

Files `test1` and `test2` are located immediately under `/shdisk1` and `/shdisk2`, respectively.

```
check: /shdisk1 test1
check: /shdisk2 test2
```

3. Restart SNMP Agent.

During startup, SNMP Agent loads `/etc/SnmpAgent.d/esafilesys.conf`. During this step, if the `/etc/SnmpAgent.d/esafilesys.conf` file contains a syntax error, SNMP Agent ignores the line containing the error and starts. The syntax error details are output to `/etc/SnmpAgent.d/esafilesys.conf.err`.

2.12.3 Settings for using PowerHA (HACMP)

When SNMP Agent is executing in a PowerHA (HACMP) environment, change SNMP Agent's SNMP request reception port from 161/udp to a free UDP port to allow the AIX `snmpd` process to use 161/udp.

The reason for this change is that, in the PowerHA (HACMP) environment, AIX processes exchange information using SNMP requests.

In the following example, SNMP Agent's SNMP reception port is changed to 8161/udp.

(1) Change the NNMi or NNM SNMP request port.

For details about changing the SNMP request port, see Help for NNMi or Help for NNM, depending on which product you are using.

(2) Change the SNMP reception port in SNMP Agent.

Perform the following operations as a superuser.

Procedure

1. Stop SNMP Agent.

Execute `/usr/CM2/ESA/bin/snmpstop` with no arguments.

2. Change the SNMP reception port for AIX `snmpd`.

- If the native agent is an SNMP v1 agent

Confirm that the value in the `snmp` column in the `/etc/services` file is 161/udp, and if it is not, change it to 161/udp.

- If the native agent is an SNMP v3 agent

Change `/usr/CM2/ESA/opt/SnmpNative` as follows:

```
SNMP_NATIVE_OPTIONS="-p 161"
```

- If `SNMP_PORT=` is defined in the `/etc/environment` file, change it to the following:

```
SNMP_PORT=161
```

3. Change SNMP Agent's SNMP reception port.

In the example below, SNMP Agent's SNMP reception port is changed to 8161/udp.

The port number does not have to be 8161/udp, this is just the value used in this example.

Perform, the following operation as a superuser:

Add the following two lines to `/usr/CM2/ESA/opt/SnmpMaster`.

```
SR_SNMP_TEST_PORT=8161
export SR_SNMP_TEST_PORT
```

4. Change the SNMP request transmission port for SNMP Agent's native agent adapter.

Edit the file `/usr/CM2/ESA/opt/SnmpNaa` while logged on as a superuser.

Add the following two lines to the end of the file:

```
SNMP_NAA_OPTIONS="-port 161 -aperror -apwarn -apverbose -hexdump -vbdump"  
export SNMP_NAA_OPTIONS
```

To enable the above settings, start SNMP Agent by executing the `snmpstart` command with no arguments.

```
/usr/CM2/ESA/bin/snmpstart
```

2.13 Notes about the amount of free space in physical memory

This section provides notes about the amount of free space in physical memory.

For details about the amount of free space in physical memory, see [4.2.2\(1\) computerSystem group](#) and [4.3.2\(23\) computerSystem64 group](#).

- Solaris physical memory

Solaris allows physical memory to be used not just for ordinary program operations, but also as a file cache (buffer cache). In this case, the portion of physical memory that can be used as a reusable file cache is not freed as soon as its current use ends. Instead, it is retained as a cache in case the file is referenced again. Therefore, after the system has run continuously for a certain period of time, the values of `computerSystemFreeMemory` and `computerSystem64FreeMemory` converge toward a fixed value (which depends on the system). Because of this, we recommend that users who want to check either the free space in system memory or memory usage in Solaris monitor virtual memory (swap space) instead of physical memory. The objects `computerSystemSwapConfig`, `computerSystemFreeSwap`, `computerSystem64SwapConfig`, and `computerSystem64FreeSwap` are useful for monitoring virtual memory (swap space). Use `computerSystemFreeMemory` and `computerSystem64FreeMemory` to monitor the amount of free space in physical memory that includes the file cache.

- The amount of free space in physical memory in AIX

In AIX, file access speed improves when physical memory is used as a file cache. The file cache is therefore included in the amount of physical memory in use, and the amount of free space in physical memory obtained by SNMP Agent is the actual amount of free memory that is available.

If you want to determine the amount of free memory capacity while excluding the file cache from the amount of physical memory in use, set the environment variable `SNMP_HTC_AIX_EXCEPT_FILECACHE` to `Y` in the environment variable definition file `Snmphpunix`. Then, the amount of free space in physical memory obtained by SNMP Agent is the sum of the free memory capacity and the file cache value.

Note that the environment variable `SNMP_HTC_AIX_EXCEPT_FILECACHE` that is set in the environment variable definition file `Snmphpunix` can be used for both `computerSystemFreeMemory` and `computerSystem64FreeMemory`.

For details about the path of the environment variable definition file, see [Appendix A. SNMP Agent Files](#).

The following shows a specification example for the `SNMP_HTC_AIX_EXCEPT_FILECACHE` environment variable.

Example

```
SNMP_HTC_AIX_EXCEPT_FILECACHE=Y
export SNMP_HTC_AIX_EXCEPT_FILECACHE
```

- The amount of free space in physical memory in Linux

Linux actively allocates free memory to buffer memory and cache memory. If a memory allocation request is issued from an application, buffer memory and cache memory are freed as necessary, and memory is allocated to the application. Therefore, by default, the amount of free memory in the physical memory acquired by SNMP Agent is the sum of the amount of free memory, buffer memory, and cache memory.

If you specify `Y` for the environment variable `SNMP_HTC_LINUX_INACTIVE_MEM` in the environment variable definition file `Snmphpunix`, SNMP Agent will acquire the sum of the amount of free memory, inactive buffer memory, and inactive cache memory as the amount of free memory in the physical memory.

Note that the environment variable `SNMP_HTC_LINUX_INACTIVE_MEM` that is set in the environment variable definition file `Snmphpunix` can be used for both `computerSystemFreeMemory` and `computerSystem64FreeMemory`.

For details about the path of the environment variable definition file, see [Appendix A. SNMP Agent Files](#).

The following shows a specification example for the `SNMP_HTC_LINUX_INACTIVE_MEM` environment variable.

Example

```
SNMP_HTC_LINUX_INACTIVE_MEM=Y
export SNMP_HTC_LINUX_INACTIVE_MEM
```

2.14 Notes about swap space size

This section provides notes about swap space size.

For details about swap space size, see [4.2.2\(1\) computerSystem group](#) and [4.3.2\(23\) computerSystem64 group](#).

By default, the Solaris device swap space size acquired by SNMP Agent does not include the reserved value. To acquire the device swap space size including the reserved value, specify Y for the `SNMP_HTC_SOLARIS_SWAP_RESERVED` environment variable in the `SnmpHpunix` file.

The environment variable `SNMP_HTC_SOLARIS_SWAP_RESERVED` that is set in the environment variable definition file `SnmpHpunix` can be used for `computerSystemSwapConfig`, `computerSystem64SwapConfig`, and `computerSystem64EnabledSwap`.

The reserved value is the amount of unallocated swap space retained in memory for later use.

The following shows a specification example for the `SNMP_HTC_SOLARIS_SWAP_RESERVED` environment variable.

Example

```
SNMP_HTC_SOLARIS_SWAP_RESERVED=Y
export SNMP_HTC_SOLARIS_SWAP_RESERVED
```

2.15 Notes about CPU information

This section provides notes about CPU information.

For details about CPU time information, see [4.2.2\(1\) computerSystem group](#) and [4.3.2\(23\) computerSystem64 group](#).

For details about CPU usage rate information, see [4.3.2\(11\) cpuUtil group](#).

For details about CPU information in HP-UX (IPF), see [4.3.2\(16\) processor64 group](#).

- By default, SNMP Agent of HP-UX (IPF) acquires all available information about processors on the OS, regardless of whether a processor is enabled or disabled.

If you set the `SNMP_HTC_HPUX_ENABLE_PROCESSOR` environment variable to `Y` in the environment variable definition file `SnmpHpunix2`, SNMP Agent only acquires information about processors that are enabled. For details about the path of the environment variable definition file, see [Appendix A. SNMP Agent Files](#). The following shows a specification example for the `SNMP_HTC_HPUX_ENABLE_PROCESSOR` environment variable.

Example

```
SNMP_HTC_HPUX_ENABLE_PROCESSOR=Y
export SNMP_HTC_HPUX_ENABLE_PROCESSOR
```

- In Solaris, AIX, and Linux, the CPU usage rate information is updated at the CPU usage rate acquisition interval that is set in SNMP Agent (the default is 5 minutes). Therefore, when you collect CPU usage rates, set a collection interval that is greater than the CPU usage rate acquisition interval.

The CPU usage rate acquisition interval (in minutes) is set in the `-i` option of the `htc_monagt1` process, which is a daemon process that regularly collects CPU usage rates. The range for this interval is from 0 to 1,440. If 0 is specified, CPU usage rate information is not collected. In Solaris, AIX, and Linux, during the interval between the start of SNMP Agent and the acquisition of the first CPU usage rate, all the MIB values in the CPU usage rate information are returned as a `noSuchName` error.

- In Solaris and AIX, CPU information is updated at the CPU time acquisition interval that is set in SNMP Agent (the default is 5 minutes). Therefore, when you collect the CPU time, set a collection interval that is greater than the CPU time acquisition interval.

This CPU time acquisition interval (in minutes) is set in the `-s` option of the `htc_monagt1` process, which is a daemon process that regularly collects CPU time. The range for this interval is from 0 to 1,440. If 0 is specified, CPU time information is not collected. In Solaris and AIX, during the interval between the start of SNMP Agent and the acquisition of the first CPU time information, all the MIB values in the CPU usage rate information are returned as a `noSuchName` error.

- In Solaris, the online/offline status of the CPU might change. If the status changes, CPU information cannot be obtained from the OS, and as a result, all the MIB values in the CPU usage rate information (except for `cpuUtilInterval`) are temporarily returned as a `noSuchName` error.

However, once the CPU time acquisition interval and CPU usage rate acquisition interval pass, making it possible to acquire the CPU information from the OS, all the MIB values can be obtained.

Even when the online/offline status of the CPU changes, the CPU number to be acquired does not change.

Note that the value of CPU time information is reset when the CPU time information acquisition interval passes after the status of the CPU changes.

- In AIX, a CPU might be dynamically added or removed by DLPAR (Dynamic Logical Partition). After a CPU is dynamically added or removed, CPU information cannot be obtained from the OS, and as a result, all the MIB values in the CPU time information and CPU usage rate information (except for `cpuUtilInterval`) are temporarily returned as a `noSuchName` error. However, once the CPU time acquisition interval and CPU usage rate acquisition interval pass, making it possible to acquire CPU information from the OS, all the MIB values can be obtained.

Note that the value of CPU time information is reset when the CPU time information acquisition interval passes after a CPU is added or removed.

- In AIX, SNMP Agent acquires the CPU utilization rate, by default, by adding up the CPU utilization rates of individual CPUs, dividing the result by the number of CPUs, and then discarding decimals.

In an SMT environment, you can obtain the CPU utilization rate of the entire machine by specifying Y in the `SNMP_HTC_AIX_CPU_SMT` environment variable in the `SnmPhtcmonagt1` environment variable definition file. For details about the path of the environment variable definition file, see [Appendix A. SNMP Agent Files](#).

The following shows a specification example for the `SNMP_HTC_AIX_CPU_SMT` environment variable.

Example

```
SNMP_HTC_AIX_CPU_SMT=Y
export SNMP_HTC_AIX_CPU_SMT
```


2.16 Settings to prevent responses with information about file systems for which a response is not required (for Linux)

In Linux, SNMP Agent returns information about CD-ROMs and floppy disks, even if there is no CD-ROM or floppy disk mounted, because the system cannot determine based on the information from the OS whether such file systems are mounted. If a file system exists for which you do not need responses, follow the procedure below to prevent the sending of responses containing information about file systems targeted by SNMP Agent. For details about file systems, see [4.2.2\(2\) *fileSystem* group](#) and [4.3.2\(20\) *fileSystem64* group](#).

Perform the following operation as a superuser.

Procedure

1. Open the `/etc/SnmpAgent.d/esafilesys.conf` file with an editor.
2. Add the following line below to the last line of the `/etc/SnmpAgent.d/esafilesys.conf` file:

```
exclude: non-response-file-system-path-name
```

You must place one space between `exclude:` and *non-response-file-system-path-name*.

The maximum length of *non-response-file-system-path-name* is 1,024 characters.

Example

The following example shows how to prevent `/mnt/cdrom` and `/mnt/floppy` information from being returned.

```
exclude: /mnt/cdrom
exclude: /mnt/floppy
```

3. Restart SNMP Agent.

During startup, SNMP Agent loads `/etc/SnmpAgent.d/esafilesys.conf`. During this step, if the `/etc/SnmpAgent.d/esafilesys.conf` file contains a syntax error, SNMP Agent ignores the line containing the error and starts. The syntax error details are output to `/etc/SnmpAgent.d/esafilesys.conf.err`.

2.17 Notes about setup

This section provides notes about setting up SNMP Agent that are common to all OSs. For OS-specific notes, see the relevant subsection.

- General notes about network environment settings
 - You must set the local host name because SNMP Agent uses the IP address for the local host name as the local host's IP address.
 - The host name defined in `trap-dest: in /etc/SnmpAgent.d/snmpd.conf` can be converted to the IP address.

There is no need to specify DNS-related settings.

- Notes about using a firewall for system-to-system communication
 - If there is a firewall between manager system and SNMP Agent, configure the environment so that the SNMP protocol is effective across the firewall. SNMP Agent generally receives SNMP requests via the 161/udp port and sends SNMP traps to the 162/udp port on the manager host.

For details about the settings that are to be added to the firewall, see [B.2 Direction in which data passes through a firewall](#).

If you have changed the port through which SNMP requests are received from SNMP Agent, change the corresponding firewall settings as well. For details about the ports that are used by SNMP Agent at the local host, see [B.1 Port numbers used by SNMP Agent](#).

- SNMP Agent sends SNMP traps to the 162/udp port of the manager host. For notes about the SNMP reception port when overwrite installation is performed on SNMP Agent version 07-50 or earlier, see [2.4 Notes about installation](#).

- Notes about renaming the local host

If you renamed the host after installing SNMP Agent and you want to use a host name with a different `sysName` value, see [3.7.4 Notes about renaming a host](#).

- Notes about changing the manager system's IP address or host name

Check and, if necessary, revise the IP address or host name defined in `trap-dest: in` the configuration file (`/etc/SnmpAgent.d/snmpd.conf`).

- Notes about the `/etc/hosts` file

To use the `naaagt` process on an OS other than HP-UX (IPF), set the IP address for `localhost` in the `/etc/hosts` file as shown in the example below:

Example:

```
127.0.0.1 localhost
```

When the `naaagt` process starts, it searches for the IP address needed for communicating with the local host's native agent, using the OS function and `localhost` as the key value. If this IP address search fails, the `naaagt` process terminates itself because it cannot communicate with the native agent.

- Notes about file systems

For file system information, the following MIB objects are available:

- `fileSystem` group (general 2)
For details, see (2) *fileSystem* group in [4.2.2 Description of Hewlett-Packard enterprise-specific MIB objects](#).
- `fileSystem64` group (hiux 21)
For details, see (20) *fileSystem64* group in [4.3.2 Description of Hitachi enterprise-specific MIB objects](#).

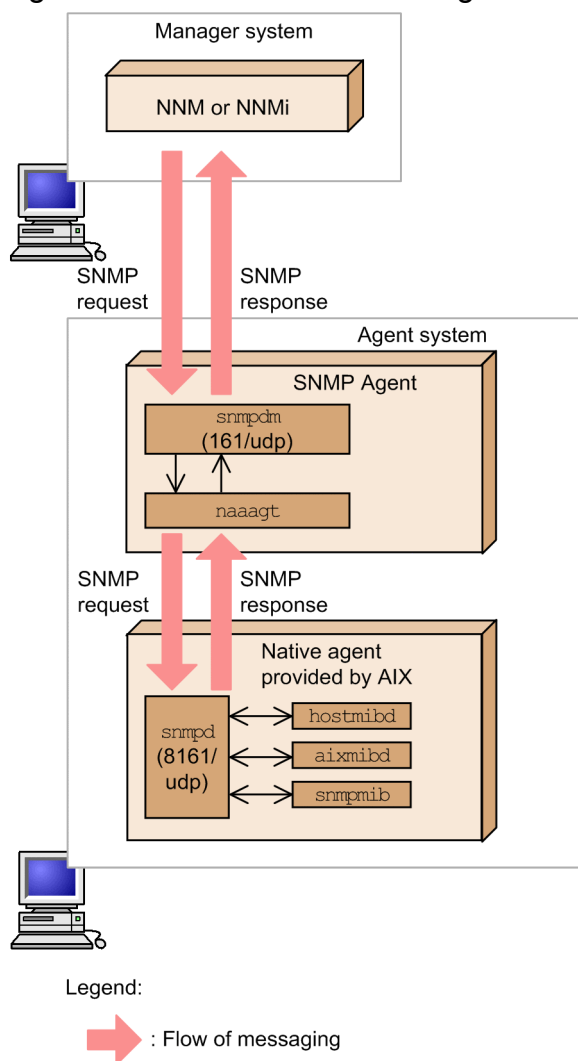
You can set these MIB objects to not send specific file system information as a response. For details, see (20) *fileSystem64 group in 4.3.3 Implementation of Hitachi enterprise-specific MIB objects.*

2.17.1 Notes about setup (for AIX)

Each of the AIX processes `hostmibd`, `aixmibd`, and `snmpmibd` acquires MIB values by issuing `GET` and `GET-NEXT` requests to the process that uses the port number specified in the `SNMP_PORT` value defined in the OS-provided file `/etc/environment` (or `161/udp` if `SNMP_PORT` is not set). If SNMP Agent is not installed, the AIX `snmpd` process typically performs this communication.

The following figure shows the flow of the native agent functions provided by AIX.

Figure 2–9: Flow of the native agent functions provided by AIX



When SNMP Agent is installed, by default it is configured so that the native agent `snmpd` process uses port `8161/udp` as the port for receiving SNMP requests. In addition, `public` is automatically set as the community name that is used when the AIX processes `hostmibd`, `aixmibd`, and `snmpmibd` acquire MIB values by issuing `GET` and `GET-NEXT` requests to the process that uses the port number specified in the `SNMP_PORT` value defined in the OS-provided file `/etc/environment` (or `161/udp` if `SNMP_PORT` is not set).

Therefore, the AIX processes `hostmibd`, `aixmibd`, and `snmpmibd` must be configured to communicate with the AIX `snmpd` process (8161/udp), and the community name must be set according to the native agent (`snmpd`).

These changes are shown in the steps listed below, and they must be made while the user is logged on as a superuser.

1. Stop SNMP Agent and the native agent by using the `snmpstop` command with no arguments.

```
/usr/CM2/ESA/bin/snmpstop
```

2. Open `/usr/CM2/ESA/opt/SnmpNative` with an editor such as `vi`.
3. Change `public` in the following lines to the community name for the native agent `snmpd` process that permits GET and GET-NEXT requests.

```
SNMP_SNMPMIBD_OPTIONS="-c public"
SNMP_HOSTMIBD_OPTIONS="-c public"
SNMP_AIXMIBD_OPTIONS="-c public"
```

4. Add the line below to the OS-provided `/etc/environment`.

If you have changed the port number used by the native agent `snmpd` process, use that port number instead of 8161.

```
SNMP_PORT=8161
```

5. Restart SNMP Agent by using the `snmpstart` command with no arguments.
Restarting SNMP Agent also restarts `hostmibd`, `aixmibd`, and `snmpmibd`.
Execute the following command:

```
/usr/CM2/ESA/bin/snmpstart
```

2.17.2 Notes about setup (for Linux)

This subsection provides Linux-specific notes about setting up an SNMP agent. For setup notes common to all OSs, see [2.17 Notes about setup](#).

- Note about the default value of the SNMP reception port number

The default value of the SNMP reception port number of the SNMP agent is not 161/udp, but 22161/udp. Change the remote port of the manager to match this setting. For details about manager SNMP settings, see the manager documentation.

Reason for changing the default value of the SNMP reception port number

When you change the SNMP reception port number of the Linux native agent from 161/udp to another number, you may not be able to change the port due to SELinux setting. This is why the Linux native agent uses 161/udp, and the SNMP agent uses 22161/udp as an SNMP reception port number.

3

Operating SNMP Agent

This chapter describes startup, termination, and operations of SNMP Agent.

3.1 Starting SNMP Agent

Normally, the master agent and subagents of SNMP Agent start automatically when you start the system. For details about the files that are executed during system startup, see [3.1.5 Files that are executed during system startup](#).

To start SNMP Agent manually, execute the `snmpstart` command as a superuser. For details about the `snmpstart` command, see *snmpstart* in [Chapter 5. Commands and Processes](#).

Important

When the system OS is Linux, execute the `snmpstart` command after starting the native agent.

If the OS being used is Solaris or AIX, and you do not want to start the native agent, execute the `snmpstart` command with the `-n` option.

Each process of SNMP Agent can be run in its default status that existed during installation; however, you can also customize the startup options and environment variable definitions for the processes provided by SNMP Agent according to your environment.

The following subsections describe the settings (customization) for enabling the startup options and environment variable definitions for the processes provided by SNMP Agent.

3.1.1 Customizing the startup options and defining environment variables for the processes

Each agent that constitutes SNMP Agent has a file for defining the startup options and environment variables for its processes (environment variable definition files). The process startup options and environment variable definitions are specified in these files. The process startup options are specified using the environment variables. This means that the environment variables include the process startup options, unless otherwise specified.

The values of environment variables specified in the environment variable definition files take effect when SNMP Agent starts.

The following procedure shows how to configure the environment variable definition files.

Procedure

1. Terminate SNMP Agent.

For details about terminating SNMP Agent, see [3.2 Terminating SNMP Agent](#).

2. Define the environment variables.

In this step, you define the environment variables.

For details about the environment variable definition files provided by SNMP Agent, see [3.1.2 Environment variable definition files provided by SNMP Agent](#).

For details about the environment variables used to specify the startup options of processes, see [3.1.3 Startup options that can be specified in the environment variable definition files](#). For details about the environment variables used for other purposes, see [3.1.4 Environment variables that can be specified for processes](#).

3. Start SNMP Agent.

When you start the system or execute the `snmpstart` command, SNMP Agent starts.

If the OS being used is Solaris or AIX, and you do not want to start the native agent, execute the `snmpstart` command with the `-n` option.

For details about the environment variables for each process of SNMP Agent, see *Processes* in *Chapter 5. Commands and Processes*.

 **Note**

If you specify arguments and start an SNMP Agent process, the specified settings remain in effect until that process is terminated. To always enable the options of processes at startup, you must specify the startup options using the environment variables.

3.1.2 Environment variable definition files provided by SNMP Agent

SNMP Agent provides the environment variable definition files for processes. The following table lists the environment variable definition files for processes and the directory names for the files.

Table 3–1: Environment variable definition files for processes provided by SNMP Agent

Process provided by SNMP Agent	File name	Directory name
snmpdm	SnmpMaster	For HP-UX (IPF) and Linux /opt/CM2/ESA/opt For Solaris /etc/rc.config.d For AIX /usr/CM2/ESA/opt
extsubagt	SnmpExtAgt	
hp_unixagt	SnmpHpunix	
htc_unixagt1	SnmpHtcunix1	
htc_unixagt2	SnmpHtcunix2	
htc_unixagt3	SnmpHtcunix3	
htc_unixagt4	SnmpHtcunix4	
htc_monagt1	SnmpHtcmonagt1	
naaagt	SnmpNaa	
trapdestagt	SnmpTrpDst	

3.1.3 Startup options that can be specified in the environment variable definition files

To specify startup options of processes, use the startup option environment variables. The following table lists the startup option environment variables for processes and the options that can be specified in these environment variables.

Table 3–2: Environment variables for specifying the startup options of processes and the specifiable options

Process provided by SNMP Agent	Startup option environment variable	Specifiable options [#]
snmpdm	SNMP_MASTER_OPTIONS	-aperror, -apwarn, -apverbose, -authfail, -Contact, -hexdump, -ip_proto, -Location, -mask, -sysDescr, -tcplocal, -vbdump
extsubagt	SNMP_EXTAGT_OPTIONS	-e, -E, -aperror, -apwarn, -apconfig, -appacket, -aptrap, -apaccess, -apemanate, -apverbose, -apuser, -retry, -fcmdguard, -pipeguard, -invokeid
hp_unixagt	SNMP_HPUNIX_OPTIONS	-aperror, -apwarn, -apconfig, -appacket, -aptrap, -apaccess, -apemanate, -apverbose, -apuser, -retry
htc_unixagt1	SNMP_HTCUNIX1_OPTIONS	-aperror, -apwarn, -apconfig, -appacket, -aptrap, -apaccess, -apemanate, -apverbose, -apuser, -retry
htc_unixagt2	SNMP_HTCUNIX2_OPTIONS	-aperror, -apwarn, -apconfig, -appacket, -aptrap, -apaccess, -apemanate, -apverbose, -apuser, -retry
htc_unixagt3	SNMP_HTCUNIX3_OPTIONS	-aperror, -apwarn, -apconfig, -appacket, -aptrap, -apaccess, -apemanate, -apverbose, -apuser, -retry
htc_unixagt4	SNMP_HTCUNIX4_OPTIONS	-aperror, -apwarn, -apconfig, -appacket, -aptrap, -apaccess, -apemanate, -apverbose, -apuser, -retry
htc_monagt1	SNMP_HTCMONAGT1_OPTIONS	-i, -t, -s, -d
naaagt	SNMP_NAA_OPTIONS	-aperror, -apwarn, -port, -readcomm, -writecomm, -timeout, -apverbose, -hexdump, -vbdump, -v1, -v2c
trapdestagt	SNMP_TRAPDEST_OPTIONS	-aperror, -apwarn, -apconfig, -appacket, -aptrap, -apaccess, -apemanate, -apverbose, -apuser, -retry

#

For details about the startup options of processes, see *Processes* in *Chapter 5. Commands and Processes*. The `-aperror` and `-apwarn` options that can be specified in the `SNMP_MASTER_OPTIONS` startup option environment variable for the `snmpdm` process correspond to the log mask values (`FACTORY_WARN` and `FACTORY_ERROR`, respectively) of the `-mask` option for this process.

3.1.4 Environment variables that can be specified for processes

You can specify the environment variables listed in the table below in the environment variable definition files for the processes provided by SNMP Agent. For the list of startup option environment variables for processes, see Table 3-2.

Table 3–3: Environment variables that can be specified in the environment variable definition files

Process provided by SNMP Agent	Specifiable environment variable [#]
snmpdm	SR_SNMP_TEST_PORT SNMP_HTC_AUTH_LOG SR_TRAP_TEST_PORT

Process provided by SNMP Agent	Specifiable environment variable#
	SNMP_HTC_INIT_WAIT_TIME SNMP_HTC_SNMPD_LOG_SIZE SNMP_HTC_SNMPD_LOG_CNT
extsubagt	SR_SNMP_TEST_PORT
hp_unixagt	SR_SNMP_TEST_PORT SNMP_HTC_SOLARIS_SWAP_RESERVED (for Solaris) SNMP_HTC_AIX_EXCEPT_FILECACHE (for AIX) SNMP_HTC_LINUX_INACTIVE_MEM (for Linux)
htc_unixagt1	SR_SNMP_TEST_PORT SNMP_HTC_FILE_EXTEND
htc_unixagt2	SR_SNMP_TEST_PORT SNMP_HTC_HPUX_ENABLE_PROCESSOR (for HP-UX(IPF))
htc_unixagt3	SR_SNMP_TEST_PORT
htc_unixagt4	SR_SNMP_TEST_PORT
htc_monagt1	SNMP_HTCMONAGT1_START SNMP_HTC_AIX_CPU_SMT (for AIX)
naaagt	SR_SNMP_TEST_PORT
trapdestagt	SR_SNMP_TEST_PORT

#

For details about the environment variables, see the descriptions under *External influences* in *Detailed process descriptions* of *Chapter 5. Commands and Processes*

3.1.5 Files that are executed during system startup

This section identifies, for each OS, the files that are executed during system startup and how the native agent is started.

For details about the native agent processes required for SNMP Agent, see *Appendix D. List of Prerequisite Patches and Processes (Services) for SNMP Agent*.

- In HP-UX (IPF)

The `/opt/CM2/ESA/bin/snmpstart` command is executed from `/sbin/rc2.d/S560esa`. If the native agent process is not running, the SNMP Agent process and native agent process are started.

If the native agent process is running, the SNMP Agent process is started, and the native agent process is restarted.

- In Solaris

`/etc/rc2.d/S97esa` is executed.

Also, only the SNMP Agent process is started, regardless of whether the native agent (`snmpd` process) is running. Starting of the native agent is controlled on the OS side.

- In AIX

The `/usr/CM2/ESA/bin/snmpstart` command is executed with no arguments from the startup shell script `/usr/CM2/ESA/bin/esa` specified in `/etc/inittab`.

If the native agent process is not running, the SNMP Agent process and native agent process are started.

If the native agent is running, the SNMP Agent process is started, and the native agent process is restarted.

- In Linux (RHEL 6, CentOS 6, and Oracle Linux 6)

According to the run level of the system, the `/opt/CM2/ESA/bin/snmpstart` command (without any arguments) is executed from one of the following files:

- `/etc/rc.d/rc2.d/S55esa`
- `/etc/rc.d/rc3.d/S55esa`
- `/etc/rc.d/rc5.d/S55esa`

Only SNMP agent processes are started, regardless of whether the native agent is running. The starting of the native agent depends on controls on the OS side.

- In Linux (RHEL 7, CentOS 7, Oracle Linux 7, SUSE Linux 12)

The `/opt/CM2/ESA/bin/snmpstart` command is executed with no arguments from the startup shell script `/opt/CM2/ESA/bin/jp1_esa` specified in `/usr/lib/systemd/system/jp1_esa.service`.

Only SNMP agent processes are started, regardless of whether the native agent is running. The starting of the native agent depends on controls on the OS side.

3.2 Terminating SNMP Agent

Normally, the master agent and subagents are automatically terminated when the system is shut down. For details about the files that are executed during system shutdown, see [3.2.2 Files that are executed during system shutdown](#).

To terminate SNMP Agent manually, execute the `snmpstop` command as a superuser. If the OS being used is Solaris or AIX, and you do not want to shut down the native agent, execute the `snmpstop` command with the `-n` option. For details about the `snmpstop` command, see [snmpstop](#) in [Chapter 5. Commands and Processes](#).

The following procedure shows how to terminate individual processes of SNMP Agent manually.

Procedure

1. Use the `ps` command to obtain the process IDs of the master agent and subagents.
2. As a superuser, execute the `kill` command specifying the process IDs obtained in step 1.
3. Use the `ps` command to determine whether the master agent and subagents have been terminated.

3.2.1 Notes about terminating processes individually

The following are notes about terminating processes of SNMP Agent individually:

- When you terminate the master agent, the subagents connected to the master agent might terminate. Also, the subagents provided by the OS and other programs might terminate.
- When you restart the master agent, make sure that you also start the subagents or restart the system.

3.2.2 Files that are executed during system shutdown

The following files are executed during system shutdown:

- In HP-UX (IPF)
The `/opt/CM2/ESA/bin/snmpstop` command is executed from `/sbin/rc1.d/K440esa`.
- In Solaris
The following files are executed:
 - `/etc/rc0.d/K02esa`
 - `/etc/rc1.d/K02esa`Only the SNMP Agent process is stopped.
Shutdown of the native agent is controlled from the OS.
- In AIX
The `/usr/CM2/ESA/bin/snmpstop` command is executed with no arguments from `/etc/rc.shutdown`.
- In Linux (RHEL 6, CentOS 6, and Oracle Linux 6)
The `/opt/CM2/ESA/bin/snmpstop` command is executed from the following files:
 - `/etc/rc.d/rc0.d/K65esa`
 - `/etc/rc.d/rc2.d/K65esa`

- /etc/rc.d/rc3.d/K65esa
- /etc/rc.d/rc5.d/K65esa
- /etc/rc.d/rc6.d/K65esa
- In Linux (RHEL 7, CentOS 7, Oracle Linux 7, SUSE Linux 12)
/opt/CM2/ESA/bin/snmpstop is executed from the shutdown shell script /opt/CM2/ESA/bin/jp1_esa specified in /usr/lib/systemd/system/jp1_esa.service.

3.3 Starting and terminating the native agent

The following describes the use of the `snmpstart` and `snmpstop` commands, provided by SNMP Agent, to start and terminate the native agent.

In Solaris and AIX

The `snmpstart` command with no arguments also starts the native agent. The `snmpstop` command with no arguments also terminates the native agent.

If you run the `snmpstart` command with the `-n` option, the native agent is not started. If you run the `snmpstop` command with the `-n` option, the native agent is not terminated.

In Linux

The `snmpstart` command and the `snmpstop` command do not start and stop the native agent. The starting and stopping of the native agent depend on controls on the OS side.

3.4 Changing the SNMP reception port on SNMP Agent

You need to change the SNMP reception port on SNMP Agent in the following cases:

- A native agent or other SNMP Agent product and SNMP Agent are using the same SNMP reception port at the same time.
- A firewall environment requires a different SNMP reception port on SNMP Agent.

The SNMP reception port on SNMP Agent is determined based on the following priority:

1. The port number specified in the `SR_SNMP_TEST_PORT` environment variable
The `SR_SNMP_TEST_PORT` environment value is set to 161 in Solaris or AIX, and 22161 in Linux, respectively by default.
2. The port number specified for the `snmp` service name in the `/etc/services` file

The following subsections describe how to change the SNMP reception port on SNMP Agent and native agent.

3.4.1 Changing the SNMP reception port on SNMP Agent

This subsection describes how to change the SNMP reception port on SNMP Agent.

There are two ways of changing the SNMP reception port on SNMP Agent. One is by using `SR_SNMP_TEST_PORT` and the other is by using the `/etc/services` file. Changing values in the `/etc/services` file might have adverse effects on other SNMP Agents. Therefore, Hitachi recommends that you use `SR_SNMP_TEST_PORT`.

The following procedure shows how to change the SNMP reception port on SNMP Agent using `SR_SNMP_TEST_PORT`.

Procedure

1. Add the following two lines to the file that defines environment variables:

```
SR_SNMP_TEST_PORT=new-port-number
export SR_SNMP_TEST_PORT
```

The file used to define the `SR_SNMP_TEST_PORT` environment variable depends on the OS in use, as the following shows:

In HP-UX (IPF) and Linux: `/opt/CM2/ESA/opt/SnmpMaster`

In Solaris: `/etc/rc.config.d/SnmpMaster`

In AIX: `/usr/CM2/ESA/opt/SnmpMaster`

2. Change the UDP port number (applicable to Solaris, AIX and Linux).

If you change the native agent's SNMP reception port from 8161/udp or 161/udp to another port number, you must also change the UDP port that the `naaagt` process uses to connect to the native agent.

Add the following two lines to the file that defines environment variables:

```
SNMP_NAA_OPTIONS="-port native-agent's-SNMP-reception-port-number"
export SNMP_NAA_OPTIONS
```

The file used to define the `SNMP_NAA_OPTIONS` environment variable depends on the OS in use, as the following shows:

In Solaris: `/etc/rc.config.d/SnmpNaa`

In AIX: `/usr/CM2/ESA/opt/SnmpNaa`

In Linux: `/opt/CM2/ESA/opt/SnmpNaa`

3. Restart SNMP Agent.

Execute the following command as a superuser:

If the OS being used is Solaris or AIX, and you have changed the native agent's SNMP reception port, execute the `snmpstart` command with no arguments.

```
/opt/CM2/ESA/bin/snmpstart
```

3.4.2 Changing the SNMP reception port on the native agent `snmpd` (for AIX)

This subsection describes how to change the SNMP reception port on the AIX-provided native agent `snmpd`.

If you have installed SNMP Agent, the SNMP reception port for the native agent `snmpd` is set to 8161/udp. To change it to another port number, follow the procedure described below. To perform this procedure, first terminate SNMP Agent and then execute the procedure as a superuser.

Procedure

1. Change the `SNMP_NATIVE_OPTIONS` environment variable.

Open the `SnmpNative` environment variable definition file using an editor such as `vi`, and then change 8161 on the following line to another port number:

```
SNMP_NATIVE_OPTIONS="-p 8161"
```

2. Execute the `snmpstart` command with no arguments.

SNMP Agent restarts. As a result, the native agent `snmpd` also restarts.

Note that if the native agent `snmpd` is an SNMP v1 agent, it uses 161/udp as the SNMP reception port, and SNMP Agent's SNMP reception port must be set to a port other than 161/udp.

The environment variable `SNMP_NATIVE_OPTIONS` is used when the native agent `snmpd` is an SNMP v3 agent.

3.5 Changing the maximum number of connected subagents

The maximum number of connected subagents indicates the maximum number of subagents that can be connected to the master agent. The default value is 22. This setting is defined in the `/etc/srconf/agt/snmpd.conf` file. If the master agent receives a connection request that exceeds the maximum number of subagents that can be connected, it will output the following message to the `/var/adm/snmpd.log` file. For details about the `snmpd.conf` file, see *Configuration file (snmpd.conf)* in *Chapter 6. Definition Files*.

```
AllocSubagent: runtime_MAX_SUBAGENTS exceeded
```

If this message is output, change the maximum number of connected subagents. To do this, log into the system as a superuser and perform the following procedure.

Procedure

1. Open `/etc/srconf/agt/snmpd.conf`.

2. Find the following line:

```
MAX_SUBAGENTS 22
```

This value is the maximum number of connected subagents.

3. After the `MAX_SUBAGENTS` label, enter a desired number of subagents that can be connected.

4. Find the following line:

```
MAX_THREADS 22
```

This value is the maximum number of threads that can be generated by the master agent at the same time.

5. After the `MAX_THREADS` label, enter the same value as that specified for the `MAX_SUBAGENTS` label.

6. Execute the `snmpstart` command.

SNMP Agent restarts.

Important

The minimum values of `MAX_SUBAGENTS` and `MAX_THREADS` are both 22. The maximum value depends on the maximum number of threads that can be generated per process by the OS.

3.6 Backing up and restoring

This section describes backup and restoration.

Perform backup and restoration using any method you wish, because no backup or restoration command is provided. For notes about performing full-backup and full-restoration using backup software, see [3.6.2 Notes about full-backup and full-restoration](#).

3.6.1 Backing up and restoring the configuration files

The following are the configuration files recommended to be backed up for purposes such as error recovery:

- `/etc/SnmpAgent.d/snmpd.conf`
- `/etc/srconf/agt/snmpd.cnf`
- `/etc/srconf/agt/naa.cnf` (this file is not needed in HP-UX (IPF))
- `/etc/SnmpAgent.d/esafilesys.conf`
- `/etc/SnmpAgent.d/esafstab.conf` (for Linux)
- `/etc/SnmpAgent.d/esadisk.conf` (for Linux)
- `/etc/SnmpAgent.d/esalocale.conf`

Additionally, back up the following files, if necessary:

When a user-defined MIB function is used:

`/etc/SnmpAgent.d/snmpd.extend`

Files under the `/opt/CM2/ESA/ext` directory (for systems other than AIX)

Files under the `/usr/CM2/ESA/ext` directory (for AIX)

When SNMP Agent's startup options have been changed:

Files beginning with `/etc/rc.config.d/Snmp` (for Solaris)

Files under the `/opt/CM2/ESA/opt` directory (for HP-UX (IPF) and Linux)

Files under the `/usr/CM2/ESA/opt` directory (for AIX)

You can perform backup processing when SNMP Agent is running and when it is terminated. However, when you perform restoration processing, make sure that SNMP Agent is terminated.

3.6.2 Notes about full-backup and full-restoration

This subsection provides notes about performing full-backup and full-restoration using backup software. Be sure to read these notes before you perform full-backup or full-restoration.

- Perform full-backup and full-restoration while SNMP Agent is terminated.
If you execute full-backup while SNMP Agent is running, the files that are generated during operations might not be backed up.
- Full-backup also backs up the information that has been acquired by SNMP Agent during operations.

If the information that was obtainable by SNMP Agent during full-backup does not match the information that is obtainable by SNMP Agent after restoration, an inconsistency occurs, and SNMP Agent might not function correctly after the full-restoration.

If you perform full-restoration on a machine that is different from the machine on which full-backup was performed, an inconsistency is more likely to occur.

- If SNMP Agent does not function correctly after the full-restoration, terminate it and then perform the following procedure:
 1. Back up SNMP Agent configuration files.
 2. Uninstall SNMP Agent.
 3. Perform a new installation of SNMP Agent.
 4. Restore the backup configuration files.

3.7 Notes about operations

This section provides notes about SNMP Agent operations common to all OSs. For OS-specific notes, see the relevant subsections.

- Notes about a file whose size continually increases with no capacity limits

The following table shows a log file whose size continually increases after operations start.

Table 3–4: Notes about a file whose size continually increases with no capacity limits

Path	Process that outputs logs	Note
/tmp/esa.log	Shell during installation and uninstallation	You can delete this file, except when this product is being installed or uninstalled.

There are no capacity limits for this file.

- Notes about log files

The following table provides notes about the log files that are output after operations start.

Table 3–5: Notes about the log files that are output after operations start

Path#1	Process that outputs logs	Note
/var/adm/snmpd.logn	snmpdm	By default, 10 log files (<i>n</i> : 1 to 10) with a file size of 10 megabytes each are acquired in wraparound mode. If this default value introduces an operating problem, change the log file size, number of log files, and output destination path.#2

#1: For AIX, the files are located under /usr/adm.

#2: For details about changing the log file size, number of files, and output destination path, see [7.3 Collecting logs](#).

- Notes about the files used by SNMP Agent

SNMP Agent uses the files in and under the /tmp/.AgentSockets directory. Do not delete any of these files while SNMP Agent is running.

You can delete these files while SNMP Agent is stopped. SNMP Agent creates this directory when it starts.

- Notes about changing the IP address of the node

If you have changed the IP address of the node while SNMP Agent is running, restart SNMP Agent.

- Notes about backing up the environment variable definition files

- When you back up an environment variable definition file, make sure that the name of the backup file does not begin with `Snmp`. The following shows an example of a name for a backup file:

Example: Backup file of /opt/CM2/ESA/opt/SnmpMaster
/opt/CM2/ESA/opt/Bak.SnmpMaster

- If your OS is Solaris, do not create backup environment variable definition files under /etc/rc.config.d.

- Notes about the native agent adapter function

In Solaris, AIX and Linux systems, use the same community name for SNMP Agent's native agent adapter and the OS-provided native agent.

- Notes about SNMP Agent execution permissions

Only a user with `root` permissions can access SNMP Agent files. Do not change the file access permissions.

- Language environment for SNMP Agent

SNMP Agent outputs only English messages regardless of the language environment used to run SNMP Agent. You can also change the system's language environment after you have installed SNMP Agent without any problem.

- Notes about using JP1/SSO to collect resources

- Specify a timeout value and retry count for each platform.

The table below lists the timeout values recommended for each OS. The timeout value depends on the system load and the network environment. Specify a timeout value appropriate to your environment.

Table 3–6: Recommended timeout value

OS	Recommended timeout value
Solaris	6.0 seconds or more
AIX	3.0 seconds or more
Linux	3.0 seconds or more
HP-UX (IPF)	0.8 seconds or more

The SNMP requests use UDP, but you must specify a retry count because UDP does not have a retry function.

- If you use NNM to acquire enterprise-specific MIBs, refer to the notes about MIBs. For details, see the notes about groups in *4.2.2 Description of Hewlett-Packard enterprise-specific MIB objects* and *4.3.2 Description of Hitachi enterprise-specific MIB objects*.

- Notes about changing the system time

To advance the system time, no special procedure is needed.

To set back the system time, follow the procedure below:

1. Use the `snmpstop` command to terminate SNMP Agent.

If the OS being used is Solaris or AIX, and you do not want to shut down the native agent, execute the `snmpstop` command with the `-n` option.

2. Change the system time.

3. Use the `snmpstart` command to restart SNMP Agent.

If the OS being used is Solaris or AIX, and you do not want to start up the native agent, execute the `snmpstart` command with the `-n` option.

- Notes about sending `coldStart` traps when the OS starts up

By default, the `snmpd` process of the master agent sends a `coldStart` trap 15 seconds after it starts up.

No response is sent to the manager's request during this time because the process sends a `coldStart` trap without checking whether startup processing is completed for other subagents. Normally 15 seconds are sufficient for subagents to complete their startup processing; however, some subagents might require more time depending on the environment. If this is the case, adjust the timing of `coldStart` transmission by specifying the appropriate time (in seconds) before the `coldStart` trap can be sent in the `SNMP_HTC_INIT_WAIT_TIME` environment variable in the `SnmpMaster` file.

The following shows a specification example for the `SNMP_HTC_INIT_WAIT_TIME` environment variable.

Example:

```
SNMP_HTC_INIT_WAIT_TIME=15
export SNMP_HTC_INIT_WAIT_TIME
```

- Notes about reloading the configuration file (`/etc/SnmpAgent.d/snmpd.conf`) during `SIGNUP` reception

While SNMP Agent is running, the configuration file (`/etc/SnmpAgent.d/snmpd.conf`) is not re-loaded during `SIGNUP` reception.

3.7.1 Notes about operations (for Solaris)

This subsection provides notes about SNMP Agent operations when the OS in use is Solaris. For notes common to all OSs, see [3.7 Notes about operations](#).

- Notes about operations in Solaris using SMF

If you execute the `snmpstop` command with no arguments to terminate SNMP Agent, the native agent processes (`snmpd`) that are managed by SMF are also terminated. If you do not want to shut down the native agent, execute the `snmpstop` command with the `-n` option. If you execute the `snmpstart` command with no arguments to start SNMP Agent, the native agent processes (`snmpd`) that are managed by SMF are also started. Therefore, there is no need to manually start and stop the processes of the native agent (`snmpd`). If you do not want to start up the native agent, execute the `snmpstart` command with the `-n` option.

- Notes about swap space size

See [2.14 Notes about swap space size](#).

3.7.2 Notes about operations (for AIX)

This subsection provides notes about SNMP Agent operation when the OS in use is AIX. For notes common to all OSs, see [3.7 Notes about operations](#).

- Notes about acquiring the MIB `page` group in AIX

Depending on the system configuration, an attempt to acquire a MIB belonging to the Hitachi enterprise-specific MIB `page` group might frequently result in an error. In such a case, measure the runtime of the `page .exe` command provided by SNMP Agent.

If execution of the `page .exe` command takes 10 seconds or more, the system cannot acquire the correct `page` group MIB. If you continue operations in this status, the operating system's command process, which runs in an attempt to acquire page information, will continue to run illegally. This could increase the system workload.

To avoid this problem, edit the `page .exe` command as shown below. Once this change is made, the command will not execute even when a MIB belonging to the `page` group has been acquired. Pseudo information indicating page 0 is always returned as the MIB value for the `page` group.

Before change:

```
lsps -a > $OUTFILE
```

After change:

```
echo  
exit 0
```

- Notes about the native agent function

- The native agent adapter provided by SNMP Agent (`naaagt` process) specifies the community name `public` in the SNMP `GET` and `GET-NEXT` requests of SNMPv1 to acquire MIBs from the AIX-provided native agent (`snmpd` process). This is not a problem because the default settings of the `snmpd` process in SNMPv1 permit SNMP `GET` and `GET-NEXT` requests for the community name `public`. However, if you change the community name for the `snmpd` process, make sure that you permit SNMP `GET` and `GET-NEXT` requests in SNMPv1. If you have changed the community name for the `snmpd` process to a value other than `public`, also change the `naaagt` process settings. For details about how to change the `naaagt` process settings, see [Configuration file \(`naa.cnf`\)](#) in [Chapter 6. Definition Files](#).

- For the `snmpd` process, you can choose to use `snmpdv1` or `snmpdv3` via the `snmpv3_ssw` command provided in AIX. With the AIX default settings, the process uses the `snmpdv3` agent.

If you start SNMP Agent and the `snmpdv1` or `snmpdv3` agent at the same time, conflict occurs on the SNMP reception port (161/udp). You must change the SNMP reception port for the `snmpdv1` or `snmpdv3` agent. SNMP Agent changes the port based on the `snmpdv3` agent that is used by default in AIX. For details about how to change the SNMP reception port, see the AIX documentation.

- Notes about avoiding process termination due to a shortage of OS memory

In the event of a shortage of OS memory in AIX, `SIGKILL` is issued and the process might terminate. To avoid this, set `PSALLOC=early` in the environment variable of the user that starts SNMP Agent, and then start SNMP Agent. If you set `early` in the `PSALLOC` environment variable, also set the `NODISCLAIM=true` environment variable, and then restart SNMP Agent.

Set the `PSALLOC` and `NODISCLAIM` environment variables in the `SnmprMaster` file.

The following shows an example.

Example:

```
SNMP_MASTER_OPTIONS="-tcplocal"           # Master Agent options
export SNMP_MASTER_OPTIONS
PSALLOC=early
export PSALLOC
NODISCLAIM=true
export NODISCLAIM
```

- For notes about the amount of free space in physical memory, see [2.13 Notes about the amount of free space in physical memory](#).
- For notes about CPU usage rate information, see [2.15 Notes about CPU information](#).

3.7.3 Notes about operations (for Linux)

This subsection provides notes about SNMP Agent operation when the OS in use is Linux. For notes common to all OSs, see [3.7 Notes about operations](#).

- Notes about SNMP Agent starts

In RHEL 7, CentOS 7, Oracle Linux 7, SUSE Linux 12, the message of `snmpstart` command is output by `/var/opt/CM2/ESA/log/esastart.log`.

3.7.4 Notes about renaming a host

SNMP Agent considers the host name in effect at the time of SNMP Agent installation to be the `sysName` value.

SNMP Agent also considers the value that is set based on the `SetRequest` from the manager to be the `sysName` value, and saves the `sysName` value in `/etc/srconf/agt/snmpd.cnf`.

Even if the host name is changed after SNMP Agent is installed, SNMP Agent does not change the `sysName` value to the new host name. This is because the `sysName` value being used might be the value that was set based on the `SetRequest` from the manager. To set the `sysName` value to the new host name, use one of the following procedures:

- Change the `sysName` value in `/etc/srconf/agt/snmpd.cnf` to the new host name, then restart SNMP Agent using the `/opt/CM2/ESA/bin/snmpstart` command. If the OS being used is Solaris or AIX, and you do not want to restart the native agent, execute the `snmpstart` command with the `-n` option.
- Delete the `sysName host-name` line in `/etc/srconf/agt/snmpd.cnf`, and then restart SNMP Agent.
- Use a `SetRequest` to set the `sysName` value to the new host name.

For details about `sysName`, see [4.1.2\(1\) System group](#) and [Configuration file \(`snmpd.cnf`\)](#).

For notes about setup, see [2.17 Notes about setup](#).

4

MIB Objects

This chapter contains tables that list the MIB objects supported by SNMP Agent and indicate how the objects are implemented.

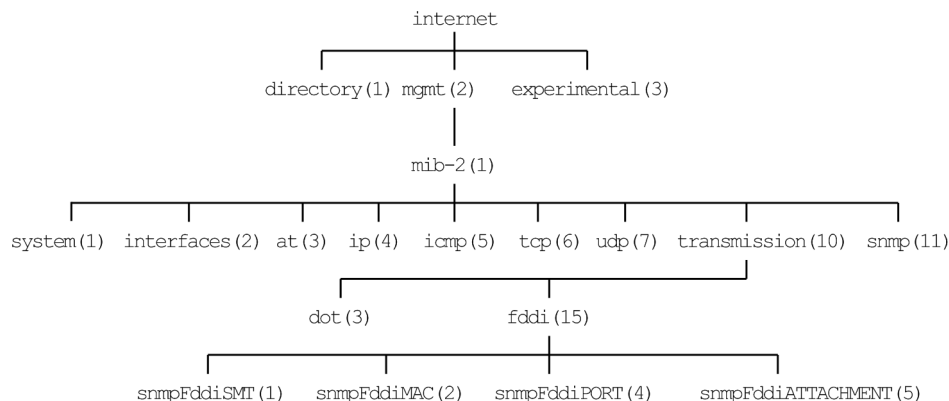
4.1 Standard MIB objects

This section contains tables that list the standard MIB objects supported by SNMP Agent and explain how these objects are implemented.

4.1.1 Organization of standard MIB objects

The next figure indicates the organization of standard MIB objects.

Figure 4–1: Organization of standard MIB objects



The following table indicates, for each group, the tables that describe the standard MIB objects and their implementation statuses.

Table 4–1: Referenced locations for the standard MIB objects

Standard MIB object group		Referencing destination	
		MIB object description	MIB object implementation status
internet.mgmt.mib-2	system	Table 4-2	Table 4-10
	interfaces	Table 4-3	--
	at	Table 4-4	--
	ip	Table 4-5	--
	icmp	Table 4-6	--
	tcp	Table 4-7	--
	udp	Table 4-8	--
	snmp	Table 4-9	--

Legend:

--: Not applicable

4.1.2 Description of standard MIB objects

This subsection describes the standard MIB objects in each group. The tables in this subsection use the following legend:

Legend:

--: Not applicable

Details about the contents of the objects can be found in RFC 1213, RFC 1285, and RFC 1398.

(1) System group

The following table describes the standard MIB objects in the `System` group.

Table 4–2: System group (`internet.mgmt.mib-2.system`) (1.2.1.1)

ID	Object name	Contents	Units
1	<code>sysDescr</code>	System-related description	--
2	<code>sysObjectID</code>	Value of the object ID assigned to the system	--
3	<code>sysUpTime</code>	The time elapsed since the system was last started	Hundredths of a second
4	<code>sysContact</code>	System administrator contact information	--
5	<code>sysName#</code>	System's host name	--
6	<code>sysLocation</code>	System's installation location	--
7	<code>sysServices</code>	Services provided by the system (OSI reference layer)	--

Legend:

--: Not applicable

#

For details about how to use a new host name as the `sysName` value after you have installed SNMP Agent and renamed the host, see [3.7.4 Notes about renaming a host](#).

(2) Interfaces group

The following table describes the standard MIB objects in the `Interfaces` group.

Table 4–3: Interfaces group (`internet.mgmt.mib-2.interfaces`) (1.2.1.2)

ID	Object name	Contents	Units
1	<code>ifNumber</code>	The number of network interfaces.	--
2	<code>ifTable</code>	An interface entity information table.	--
2.1	<code>ifEntry</code>	An interface entry containing objects at the subnetwork layer and below for a particular interface.	--
2.1.1	<code>ifIndex</code>	A unique value for each interface.	--
2.1.2	<code>ifDescr</code>	A textual string containing information about the interface.	--
2.1.3	<code>ifType</code>	The type of interface. other (1), regular1822 (2), hdh1822 (3), ddn-x25 (4), rfc877x25 (5), ethernet-csmacd (6), iso88023-csmacd (7), iso88024-tokenBus (8), iso88025-tokenRing (9), iso88026-man (10), starLan (11), proteon-,10Mbit (12), proteon-80Mbit (13), hyperchannel (14), fddi (15), lapb (16), sdlc (17), dsl (18), el (19), basicISDN (20), primaryISDN (21), propPintToPointSerial (22), ppp (23), softwareLoopback (24), eon (25), ethernet-3Mbit (26), nsip (27), slip (28), ultra (29), ds3 (30), sip (31), frame-relay (32)	--

ID	Object name	Contents	Units
2.1.4	ifMtu	The size of the largest datagram which can be sent/received on the interface, specified in octets.	Octets
2.1.5	ifSpeed	An estimate of the interface's current bandwidth in bits per second.	Bits/ second
2.1.6	ifPhysAddress	The interface's address at the protocol layer immediately 'below' the network layer in the protocol stack.	--
2.1.7	ifAdminStatus	The desired state of the interface. up (1), down (2), testing (3)	--
2.1.8	ifOperStatus	The current operational state of the interface. up (1), down (2), testing (3)	--
2.1.9	ifLastChange	The value of sysUpTime at the time the interface entered its current operational state.	Hundredths of a second
2.1.10	ifInOctets	The total number of octets received on the interface, including framing characters.	Octets
2.1.11	ifInUcastPkts	The number of subnetwork-unicast packets delivered to a higher-layer protocol.	Packets
2.1.12	ifInNUcastPkts	The number of subnetwork-broadcast packets or multicast packets delivered to a higher-layer protocol.	Packets
2.1.13	ifInDiscards	The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol.	Packets
2.1.14	ifInErrors	The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.	Packets
2.1.15	ifInUnknownProtos	The number of packets received via the interface which were discarded because of an unknown or unsupported protocol.	Packets
2.1.16	ifOutOctets	The total number of octets transmitted out of the interface, including framing characters.	Octets
2.1.17	ifOutUcastPkts	The total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or not sent.	Packets
2.1.18	ifOutNUcastPkts	The total number of packets that higher-level protocols requested be transmitted to a non-unicast (that is, a subnetwork-broadcast or subnetwork-multicast) address, including those that were discarded or not sent.	Packets
2.1.19	ifOutDiscards	The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted.	Packets
2.1.20	ifOutErrors	The number of outbound packets that could not be transmitted because of errors.	Packets
2.1.21	ifOutQLen	The length of the output packet queue (in packets).	Packets
2.1.22	ifSpecific	A reference to MIB definitions specific to the particular media being used to realize the interface.	--

(3) AddressTranslation group

The following table describes the standard MIB objects in the AddressTranslation group.

Table 4–4: AddressTranslation group (internet.mgmt.mib-2.at) (1.2.1.3)

ID	Object name	Contents	Units
1	atTable	The Address Translation tables contain the NetworkAddress to physical address equivalence.	--
1.1	atEntry	Each entry contains one NetworkAddress to physical address equivalence.	--
1.1.1	atIfIndex	The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.	--
1.1.2	atPhysAddress	The media-dependent physical address.	--
1.1.3	atNetAddress	The NetworkAddress corresponding to the media-dependent physical address.	--

(4) IP group

The following table describes the standard MIB objects in the IP group.

Table 4–5: IP group (internet.mgmt.mib-2.ip) (1.2.1.4)

ID	Object name	Contents	Units
1	ipForwarding	The indication of whether this entity is acting as an IP gateway in respect to the forwarding of datagrams received by, but not addressed to, this entity. forwarding (1), not-forwarding (2)	--
2	ipDefaultTTL	The default value inserted into the Time-To-Live field of the IP header of datagrams originated at this entity.	--
3	ipInReceives	The total number of input datagrams received from interfaces, including those received in error.	Datagrams
4	ipInHdrErrors	The number of input datagrams discarded due to errors in their IP headers.	Datagrams
5	ipInAddrErrors	The number of input datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity.	Datagrams
6	ipForwDatagrams	The number of input datagrams for which this entity was not their final IP destination.	Datagrams
7	ipInUnknownProtos	The number of locally-addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.	Datagrams
8	ipInDiscards	The number of input IP datagrams for which no problems were encountered to prevent their continued processing, but which were discarded.	Datagrams
9	ipInDelivers	The total number of input datagrams successfully delivered to IP user-protocols (including ICMP).	Datagrams
10	ipOutRequests	The total number of IP datagrams which local IP user-protocols (including ICMP) supplied to IP in requests for transmission.	Datagrams
11	ipOutDiscards	The number of output IP datagrams for which no problem was encountered to prevent their transmission to their destination, but which were discarded.	Datagrams
12	ipOutNoRoutes	The number of IP datagrams discarded because no route could be found to transmit them to their destination.	Datagrams
13	ipReasmTimeout	The maximum number of seconds which received fragments are held while they are awaiting reassembly at this entity.	Seconds

ID	Object name	Contents	Units
14	ipReasmReqds	The number of IP fragments received which needed to be reassembled at this entity.	--
15	ipReasmOKs	The number of IP datagrams successfully reassembled.	--
16	ipReasmFails	The number of failures detected by the IP reassembly algorithm.	--
17	ipFragOKs	The number of IP datagrams that have been successfully fragmented at this entity.	Datagrams
18	ipFragFails	The number of IP datagrams that have been discarded because they needed to be fragmented at this entity.	Datagrams
19	ipFragCreates	The number of IP datagram fragments that have been generated as a result of fragmentation at this entity.	--
20	ipAddrTable	The table of addressing information relevant to this entity's IP addresses.	--
20.1	ipAddrEntry	The addressing information for one of this entity's IP addresses.	--
20.1.1	ipAdEntAddr	The IP address to which this entry's addressing information pertains.	--
20.1.2	ipAdEntIfIndex	The index value which uniquely identifies the interface to which this entry is applicable.	--
20.1.3	ipAdEntNetMask	The subnet mask associated with the IP address of this entry.	--
20.1.4	ipAdEntBcastAddr	The value of the least-significant bit in the IP broadcast address used for sending datagrams on the (logical) interface associated with the IP address of this entry.	--
20.1.5	ipAdEntReasmMaxSize	The maximum size of the IP datagram into which the received IP datagram fragments can be reassembled.	Datagrams
21	ipRouteTable	This entity's IP Routing table.	--
21.1	ipRouteEntry	A route to a particular destination.	--
21.1.1	ipRouteDest	The destination IP address of this route.	--
21.1.2	ipRouteIfIndex	The index value which uniquely identifies the local interface through which the next hop of this route will be reached.	--
21.1.3	ipRouteMetric1	The primary routing metric for this route.	--
21.1.4	ipRouteMetric2	An alternate routing metric for this route.	--
21.1.5	ipRouteMetric3	An alternate routing metric for this route.	--
21.1.6	ipRouteMetric4	An alternate routing metric for this route.	--
21.1.7	ipRouteNextHop	The IP address of the next hop of this route.	--
21.1.8	ipRouteType	The type of route. other (1), invalid (2), direct (3), indirect (4)	--
21.1.9	ipRouteProto	The routing mechanism via which this route was learned. other (1), local (2), netmgmt (3), icmp (4), egp (5), ggp (6), hello (7), rip (8), is-is (9), es-is (10), cisco-Igrp (11), bbnSpfIgp (12), ospf (13), bgp (14)	--
21.1.10	ipRouteAge	The number of seconds since this route was last updated or otherwise determined to be correct.	Seconds
21.1.11	ipRouteMask	The mask value to be logical-AND-ed with the destination address.	--
21.1.12	ipRouteMetric5	An alternate routing metric for this route.	--

ID	Object name	Contents	Units
21.1.13	ipRouteInfo	A reference to MIB definitions specific to the particular routing protocol.	--
22	ipNetToMediaTable	The IP Address Translation table used for mapping from IP addresses to physical addresses.	--
22.1	ipNetToMediaEntry	Each entry contains one IpAddress to physical address equivalence.	--
22.1.1	ipNetToMediaIfIndex	The interface on which this entry's equivalence is effective.	--
22.1.2	ipNetToMediaPhysAddresses	The media-dependent physical address.	--
22.1.3	ipNetToMediaNetAddress	The IpAddress corresponding to the media-dependent physical address.	--
22.1.4	ipNetToMediaType	The type of mapping. other (1), invalid (2), dynamic (3), static (4)	--
23	ipRoutingDiscards	The number of routing entries which were chosen to be discarded even though they are valid.	Entries

(5) ICMP group

The following table describes the standard MIB objects in the ICMP group.

Table 4–6: ICMP group (internet.mgmt.mib-2.icmp) (1.2.1.5)

ID	Object name	Contents	Units
1	icmpInMsgs	The total number of ICMP messages which the entity received.	Messages
2	icmpInErrors	The number of ICMP messages which the entity received.	Messages
3	icmpInDestUnreachs	The number of ICMP Destination Unreachable messages received.	Messages
4	icmpInTimeExcds	The number of ICMP Time Exceeded messages received.	Messages
5	icmpInParmProbs	The number of ICMP Parameter Problem messages received.	Messages
6	icmpInSrcQuenchs	The number of ICMP Source Quench messages received.	Messages
7	icmpInRedirects	The number of ICMP Redirect messages received.	Messages
8	icmpInEchos	The number of ICMP Echo (request) messages received.	Messages
9	icmpInEchoReps	The number of ICMP Echo Reply messages received.	Messages
10	icmpInTimestamps	The number of ICMP Timestamp (request) messages received.	Messages
11	icmpInTimestampReps	The number of ICMP Timestamp Reply messages received.	Messages
12	icmpInAddrMasks	The number of ICMP Address Mask Request messages received.	Messages
13	icmpInAddrMaskReps	The number of ICMP Address Mask Reply messages received.	Messages
14	icmpOutMsgs	The total number of ICMP messages which this entity attempted to send.	Messages
15	icmpOutErrors	The number of ICMP messages which this entity did not send due to problems discovered within ICMP such as a lack of buffers.	Messages
16	icmpOutDestUnreachs	The number of ICMP Destination Unreachable messages sent.	Messages
17	icmpOutTimeExcds	The number of ICMP Time Exceeded messages sent.	Messages
18	icmpOutParmProbs	The number of ICMP Parameter Problem messages sent.	Messages

ID	Object name	Contents	Units
19	icmpOutSrcQuenchs	The number of ICMP Source Quench messages sent.	Messages
20	icmpOutRedirects	The number of ICMP Redirect messages sent.	Messages
21	icmpOutEchos	The number of ICMP Echo (request) messages sent.	Messages
22	icmpOutEchoReps	The number of ICMP Echo Reply messages sent.	Messages
23	icmpOutTimestamps	The number of ICMP Timestamp (request) messages sent.	Messages
24	icmpOutTimestampReps	The number of ICMP Timestamp Reply messages sent.	Messages
25	icmpOutAddrMasks	The number of ICMP Address Mask Request messages sent.	Messages
26	icmpOutAddrMaskReps	The number of ICMP Address Mask Reply messages sent.	Messages

(6) TCP group

The following table describes the standard MIB objects in the TCP group.

Table 4–7: TCP group (internet.mgmt.mib-2.tcp) (1.2.1.6)

ID	Object name	Contents	Units
1	tcpRtoAlgorithm	The algorithm used to determine the timeout value used for retransmitting unacknowledged octets. other (1), constant (2), rsre (3), vanj (4)	--
2	tcpRtoMin	The minimum value permitted by a TCP implementation for the retransmission timeout.	Milliseconds
3	tcpRtoMax	The maximum value permitted by a TCP implementation for the retransmission timeout.	Milliseconds
4	tcpMaxConn	The limit on the total number of TCP connections the entity can support.	--
5	tcpActiveOpens	The number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.	--
6	tcpPassiveOpens	The number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.	--
7	tcpAttemptFails	The number of times TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state.	--
8	tcpEstabResets	The number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.	--
9	tcpCurrEstab	The number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT.	--
10	tcpInSegs	The total number of segments received.	Segments
11	tcpOutSegs	The total number of segments sent.	Segments
12	tcpRetransSegs	The total number of segments retransmitted.	Segments
13	tcpConnTable	A table containing TCP connection-specific information.	--
13.1	tcpConnEntry	Information about a particular current TCP connection.	--

ID	Object name	Contents	Units
		tcpConnLocalAddress tcpConnLocalPort tcpConnRemAddress tcpConnRemPort	
13.1.1	tcpConnState	The state of this TCP connection. closed (1), listen (2), synSent (3), synReceived (4), established (5), finWait1 (6), finWait2 (7), closeWait (8), lastAck (9), closing (10), timeWait (11), deleteTCB (12)	--
13.1.2	tcpConnLocalAddress	The local IP address for this TCP connection.	--
13.1.3	tcpConnLocalPort	The local port number for this TCP connection.	--
13.1.4	tcpConnRemAddress	The remote IP address for this TCP connection.	--
13.1.5	tcpConnRemPort	The remote port number for this TCP connection.	--
14	tcpInErrs	The total number of segments received in error.	Segments
15	tcpOutRsts	The number of TCP segments sent containing the RST flag.	Segments

(7) UDP group

The following table describes the standard MIB objects in the UDP group.

Table 4–8: UDP group (internet.mgmt.mib-2.udp) (1.2.1.7)

ID	Object name	Contents	Units
1	udpInDatagrams	The total number of UDP datagrams delivered to UDP users.	Datagrams
2	udpNoPorts	The total number of received UDP datagrams for which there was no application at the destination port.	Datagrams
3	udpInErrors	The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.	Datagrams
4	udpOutDatagrams	The total number of UDP datagrams sent from this entity.	Datagrams
5	udpTable	A table containing UDP listener information.	--
5.1	udpEntry	Information about a particular current UDP listener.	--
5.1.1	udpLocalAddress	The local IP address for this UDP listener.	--
5.1.2	udpLocalPort	The local port number for this UDP listener.	--

(8) SNMP group

The following table describes the standard MIB objects in the SNMP group.

Table 4–9: SNMP group (internet.mgmt.mib-2.snmp) (1.2.1.11)

ID	Object name	Contents	Units
1	snmpInPkts	The total number of Messages delivered to the SNMP entity from the transport service.	Messages
2	snmpOutPkts	The total number of SNMP Messages which were passed from the SNMP protocol entity to the transport service.	Messages

ID	Object name	Contents	Units
3	snmpInBadVersions	The total number of SNMP Messages which were delivered to the SNMP protocol entity and were for an unsupported SNMP version.	Messages
4	snmpInBadCommunityNames	The total number of SNMP Messages delivered to the SNMP protocol entity which used a SNMP community name not known to said entity.	Messages
5	snmpInBadCommunityUses	The total number of SNMP Messages delivered to the SNMP protocol entity which represented an SNMP operation which was not allowed by the SNMP community named in the Message.	Messages
6	snmpInASNParseErrs	The total number of ASN.1 or BER errors encountered by the SNMP protocol entity when decoding received SNMP Messages.	Messages
8	snmpInTooBigs	The total number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is tooBig.	Messages
9	snmpInNoSuchNames	The total number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is noSuchName.	Messages
10	snmpInBadValues	The total number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is badValue.	Messages
11	snmpInReadOnlys	The total number valid SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is readOnly.	Messages
12	snmpInGenErrs	The total number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is genErr.	Messages
13	snmpInTotalReqVars	The total number of MIB objects which have been retrieved successfully by the SNMP protocol entity as the result of receiving valid SNMP Get-Request and Get-Next PDUs.	--
14	snmpInTotalSetVars	The total number of MIB objects which have been altered successfully by the SNMP protocol entity as the result of receiving valid SNMP Set-Request PDUs.	--
15	snmpInGetRequests	The total number of SNMP Get-Request PDUs which have been accepted and processed by the SNMP protocol entity.	--
16	snmpInGetNexts	The total number of SNMP Get-Next PDUs which have been accepted and processed by the SNMP protocol entity.	--
17	snmpInSetRequests	The total number of SNMP Set-Request PDUs which have been accepted and processed by the SNMP protocol entity.	--
18	snmpInGetResponses	The total number of SNMP Get-Response PDUs which have been accepted and processed by the SNMP protocol entity.	--
19	snmpInTraps	The total number of SNMP Trap PDUs which have been accepted and processed by the SNMP protocol entity.	--
20	snmpOutTooBigs	The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field is tooBig.	Messages

ID	Object name	Contents	Units
21	snmpOutNoSuchNames	The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status is <code>noSuchName</code> .	--
22	snmpOutBadValues	The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field is <code>badValue</code> .	--
24	snmpOutGenErrs	The total number of SNMP PDUs which were generated by the SNMP protocol entity and for which the value of the error-status field is <code>genErr</code> .	--
25	snmpOutGetRequests	The total number of SNMP Get-Request PDUs.	--
26	snmpOutGetNexts	The total number of SNMP Get-Next PDUs.	--
27	snmpOutSetRequests	The total number of SNMP Set-Request PDUs.	--
28	snmpOutGetResponses	The total number of SNMP Get-Response PDUs.	--
29	snmpOutTraps	The total number of SNMP Trap PDUs.	--
30	snmpEnableAuthenTraps	Indicates whether the SNMP Agent process is permitted to generate authentication-failure traps. enabled (1), disabled (2)	--

4.1.3 Implementation of standard MIB objects

This section describes the implementation status of system-related standard MIB objects.

For the implementation status of the standard MIB objects (`interfaces`, `at`, `ip`, `icmp`, `tcp`, and `udp`) of the native agent, see the documentation for each OS. Acquire the standard MIB objects of the native agent using one of the following methods:

- For HP-UX (IPF)
Acquire the values from the OS-provided `mib2agt` and `ipv6agt` processes.
- For Solaris, AIX, and Linux
Acquire the values from the native agent. If the native agent is not active, you cannot acquire the values.

(1) System group

The following table shows the implementation status of the standard MIB objects in the `System` group.

Table 4–10: Implementation of standard MIB objects (System group) (`internet.mgmt.mib-2.system`) (1.2.1.1)

Object name	MIB operation							
	HP-UX (IPF)		Solaris		AIX		Linux	
	get	set	get	set	get	set	get	set
<code>sysDescr</code>	Y	--	Y	--	Y	--	Y	--
<code>sysObjectID</code>	Y	--	Y	--	Y	--	Y	--
<code>sysUpTime</code>	Y	--	Y	--	Y	--	Y	--

Object name	MIB operation							
	HP-UX (IPF)		Solaris		AIX		Linux	
	get	set	get	set	get	set	get	set
sysContact	Y	Y	Y	Y	Y	Y	Y	Y
sysName	Y	Y	Y	Y	Y	Y	Y	Y
sysLocation	Y	Y	Y	Y	Y	Y	Y	Y
sysServices	Y	--	Y	--	Y	--	Y	--

Legend:

Y: A get or set operation can get or set the value of this MIB object.

--: No access permission. A noSuchName error is returned.

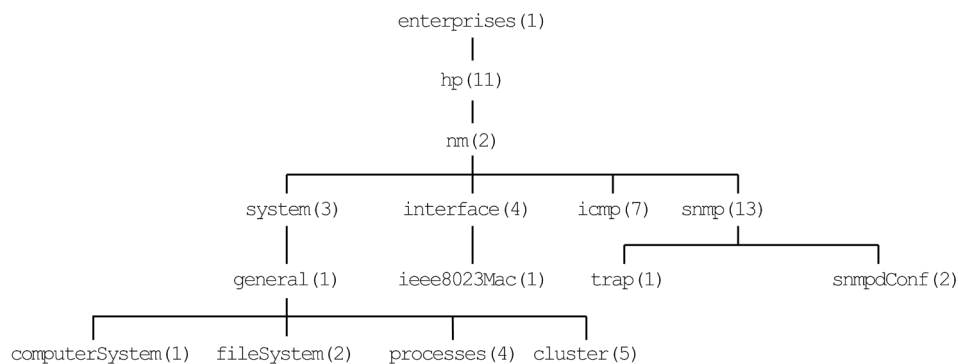
4.2 Hewlett-Packard enterprise-specific MIB objects

This section lists the Hewlett-Packard enterprise-specific MIB objects implemented by SNMP Agent and explains their implementation status.

4.2.1 Organization of Hewlett-Packard enterprise-specific MIB objects

The following figure shows the organization of the Hewlett-Packard enterprise-specific MIB objects.

Figure 4–2: Organization of Hewlett-Packard enterprise-specific MIB objects



The following table indicates, for each group, the tables that describe the Hewlett-Packard enterprise-specific MIB objects and their implementation statuses.

Table 4–11: Referenced locations for Hewlett-Packard enterprise-specific MIB objects

Hewlett-Packard enterprise-specific MIB object group				Referencing destination	
				MIB object description	MIB object implementation status
enterprises.hp.nm	system	general	computerSystem	Table 4-12	Table 4-20
			fileSystem	Table 4-13	Table 4-21
			processes	Table 4-14	Table 4-22
			cluster	Table 4-15	Table 4-23
	interface	ieee8023Mac		Table 4-16	Table 4-24
	icmp			Table 4-17	Table 4-25
	snmp	trap		Table 4-18	Table 4-26
		snmpdConf		Table 4-19	Table 4-27

4.2.2 Description of Hewlett-Packard enterprise-specific MIB objects

This subsection describes the Hewlett-Packard enterprise-specific MIB objects in each group. The tables in this subsection use the following legend:

Legend:

--: Not applicable

You can also view MIB object descriptions in `/var/opt/OV/share/snmp_mibs/eagent/hp-unix`.

(1) computerSystem group

The following table describes the Hewlett-Packard enterprise-specific MIB objects in the `computerSystem` group.

Table 4–12: `computerSystem` group (`enterprises.hp.nm.system.general.computerSystem`)
(1.11.2.3.1.1)

ID	Object name	Contents	Units
1	<code>computerSystemUpTime</code>	The time since the last boot.	Hundredths of a second
2	<code>computerSystemUsers</code>	The number of users logged on to system.	--
3	<code>computerSystemAvgJobs1</code>	The average number of jobs in the execution queue in the last 1 minute x 100. The average number of jobs in the execution queue means the average number of processes and threads that were in execution status or executable status in the last 1 minute. For example, if the average number of jobs in the execution queue is 1, this means that there was an average of 1 process or thread that was in execution status or executable status in the last 1 minute, and therefore it can be assumed that the CPU was always executing a process.	--
4	<code>computerSystemAvgJobs5</code>	The average number of jobs in the last 5 minutes * 100.	--
5	<code>computerSystemAvgJobs15</code>	The average number of jobs in the last 15 minutes * 100.	--
6	<code>computerSystemMaxProc</code>	The maximum number of processes allowed in system. Implemented.	--
7	<code>computerSystemFreeMemory#1</code>	Free memory in the physical memory.	Kilobytes
8	<code>computerSystemPhysMemory</code>	Physical memory.	Kilobytes
9	<code>computerSystemMaxUserMem</code>	Maximum user memory.	Kilobytes
10	<code>computerSystemSwapConfig#2, #3</code>	Size of the device swap space.	Kilobytes#4
11	<code>computerSystemEnabledSwap</code>	Enabled via <code>swapon</code> .	Kilobytes
12	<code>computerSystemFreeSwap#2</code>	Size of the actual free swap space.	Kilobytes#4
13	<code>computerSystemUserCPU#5</code>	The CPU time used in the user mode with a <code>nice</code> value of 21 or above. In Solaris and AIX, the CPU time used in the user mode after startup of SNMP Agent.	Hundredths of a second
14	<code>computerSystemSysCPU#5</code>	The CPU time used in the kernel mode. In Solaris and AIX, the CPU time used in the kernel mode after startup of SNMP Agent.	Hundredths of a second
15	<code>computerSystemIdleCPU#5</code>	CPU idle time. In Solaris and AIX, the CPU idle time after startup of SNMP Agent.	Hundredths of a second
16	<code>computerSystemNiceCPU#5</code>	The CPU time used in the user mode with a <code>nice</code> value of 20 or smaller. In Solaris and AIX, the CPU time used in the user mode after startup of SNMP Agent.	Hundredths of a second

#1

Note the following about `computerSystemFreeMemory`:

Time required to obtain

Six seconds or longer is required to obtain the `computerSystemFreeMemory` value in Solaris. Therefore, if the manager system issues SNMP requests addressed to SNMP Agent, specify a time-out value of 6 seconds or longer in the manager system.

For details about the amount of free space in physical memory in Solaris, AIX, and Linux, see [2.13 Notes about the amount of free space in physical memory](#).

#2

The following indicates whether the objects `computerSystemSwapConfig` and `computerSystemFreeSwap` of each OS include physical memory.

HP-UX (IPF) and Linux

Physical memory is not included.

Solaris

Physical memory is included.

AIX

In AIX, the actual paging space usage status is returned. Physical memory is not included.

#3

Note the following about the swap space in Solaris:

In the swap space in Solaris, the swap area on the disk contains real memory that is not used. In real memory, a virtual storage area is dynamically allocated. Therefore, the `computerSystemSwapConfig` value varies dynamically.

#4

In AIX, the unit is bytes.

#5

For details about CPU information, see [2.15 Notes about CPU information](#).

(2) fileSystem group

The following table describes the Hewlett-Packard enterprise-specific MIB objects in the `fileSystem` group.

Table 4–13: `fileSystem` group (`enterprises.hp.nm.system.general.fileSystem`) (1.11.2.3.1.2)

ID	Object name	Contents	Units
1	<code>fileSystemMounted</code>	The number of file systems mounted.	--
2	<code>fileSystemTable</code>	The file system table.	--
2.1	<code>fileSystemEntry</code>	Each entry contains objects for a particular file system.	--
2.1.1	<code>fileSystemID1</code>	The first file system ID.	--
2.1.2	<code>fileSystemID2</code>	The second file system ID.	--
2.1.3	<code>fileSystemName</code>	The name of mounted file system.	--
2.1.4	<code>fileSystemBlock</code>	Total blocks in file system.	Blocks
2.1.5	<code>fileSystemBfree</code>	Free blocks in file system.	Blocks
2.1.6	<code>fileSystemBavail</code>	Free blocks avail to non-superuser.	Blocks
2.1.7	<code>fileSystemBsize</code>	The fundamental file system block size.	Bytes
2.1.8	<code>fileSystemFiles</code>	Total inodes in file system.	--

ID	Object name	Contents	Units
2.1.9	fileSystemFfree	Free inodes in file system.	--
2.1.10	fileSystemDir	The file system path prefix.	--

Important

Note the following about the Hewlett-Packard enterprise-specific MIB objects in the `fileSystem` group:

- In AIX and Linux, the setting in `/etc/SnmpAgent.d/esafilesys.conf` becomes valid for the `fileSystem` group as for the `fileSystem64` group. For details about the settings in `/etc/SnmpAgent.d/esafilesys.conf` in the `fileSystem64` group, see [2.12.2 Settings for suppressing an invalid shared disk capacity response \(for AIX and Linux\)](#).

Note that in Solaris and HP-UX (IPF), the setting in `/etc/SnmpAgent.d/esafilesys.conf` is invalid.

- In AIX, a file system of up to 4 petabytes can be built using the JFS2 file system. However, the table below shows the maximum size of the file system that can be utilized in MIB in the `hp.nm.system.general.fileSystem` group for SNMP Agent. The maximum total number of inodes is $2^{31} - 1$.

File system block size (bytes)	File system size (terabytes)
512	1
1,024	2
4,096	8

- For details about how to prevent unnecessary file system information from being returned in Linux, see [2.16 Settings to prevent responses with information about file systems for which a response is not required \(for Linux\)](#).

- In Linux, The name of mounted file system information item in the file system information in the `hp.nm.system.general.fileSystem` group is different from the name output by the `df` command, because this software gets this information by referencing the information in `/etc/fstab`. This information corresponds to `/etc/fstab` information.

The following is an example.

Example:

- MIB information

```
hp.nm.system.general.fileSystem.fileSystemTable.fileSystemEntry.fileSystemName.1.1
: DISPLAY STRING- (ascii): LABEL=/
```

- `/etc/fstab` information

```
LABEL=/          /          ext2  defaults    1 1
```

- `df` command information

```
df
Filesystem      1k-blocks      Used Available Use% Mounted on
/dev/hda1        6048320      727156   5013924  13% /
```

- The `hp.nm.system.general.fileSystem` group of Solaris has not added to the MIB value a file system in the `tmpfs` file system format. Consequently, file systems in the `tmpfs` file system format cannot be monitored.

(3) processes group

The following table describes the Hewlett-Packard enterprise-specific MIB objects in the `processes` group.

Table 4–14: `processes` group (`enterprises.hp.nm.system.general.processes`) (1.11.2.3.1.4)

ID	Object name	Contents	Units
1	<code>processNum</code>	The number of processes running.	--
2	<code>processTable</code>	The processes Table.	--
2.1	<code>processEntry</code>	Each entry contains information about a process running on the system.	--
2.1.1	<code>processPID</code>	The process ID (pid).	--
2.1.2	<code>processIdx</code>	The index for <code>pstat()</code> requests.	--
2.1.3	<code>processUID</code>	The process User ID.	--
2.1.4	<code>processPPID</code>	The parent process ID.	--
2.1.5	<code>processDsize</code>	The process data size.	Pages
2.1.6	<code>processTsize</code>	The process text size.	Pages
2.1.7	<code>processSsize</code>	The process stack size.	Pages
2.1.8	<code>processNice</code> ^{#1}	The process <code>nice</code> value.	--
2.1.9	<code>processMajor</code>	The process tty major number.	--
2.1.10	<code>processMinor</code>	The process tty minor number. SunOS - not implemented.	--
2.1.11	<code>processPgrp</code>	The process group of this process.	--
2.1.12	<code>processPrio</code>	The process priority.	--
2.1.13	<code>processAddr</code>	The address of the process (in memory).	--
2.1.14	<code>processCPU</code>	The processor utilization for scheduling.	--
2.1.15	<code>processUtime</code>	The user time spent executing.	Hundredths of a second
2.1.16	<code>processStime</code>	The system time spent executing in the system (kernel) mode.	Hundredths of a second
2.1.17	<code>processStart</code>	The time elapsed since the process started.	Seconds
2.1.18	<code>processFlags</code> ^{#2}	The process flag. incore (1), sys (2), locked (4), trace (8), trace2 (16)	--
2.1.19	<code>processStatus</code>	The process status. sleep (1), run (2), stop (3), zombie (4), other (5), idle (6)	--
2.1.20	<code>processWchan</code>	If <code>processStatus</code> is sleep, value sleeping on.	--
2.1.21	<code>processProcNum</code>	The processor this process last run on.	--
2.1.22	<code>processCmd</code>	The command the process is running.	--

ID	Object name	Contents	Units
2.1.23	processTime	The resident time for scheduling.	Seconds
2.1.24	processCPUticks	Ticks of CPU time.	--
2.1.25	processCPUticksTotal	The total number of ticks of CPU time consumed by a process since its generation.	--
2.1.26	processFss	The Fair Share Scheduler Group.	--
2.1.27	processPctCPU	The Percent CPU * 100 for this process.	%
2.1.28	processRssize	The Resident Set Size for process (private pages).	Pages
2.1.29	processSUID	The saved UID.	--
2.1.30	processUname	The user name.	--
2.1.31	processTTY	The process TTY.	--

#1

In Linux, `nice` values range from -20 to 19.

The `processNice` MIB value is acquired by a Gauge type that does not support negative values.

As such, for the `processNice` MIB value, the SNMP agent returns the `nice` value plus 20 (that is, a value between 0 and 39).

#2

In Solaris, the value of `process.processTable.processEntry.processFlag` has no meaning.

(4) cluster group

The following table describes the Hewlett-Packard enterprise-specific MIB objects in the `cluster` group.

Table 4–15: cluster group (enterprises.hp.nm.system.general.cluster) (1.11.2.3.1.5)

ID	Object name	Contents	Units
1	isClustered	Identifies whether the machine is clustered or not. standalone (1), rootserver (2), cnode (3)	--
2	clusterTable	A list of nodes on the cluster.	--
2.1	clusterEntry	Each entry contains information about the clustered node.	--
2.1.1	clusterID	The cnode id.	--
2.1.2	clusterMachineID	The cnode machine id.	--
2.1.3	clusterType	The cnode type (r or c).	--
2.1.4	clusterCnodeName	The cnode name.	--
2.1.5	clusterSwapServingCnode	The swap serving cnode.	--
2.1.6	clusterKcsp	KCSP.	--
2.1.7	clusterCnodeAddress	The cnode IP Address.	--
3	clusterCnodeID	The machine's cnode id.	--

(5) ieee8023Mac group

The following table describes the Hewlett-Packard enterprise-specific MIB objects in the `ieee8023Mac` group.

Table 4–16: ieee8023Mac group (enterprises.hp.nm.interface.ieee8023Mac) (1.11.2.4.1)

ID	Object name	Contents	Units
1	ieee8023MacTable	A list of IEEE 802.3 Interface entries.	--
1.1	ieee8023MacEntry	Each entry contains statistics for ieee 802.3 interfaces.	--
1.1.1	ieee8023MacIndex	The index value that uniquely identifies the interface to which this entry is applicable.	--
1.1.2	ieee8023MacTransmitted	The number of frames successfully transmitted.	Frame
1.1.3	ieee8023MacNotTransmitted	The number of frames not transmitted.	Frame
1.1.4	ieee8023MacDeferred	The number of frames deferred because the medium was busy.	Frame
1.1.5	ieee8023MacCollisions	The total number of transmit attempts that were retransmitted due to collisions.	--
1.1.6	ieee8023MacSingleCollisions	The number of transmit attempts that are involved in a single collision and are subsequently transmitted successfully.	--
1.1.7	ieee8023MacMultipleCollisions	The number of transmit attempts that are involved in between 2 and 5 collision attempts and are subsequently transmitted successfully.	--
1.1.8	ieee8023MacExcessCollisions	The number of transmit attempts that are involved in more than 15 collision attempts and are subsequently transmitted successfully.	--
1.1.9	ieee8023MacLateCollisions	The number of transmit attempts aborted because a collision occurred after the allotted channel time had elapsed.	--
1.1.10	ieee8023MacCarrierLostErrors	The number of times that carrier sense was lost when attempting to transmit.	--
1.1.11	ieee8023MacNoHeartBeatErrors	The number of times no heart beat was indicated after a transmission.	--
1.1.12	ieee8023MacFramesReceived	The number of frames successfully received.	Frame
1.1.13	ieee8023MacUndeliverableFramesReceived	The number of frames received that were not delivered because the software buffer was overrun when frames were sent faster than they could be received.	Frame
1.1.14	ieee8023MacCRCErrors	The number of Cyclical Redundancy Check (CRC) errors detected.	--
1.1.15	ieee8023MacAlignmentErrors	The number of frames received that were both misaligned and had bad CRC.	Frame
1.1.16	ieee8023MacResourceErrors	The number of frames received that were lost due to lack of resources.	Frame
1.1.17	ieee8023MacControlFieldErrors	The number of frames received with errors in the control field.	Frame
1.1.18	ieee8023MacUnknownProtocolErrors	The number of frames dropped because the type field or sap field referenced an invalid protocol.	Frame
1.1.19	ieee8023MacMulticastsAccepted	The number of accepted multi cast addresses.	--

(6) icmp group

The following table describes the Hewlett-Packard enterprise-specific MIB objects in the `icmp` group.

Table 4–17: `icmp` group (`enterprises.hp.nm.icmp`) (1.11.2.7)

ID	Object name	Contents	Units
1	<code>icmpEchoReq#</code>	The number of seconds it takes to respond to an icmp echo request and error information.	--

#: Explanation of `icmpEchoReq`:

An icmp echo request is sent from the host on which SNMP Agent is installed to a specified host. The time in milliseconds required for this response is treated as a MIB value. If the icmp echo request encounters an error, this MIB value is as follows.

- 1: An internal error occurred.
- 2: The icmp echo request reached its time-out.
- 3: The echo reply is incorrect.
- 4: The packet size is too large.
- 5: The time-out value is incorrect.

This MIB value can be obtained with an SNMP GET request only. It cannot be obtained with an SNMP GET-NEXT request. When issuing an SNMP GET request, from the host on which SNMP Agent is installed, specify the IP address of the icmp echo request source host, the packet size (in bytes) of the icmp echo request, and a time-out value (in seconds) for the icmp echo request. If the IP address is `a1.a2.a3.a4`, the packet size is `s`, and the time-out value is `t`, then the request format is `icmpEchoReq.s.t.a1.a2.a3.a4`.

An example follows.

Example:

To send an icmp echo request to IP address `15.2.112.113` with the specification of a time-out value of 8 seconds and a packet size of 75 bytes, specify the following:

```
icmpEchoReq.75.8.15.2.112.113
```

(7) trap group

The following table describes the Hewlett-Packard enterprise-specific MIB objects in the `trap` group.

Table 4–18: `trap` group (`enterprises.hp.nm.snmp.trap`) (1.11.2.13.1)

ID	Object name	Contents	Units
1	<code>trapDestinationNum</code>	The number of trap destinations.	--
2	<code>trapDestinationTable</code>	A list of addresses to which the agent sends traps.	--
2.1	<code>trapDestinationEntry</code>	Each entry contains the address of a management station.	--
2.1.1	<code>trapDestination</code>	The address to which the agent sends traps.	--

(8) snmpdConf group

The following table describes the Hewlett-Packard enterprise-specific MIB objects in the `snmpdConf` group.

Table 4–19: `snmpdConf` group (`enterprises.hp.nm.snmp.snmpdConf`) (1.11.2.13.2)

ID	Object name	Contents	Units
1	<code>snmpdConfRespond</code>	SNMP Agent was configured to respond to all objects if <code>snmpdConfRespond</code> is true. true (1), false (2)	--
2	<code>snmpdReConfigure</code>	The agent will re-configure itself if <code>snmpdReConfigure</code> is set to reset (1).	--
3	<code>snmpdFlag</code>	Indicates the capability of the agent.	--

ID	Object name	Contents	Units
		removetraps (1), netwareproxy (2)	
4	snmpdLogMask	The agent's log mask.	--
5	snmpdVersion	The agent's version number.	--
6	snmpdStatus	Indicates the status of the agent. up (1), down (2)	--
7	snmpdSize	The size of the agent data segment.	Bytes
9	snmpdWhatString	Agent profile. Product name, version, date, and copyright	--

4.2.3 Implementation of Hewlett-Packard enterprise-specific MIB objects

This section describes the implementation status of Hewlett-Packard enterprise-specific MIB objects. These tables use the following legends:

Legends:

Y: A `get` or `set` operation can get or set the value of this MIB object.

N: A `get` or `set` operation cannot get or set the value of this MIB object (a `noSuchName` error is returned).

F (*value*): The object returns a fixed value indicated by *value*.

--: No access permission. A `noSuchName` error is returned.

(1) computerSystem group

The following table shows the implementation status of the Hewlett-Packard enterprise-specific MIB objects in the `computerSystem` group.

Table 4–20: Implementation of Hewlett-Packard enterprise-specific MIB objects (`computerSystem` group) (`enterprises.hp.nm.system.general.computerSystem`) (1.11.2.3.1.1)

Object name	MIB operation							
	HP-UX (IPF)		Solaris		AIX		Linux	
	get	set	get	set	get	set	get	set
<code>computerSystemUpTime</code>	Y	--	Y	--	Y	--	Y	--
<code>computerSystemUsers</code>	Y	--	Y	--	Y	--	Y	--
<code>computerSystemAvgJobs1</code>	Y	--	Y	--	Y	--	Y	--
<code>computerSystemAvgJobs5</code>	Y	--	Y	--	Y	--	Y	--
<code>computerSystemAvgJobs15</code>	Y	--	Y	--	Y	--	Y	--
<code>computerSystemMaxProc</code>	Y	--	N	--	--	--	N	--
<code>computerSystemFreeMemory</code>	Y	--	Y	--	Y	--	Y	--
<code>computerSystemPhysMemory</code>	Y	--	Y	--	Y	--	Y	--
<code>computerSystemMaxUserMem</code>	Y	--	N	--	--	--	N	--
<code>computerSystemSwapConfig</code>	Y	--	Y	--	Y	--	Y	--

Object name	MIB operation							
	HP-UX (IPF)		Solaris		AIX		Linux	
	get	set	get	set	get	set	get	set
computerSystemEnabledSwap	Y	--	N	--	--	--	N	--
computerSystemFreeSwap	Y	--	Y	--	Y	--	Y	--
computerSystemUserCPU	Y	--	Y	--	Y	--	Y	--
computerSystemSysCPU	Y	--	Y	--	Y	--	Y	--
computerSystemIdleCPU	Y	--	Y	--	Y	--	Y	--
computerSystemNiceCPU	Y	--	N	--	--	--	Y	--

(2) fileSystem group

The following table shows the implementation status of the Hewlett-Packard enterprise-specific MIB objects in the `fileSystem` group.

Table 4–21: Implementation of Hewlett-Packard enterprise-specific MIB objects (fileSystem group) (enterprises.hp.nm.system.general.fileSystem) (1.11.2.3.1.2)

Object name		MIB operation							
		HP-UX (IPF)		Solaris		AIX		Linux	
		get	set	get	set	get	set	get	set
fileSystemMounted		Y	--	Y	--	Y	--	Y	--
fileSystemTable. fileSystemEntry. ~	fileSystemID1	Y	--	Y	--	Y	--	Y	--
	fileSystemID2	Y	--	Y	--	Y	--	Y	--
	fileSystemName	Y	--	Y	--	Y	--	Y	--
	fileSystemBlock	Y	--	Y	--	Y	--	Y	--
	fileSystemBfree	Y	--	Y	--	Y	--	Y	--
	fileSystemBavail	Y	--	Y	--	Y	--	Y	--
	fileSystemBsize	Y	--	Y	--	Y	--	Y	--
	fileSystemFiles	Y	--	Y	--	Y	--	Y	--
	fileSystemFfree	Y	--	Y	--	Y	--	Y	--
fileSystemDir	Y	--	Y	--	Y	--	Y	--	

(3) processes group

The following table shows the implementation status of the Hewlett-Packard enterprise-specific MIB objects in the `processes` group.

Table 4–22: Implementation of Hewlett-Packard enterprise-specific MIB objects (processes group) (enterprises.hp.nm.system.general.processes) (1.11.2.3.1.4)

Object name		MIB operation							
		HP-UX (IPF)		Solaris		AIX		Linux	
		get	set	get	set	get	set	get	set
processNum		Y	--	Y	--	Y	--	Y	--
processTable. processEntry.~	processPID	Y	--	Y	--	Y	--	Y	--
	processIdx	Y	--	N	--	--	--	N	--
	processUID	Y	--	Y	--	--	--	Y	--
	processPPID	Y	--	Y	--	Y	--	Y	--
	processDsize	Y	--	N	--	--	--	N	--
	processTsize	Y	--	N	--	--	--	N	--
	processSsize	Y	--	N	--	--	--	N	--
	processNice	Y	--	Y	--	Y	--	Y	--
	processMajor	Y	--	N	--	--	--	N	--
	processMinor	Y	--	N	--	--	--	N	--
	processPgrp	Y	--	Y	--	--	--	Y	--
	processPrio	Y	--	Y	--	Y	--	Y	--
	processAddr	Y	--	N	--	--	--	N	--
	processCPU	Y	--	Y	--	--	--	N	--
	processUtime	Y	--	Y	--	--	--	N	--
	processStime	Y	--	Y	--	--	--	N	--
	processStart	Y	--	Y	--	--	--	N	--
	processFlags	Y	--	Y	--	--	--	N	--
	processStatus	Y	--	Y	--	--	--	N	--
	processWchan	Y	--	Y	--	--	--	N	--
	processProcNum	Y	--	Y	--	--	--	N	--
	processCmd	Y	--	Y	--	Y	--	Y	--
	processTime	Y	--	F(0)	--	--	--	N	--
	processCPUticks	Y	--	N	--	--	--	N	--
	processCPUticksTotal	Y	--	Y	--	Y	--	Y	--
	processFss	Y	--	N	--	--	--	N	--
processPctCPU	Y	--	N	--	--	--	N	--	
processRssize	Y	--	Y	--	--	--	N	--	
processSUID	Y	--	Y	--	--	--	N	--	
processUname	Y	--	Y	--	Y	--	Y	--	

Object name		MIB operation							
		HP-UX (IPF)		Solaris		AIX		Linux	
		get	set	get	set	get	set	get	set
	processTTY	N	--	N	--	N	--	N	--

(4) cluster group

The following table shows the implementation status of the Hewlett-Packard enterprise-specific MIB objects in the `cluster` group.

Table 4–23: Implementation of Hewlett-Packard enterprise-specific MIB objects (cluster group) (enterprises.hp.nm.system.general.cluster) (1.11.2.3.1.5)

Object name		MIB operation							
		HP-UX (IPF)		Solaris		AIX		Linux	
		get	set	get	set	get	set	get	set
	isClustered	Y	--	N	--	--	--	--	--
clusterTable. clusterEntry.~	clusterID	Y	--	N	--	--	--	--	--
	clusterMachineID	Y	--	N	--	--	--	--	--
	clusterType	Y	--	N	--	--	--	--	--
	clusterCnodeName	Y	--	N	--	--	--	--	--
	clusterSwapServingCnode	Y	--	N	--	--	--	--	--
	clusterKcsp	Y	--	N	--	--	--	--	--
	clusterCnodeAddress	Y	--	N	--	--	--	--	--
	clusterCnodeID	Y	--	N	--	--	--	--	--

(5) ieee8023Mac group

The following table shows the implementation status of the Hewlett-Packard enterprise-specific MIB objects in the `ieee8023Mac` group.

Table 4–24: Implementation of Hewlett-Packard enterprise-specific MIB objects (ieee8023Mac group) (enterprises.hp.nm.interface.ieee8023Mac) (1.11.2.4.1)

Object name		MIB operation							
		HP-UX (IPF)		Solaris		AIX		Linux	
		g et	s et	g et	s et	g et	s et	g et	s et
ieee8023MacTable. ieee8023MacEntry.~	ieee8023MacIndex	Y	--	N	--	--	--	--	--
	ieee8023MacTransmitted	Y	--	N	--	--	--	--	--

Object name		MIB operation							
		HP-UX (IPF)		Solaris		AIX		Linux	
		g et	s et	g et	s et	g et	s et	g et	s et
	ieee8023MacNotTransmitted	Y	--	N	--	--	--	--	--
	ieee8023MacDeferred	Y	--	N	--	--	--	--	--
	ieee8023MacCollisions	Y	--	N	--	--	--	--	--
	ieee8023MacSingleCollisions	Y	--	N	--	--	--	--	--
	ieee8023MacMultipleCollisions	Y	--	N	--	--	--	--	--
	ieee8023MacExcessCollisions	Y	--	N	--	--	--	--	--
	ieee8023MacLateCollisions	Y	--	N	--	--	--	--	--
	ieee8023MacCarrierLostErrors	Y	--	N	--	--	--	--	--
	ieee8023MacNoHeartBeatErrors	Y	--	N	--	--	--	--	--
	ieee8023MacFramesReceived	Y	--	N	--	--	--	--	--
	ieee8023MacUndeliver-ableFramesReceived	Y	--	N	--	--	--	--	--
	ieee8023MacCRCErrors	Y	--	N	--	--	--	--	--
	ieee8023MacAlignmentErrors	Y	--	N	--	--	--	--	--
	ieee8023MacResourceErrors	Y	--	N	--	--	--	--	--
	ieee8023MacControlFieldErrors	Y	--	N	--	--	--	--	--
	ieee8023MacUnknownProtocolErrors	Y	--	N	--	--	--	--	--
	ieee8023MacMulticastsAccepted	Y	--	N	--	--	--	--	--

(6) icmp group

The following table shows the implementation status of the Hewlett-Packard enterprise-specific MIB objects in the `icmp` group.

Table 4–25: Implementation of Hewlett-Packard enterprise-specific MIB objects (`icmp` group) (`enterprises.hp.nm.icmp`) (1.11.2.7)

Object name	MIB operation							
	HP-UX (IPF)		Solaris		AIX		Linux	
	get	set	get	set	get	set	get	set
<code>icmpEchoReq</code>	Y	--	Y	--	Y	--	Y	--

(7) trap group

The following table shows the implementation status of the Hewlett-Packard enterprise-specific MIB objects in the `trap` group.

Table 4–26: Implementation of Hewlett-Packard enterprise-specific MIB objects (trap group) (enterprises.hp.nm.snmp.trap) (1.11.2.13.1)

Object name		MIB operation							
		HP-UX (IPF)		Solaris		AIX		Linux	
		get	set	get	set	get	set	get	set
trapDestinationNum		Y	--	Y	--	Y	--	Y	--
trapDestinationTable. trapDestinationEntry.~	trapDestin ation	Y	Y	Y	Y	Y	Y	Y	Y

(8) snmpdConf group

The following table shows the implementation status of the Hewlett-Packard enterprise-specific MIB objects in the snmpdConf group.

Table 4–27: Implementation of Hewlett-Packard enterprise-specific MIB objects (snmpdConf group) (enterprises.hp.nm.snmp.snmpdConf) (1.11.2.13.2)

Object name		MIB operation							
		HP-UX (IPF)		Solaris		AIX		Linux	
		get	set	get	set	get	set	get	set
snmpdConfRespond		Y	Y	Y	Y	Y	Y	Y	Y
snmpdReConfigure		Y	Y	Y	Y	Y	Y	Y	Y
snmpdFlag		Y	--	Y	--	Y	--	Y	--
snmpdLogMask		Y	Y	Y	Y	Y	Y	Y	Y
snmpdVersion		Y	--	Y	--	Y	--	Y	--
snmpdStatus		Y	Y	Y	Y	Y	Y	Y	Y
snmpdSize		Y	--	Y	--	Y	--	Y	--
snmpdWhatString		Y	--	Y	--	Y	--	Y	--

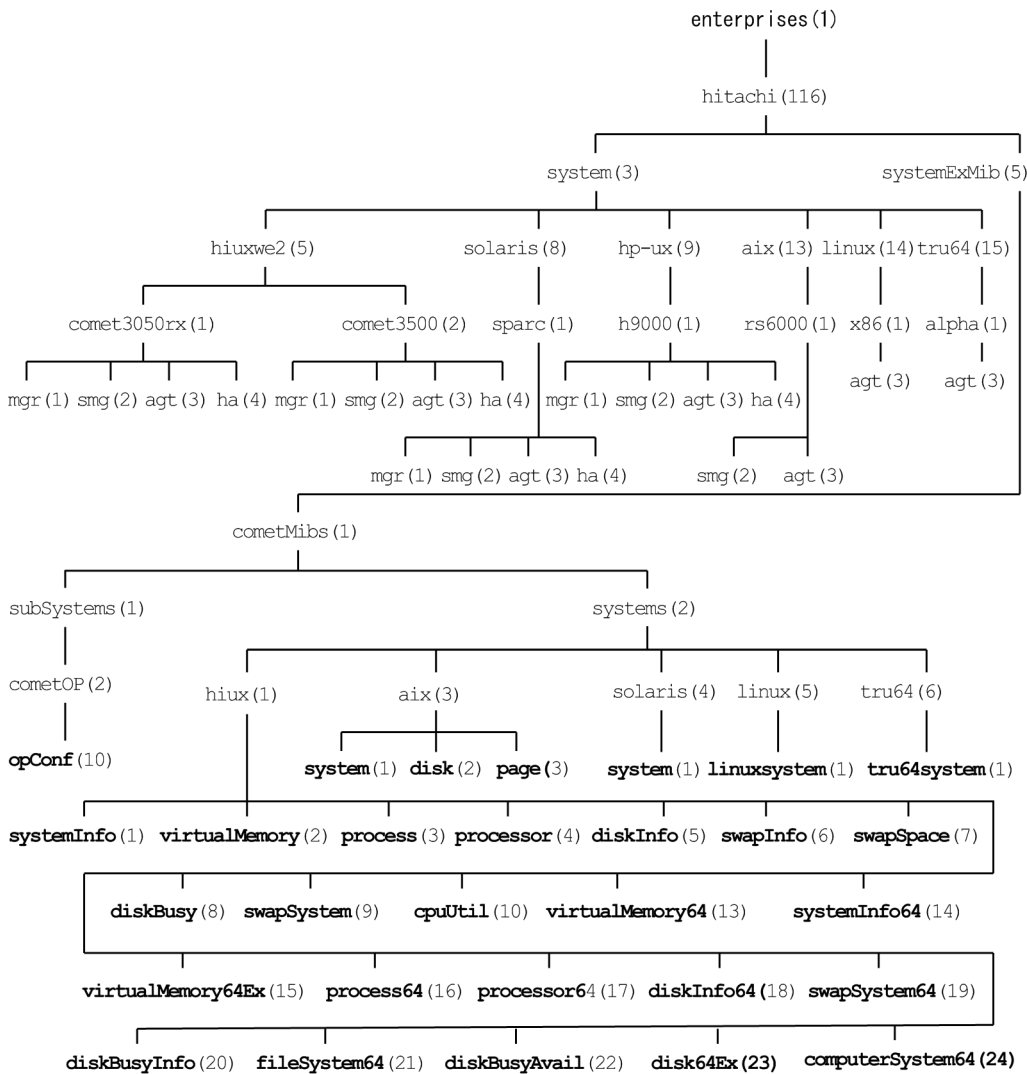
4.3 Hitachi enterprise-specific MIB objects

This section lists the Hitachi enterprise-specific MIB objects implemented by SNMP Agent and explains their implementation status.

4.3.1 Organization of Hitachi enterprise-specific MIB objects

The following figure shows the organization of the Hitachi enterprise-specific MIB objects.

Figure 4–3: Organization of Hitachi enterprise-specific MIB objects



Legend:

Bold text: This subsection explains the Hitachi enterprise-specific MIB objects shown in bold.

The following table indicates, for each group, the tables that describe the Hitachi enterprise-specific MIB objects and their implementation statuses.

Table 4–28: Referencing destinations for Hitachi enterprise-specific MIB objects

Hitachi enterprise-specific MIB object group				Referencing destination	
				MIB object description	MIB object implementation status
enterprises. hitachi.systemExMib.cometMibs	subSystems	cometOP	opConf	Table 4-29	Table 4-58
	systems	hiux	systemInfo	Table 4-30	Table 4-59
			virtualMemory	Table 4-31	Table 4-60
			process	Table 4-32	Table 4-61
			processor	Table 4-33	Table 4-62
			diskInfo	Table 4-34	Table 4-63
			swapInfo	Table 4-35	Table 4-64
			swapSpace	Table 4-36	Table 4-65
			diskBusy	Table 4-37	Table 4-66
			swapSystem	Table 4-38	Table 4-67
			cpuUtil	Table 4-39	Table 4-68
			virtualMemory64	Table 4-40	Table 4-69
			systemInfo64	Table 4-41	Table 4-70
			virtualMemory64Ex	Table 4-42	Table 4-71
			process64	Table 4-43	Table 4-72
			processor64	Table 4-44	Table 4-73
			diskInfo64	Table 4-45	Table 4-74
			swapSystem64	Table 4-46	Table 4-75
			diskBusyInfo	Table 4-47	Table 4-76
			fileSystem64	Table 4-48	Table 4-77
			diskBusyAvail	Table 4-49	Table 4-78
	disk64Ex	Table 4-50	Table 4-79		
	computerSystem64	Table 4-51	Table 4-80		
	aix#1	system	Table 4-52	Table 4-81	
		disk	Table 4-53		
		page	Table 4-54		
	solaris#2	system	Table 4-55	Table 4-82	
	linux#3	linuxsystem	Table 4-56	Table 4-83	

#1: MIB object group specific to an AIX system.

#2: MIB object group specific to a Solaris system.

#3: MIB object group specific to a Linux system.

4.3.2 Description of Hitachi enterprise-specific MIB objects

This subsection describes the Hitachi enterprise-specific MIB objects in each group. The tables in this subsection use the following legend:

Legend:

--: Not applicable

The following files in `/var/opt/OV/share/snmp_mibs/eagent` also provide information about the MIB objects:

- `hitachi-cometAgt`
- `hitachi-cometAgt-solaris`
- `hitachi-cometAgt-aix`
- `hitachi-cometAgt-linux`

(1) opConf group

The following table describes the Hitachi enterprise-specific MIB objects in the `opConf` group.

Table 4–29: `opConf` group
(`enterprises.hitachi.systemExMib.cometMibs.subSystems.cometOP.opConf`)
(1.116.5.1.1.2.10)

ID	Object name	Contents	Units
1	<code>opConfCharCode</code>	Character code used by control support for code conversion	--

(2) systemInfo group

The following table describes the Hitachi enterprise-specific MIB objects in the `systemInfo` group.

Table 4–30: `systemInfo` group
(`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.systemInfo`)
(1.116.5.1.2.1.1)

ID	Object name	Contents	Units
1	<code>systemInfoTTYMajor</code>	The major number of the console device.	--
2	<code>systemInfoTTYMinor</code>	The minor number of the console device.	--
3	<code>systemInfoBootTime</code>	The time since the last boot.	Hundredths of a second
4	<code>systemInfoActiveProcessors</code>	The number of processors which are running when system call is issued.	--
5	<code>systemInfoProcessors</code>	The maximum number of processors which had run at the same time since the last boot.	--
6	<code>systemInfoMaxProcesses</code>	The maximum number of processes which were allowed to run to system.	--
7	<code>systemInfoRunQueueProcesses</code>	The number of processes which are being loaded to main storage and waiting to run except system process.	--

ID	Object name	Contents	Units
8	systemInfoXferWaitProcesses	The number of processes which are waiting for data transfer from disk.	--
9	systemInfoPageInWaitProcesses	The number of processes which are waiting for pagein from disk.	--
10	systemInfoSleepProcesses	Number of sleeping processes in physical memory. However, the following processes are excluded: <ul style="list-style-type: none"> • Processes that are sleeping while waiting for I/O • Processes that have slept continuously for at least 20 seconds 	--
11	systemInfoSwapOutProcesses	The number of processes which are being swapped out, but it is only process which immediately becomes to runnable when be swapped in.	--
12	systemInfoPhysicalMemorySize	The number of bytes for physical memory.	Bytes
13	systemInfoPhysicalMemoryFreeSize	The number of bytes for free physical memory which is able to allocate to process.	Bytes
14	systemInfoVirtualMemoryProcessSize	The number of bytes for Virtual memory which are using for text, data and stack area by processes except system process.	Bytes
15	systemInfoVirtualMemoryWaitProcessSize	The number of bytes for virtual memory which are using by whole runnable processes.	Bytes
16	systemInfoPhysicalMemoryProcessSize	The number of bytes for physical memory which are using for text, data and stack area by whole processes.	Bytes
17	systemInfoPhysicalMemoryWaitProcessSize	The number of bytes for physical memory which are using by whole runnable processes.	Bytes
18	systemInfoCPUStates	The number of states which belong to system.	--
19	systemInfoOpenLogicalVolumes	The number of open logical volumes.	--
20	systemInfoOpenLogicalVolumeGrps	The number of open logical volume groups.	--
21	systemInfoAllocPBUFs	The number of PBUFs which are being allocated to logical volume.	--
22	systemInfoUsedPBUFs	The number of PBUFs which is being used to logical volume.	--
23	systemInfoMaxPBUFs	The maximum number of PBUFs which had been used to logical volume till now.	--
24	systemInfoActiveProcessEntries	The number of active process table entries in the number of all process table entries.	Entries
25	systemInfoActiveInodeEntries	The number of active inode table entries in the number of all inode table entries.	Entries
26	systemInfoActiveFileEntries	The number of active file table entries in the number of all file table entries.	Entries

(3) virtualMemory group

The following table describes the Hitachi enterprise-specific MIB objects in the virtualMemory group:

**Table 4–31: virtualMemory group
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.virtualMemory)
(1.116.5.1.2.1.2)**

ID	Object name	Contents	Units
1	vmPageSize	The number of bytes per page for virtual storage.	Bytes
2	vmDaemonFreePages	The number of pages which were been free by pageout daemon. This value is the average of once a second and updated every 5 seconds.	Pages
3	vmInterruptions	The number of device interruptions. This value is the average of once a second and updated every 5 seconds.	--
4	vmPageInPages	The number of pages which were paged in. This value is the average of once a second and updated every 5 seconds.	Pages
5	vmPageOutPages	The number of pages which were paged out. This value is the average of once a second and updated every 5 seconds.	Pages
6	vmPageReclaims	The total number of page reclaims. This value is the average of once a second and updated every 5 seconds.	--
7	vmTLBFlashes	The number of times a TLB flush occurred. This value is the average for one second and updated every 5 seconds.	--
8	vmDaemonScanPages	The number of pages which were scanned by pageout daemon. This value is the average of once a second and updated every 5 seconds.	Pages
9	vmContextSwitches	The number of context switches. This value is the average of once a second and updated every 5 seconds.	--
10	vmSystemCalls	The number of calls to system call. This value is the average of once a second and updated every 5 seconds.	--
11	vmXFileSystemFreelistPages	The number of pages which found in free list rather than in file system. This value is the average of once a second and updated every 5 seconds.	Pages
12	vmXSwapDeviceFreeListPages	The number of pages which found in freelist rather than on swapdevice. This value is the average of once a second and updated every 5 seconds.	Pages
13	vmFreeMemoryPages	The number of free memory pages. This value is the average of once a second and updated every 5 seconds.	Pages
14	vmTotalSwapIns	The number of swapins since the last boot.	--
15	vmTotalSwapOuts	The number of swapouts since the last boot.	--
16	vmTotalDaemonFreePages	The number of pages which have been free by pageout daemon since the last boot.	Pages
17	vmTotalDemandLoadPages	The number of pages which have been filled on demand from executable file since the last boot.	Pages
18	vmTotalPageFaults	The number of page faults since the last boot.	--
19	vmTotalInterruptions	The number of device interruptions since the last boot.	--
20	vmTotalIntransitPageFaults	The number of intransit blocking page faults since the last boot.	--
21	vmTotalDemandLoadCreatePages	The number of pages created which have been filled on demand since the last boot.	Pages
22	vmTotalZeroFillCreatePages	The number of new zero-filled pages created since the last boot.	Pages

ID	Object name	Contents	Units
23	vmTotalFreeListReclaimedPages	The number of pages reclaimed from freelist since the last boot.	Pages
24	vmTotalPageIns	The number of pageins since the last boot.	--
25	vmTotalPageOuts	The number of pageouts since the last boot.	--
26	vmTotalPageInPages	The number of pages which have been paged in since the last boot.	Pages
27	vmTotalPageOutPages	The number of pages which have been paged out since the last boot.	Pages
28	vmTotalSwapInPages	The number of pages which have been swapped in since the last boot.	Pages
29	vmTotalSwapOutPages	The number of pages which have been swapped out since the last boot.	Pages
30	vmTotalDaemonTicksNum	The number of pageouts by the pageout daemon since the last boot.	--
31	vmTotalContextSwitches	The number of context switches since the last boot.	--
32	vmTotalSystemCalls	The number of calls to system call since the last boot.	--
33	vmTotalTraps	The number of calls to trap since the last boot.	--
34	vmTotalXFileSystemFreeListPages	The number of pages which have found in freelist rather than in file system since the last boot.	Pages
35	vmTotalXSwapDeviceFreeListPages	The number of pages which have found in freelist rather than on swap device since the last boot.	Pages
36	vmTotalDemandZeroFillPages	The number of pages which have been zero filled on demand since the last boot.	Pages
37	vmTotalDaemonScanPages	The number of pages which have been scanned by pageout daemon since the last boot.	Pages
38	vmTotalReclaimedPages	The number of page reclaims since the last boot.	Pages
39	vmTotalDeficitPages	The number of pages that are estimated to be needed for processes that will be newly swapped in.	Pages
40	vmTotalReadChars	The number of characters which have been read from TTY devices since the last boot.	Characters
41	vmTotalWriteChars	The number of characters which have been written to TTY devices since the last boot.	Characters
42	vmTotalForks	The number of forks since the last boot.	--
43	vmTotalForkPages	The number of pages that have been forked since the last boot.	Pages
44	vmTotalDiskBlockReads	The number of disk block reads which have been issued since the last boot.	--
45	vmTotalDiskBlockWrites	The number of disk block writes which have been issued since the last boot.	--
46	vmTotalProcessSwapOuts	The total number of times for which system have swapped out the executable processes. This value is updated once a second.	--
47	vmTotalSwapOutProcesses	The total number of executable processes which have been swapped out by system. This value is updated once a second.	--

(4) process group

The following table describes the Hitachi enterprise-specific MIB objects in the `process` group.

Table 4–32: process group (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.process)
(1.116.5.1.2.1.3)

ID	Object name	Contents	Units
1	<code>processNum</code>	The number of processes which are running.	--
2	<code>processTable</code>	Information of processes which are running.	--
2.1	<code>processEntry</code>	Each entry is distinguished from process ID.	--
2.1.1	<code>processID</code>	Process ID.	--
2.1.2	<code>processIndex</code>	The number associated with the process.	--
2.1.3	<code>processStatus</code>	The process status. sleep (1), run (2), stop (3), zombie (4), other (5), idle (6)	--
2.1.4	<code>processStateFlags</code>	The flags associated with process. incore (1), sys (2), locked (4), trace (8), trace2 (16)	--
2.1.5	<code>processUserID</code>	The process user ID.	--
2.1.6	<code>processSavedUserID</code>	The saved process user ID.	--
2.1.7	<code>processParentID</code>	The parent process ID.	--
2.1.8	<code>processGroupID</code>	The process group ID.	--
2.1.9	<code>processCPUUtilization</code>	The CPU utilization for scheduling of process.	--
2.1.10	<code>processPriority</code>	The process priority.	--
2.1.11	<code>processCPUNice</code>	The process CPU nice.	--
2.1.12	<code>processProcessor</code>	The processor on which this process last run.	--
2.1.13	<code>processStartTime</code>	The time since the process started.	Hundredths of a second
2.1.14	<code>processPhysicalMemoryTextSize</code>	The number of bytes for physical memory which are using for text.	Bytes
2.1.15	<code>processPhysicalMemoryDataSize</code>	The number of bytes for physical memory which are using for data.	Bytes
2.1.16	<code>processPhysicalMemoryStackSize</code>	The number of bytes for physical memory which are using for stack.	Bytes
2.1.17	<code>processPhysicalMemorySharedMemorySize</code>	The number of bytes for physical memory which are using for shared memory.	Bytes
2.1.18	<code>processPhysicalMemoryMemoryMappedSize</code>	The number of bytes for physical memory which are using for memory mapped file.	Bytes
2.1.19	<code>processPhysicalMemoryUserSize</code>	The number of bytes for physical memory which are using for process data (u area).	Bytes
2.1.20	<code>processPhysicalMemoryIOSize</code>	The number of bytes for physical memory which are using for I/O device mapping.	Bytes
2.1.21	<code>processVirtualMemoryTextSize</code>	The number of bytes for virtual memory which are using for text.	Bytes

ID	Object name	Contents	Units
2.1.22	processVirtualMemoryDataSize	The number of bytes for virtual memory which are using for data.	Bytes
2.1.23	processVirtualMemoryStackSize	The number of bytes for virtual memory which are using for stack.	Bytes
2.1.24	processVirtualMemorySharedMemorySize	The number of bytes for virtual memory which are using for shared memory.	Bytes
2.1.25	processVirtualMemoryMemoryMappedSize	The number of bytes for virtual memory which are using for memory mapped file.	Bytes
2.1.26	processVirtualMemoryUserSize	The number of bytes for virtual memory which are using for process data (u area).	Bytes
2.1.27	processVirtualMemoryIOSize	The number of bytes for virtual memory which are using for I/O device mapping.	Bytes
2.1.28	processResidentSize	The number of bytes that are resident in the memory being used by the process, without being paged out to a disk.	Bytes
2.1.29	processAddress	The user area address of process. When this process is being in physical memory, this value is a physical address in main storage, and when this process is being swapped out, this value is an address on disk.	--
2.1.30	processSleepAddress	The address for which process is sleeping. When the process is not sleeping, this value is zero.	--
2.1.31	processUserTime	The time at which process spent to run on user mode.	Hundredths of a second
2.1.32	processSystemTime	The time at which process spent to run on system mode.	Hundredths of a second
2.1.33	processTTYMajor	If the process has a control terminal, this value indicates the TTY device major number. If both major and minor numbers are -1, the process does not have a console device.	--
2.1.34	processTTYMinor	If the process has a control terminal, this value indicates the TTY device minor number. If both major and minor numbers are -1, the process does not have a console device.	--
2.1.35	processCommand	Command line characters for which process started.	--
2.1.36	processExecutable	The executable file name for the process.	--
2.1.37	processResidentTime	The residents time for scheduling this process.	Seconds
2.1.38	processCPUTimeTicks	CPU time ticks of which this process spent in present time slice.	--
2.1.39	processTotalCPUTimeTicks	CPU time ticks of which this process have spent since start.	--
2.1.40	processFssID	The group ID of the fair share scheduler to which the process belongs.	--
2.1.41	processResidentTimeCPU	The rate of CPU time ticks during processResidentTime.	%
2.1.42	processMinorFaults	The number of page reclaims.	--
2.1.43	processMajorFaults	The number of page faults to need disk access.	--
2.1.44	processSwapOuts	The number of swapouts.	--
2.1.45	ProcessSignals	The number of signals which were received by process.	--
2.1.46	processReceivedMessages	The number of messages which were received by process.	--

ID	Object name	Contents	Units
2.1.47	processSentMessages	The number of messages which were sent by process.	--
2.1.48	processMaxResidentSize	The maximum number of resident bytes which were allowed to this process.	Bytes
2.1.49	processUser	The name of the user who started this process.	--

(5) processor group

The following table describes the Hitachi enterprise-specific MIB objects in the `processor` group.

Table 4–33: processor group
 (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.pocessor)
 (1.116.5.1.2.1.4)

ID	Object name	Contents	Unit
1	processorNum	The number of entries for processor information.	--
2	processorTable	The processor information table.	--
2.1	processorEntry	Each entry is distinguished from number associated with processor.	--
2.1.1	processorIndex	The number associated with the processor.	--
2.1.2	processorFileSystemReadBytes	The total number of bytes that were read from the file system since the last boot.	Bytes
2.1.3	processorFileSystemWriteBytes	The total number of bytes that were written to the file system since the last boot.	Bytes
2.1.4	processorDiskBlockReadRequests	The total number of read requests issued to the NFS-mounted disk block since the last boot.	--
2.1.5	processorDiskBlockWriteRequests	The total number of write requests issued to the NFS-mounted disk block since the last boot.	--
2.1.6	processorNFSReadBytes	The total number of bytes read from NFS since the last boot.	Bytes
2.1.7	processorNFSWriteBytes	The total number of bytes written to NFS since the last boot.	Bytes
2.1.8	processorPhysicalReads	The total number of physical reads executed by the processor from raw devices since the last boot.	--
2.1.9	processorPhysicalWrites	The total number of physical writes executed by the processor to raw devices since the last boot.	--
2.1.10	processorRunQueues	The number of times the processor had processes waiting to run since the last boot. This value is updated once every second.	--
2.1.11	processorRunQueueProcesses	The number of processes waiting to run on the processor since the last boot. This value is updated every second.	--
2.1.12	processorSysExecs	The total number of <code>exec</code> system calls issued on the processor since the last boot.	--
2.1.13	processorSysReads	The total number of <code>read</code> system calls issued on the processor since the last boot.	--
2.1.14	processorSysWrites	The total number of <code>write</code> system calls issued on the processor since the last boot.	--

ID	Object name	Contents	Unit
2.1.15	processorSysNamis	The total number of <code>sysnamis()</code> functions issued on the processor since the last boot.	--
2.1.16	processorSysIgets	The total number of <code>sysiget()</code> functions issued on the processor since the last boot.	--
2.1.17	processorDirFileSystemReadBytes	The total number of file system bytes that were read on the processor during directory lookup since the last boot.	Bytes
2.1.18	processorSemaphoreOperations	The number of System V semaphore operations executed on the processor since the last boot.	--
2.1.19	processorMessageOperations	The number of System V message operations executed on the processor since the last boot.	--
2.1.20	processorInMUXInterruptions	The number of MUX interrupts received by the processor since the last boot.	--
2.1.21	processorOutMUXInterruptions	The number of MUX interrupts sent by the processor since the last boot.	--
2.1.22	processorTTYRawChars	The number of characters read by the processor via raw device read since the last boot.	Characters
2.1.23	processorTTYCanonChars	The number of canonical characters manipulated by the processor since the last boot.	Characters
2.1.24	processorTTYOutChars	The number of characters output by the processor since the last boot.	Characters
2.1.25	processorCPULoadAvg1	The average load on the processor in the last 1 minute x 100. The average load means the average number of processes and threads in execution status or executable status on the processor in the last 1 minute. For example, if the average load is 1, this means that there was an average of 1 process or thread in execution status or executable status in the last 1 minute, and therefore it can be assumed that the CPU was always executing a process.	--
2.1.26	processorCPULoadAvg5	The average load on the processor in the last 5 minutes x 100.	--
2.1.27	processorCPULoadAvg15	The average load on the processor in the last 15 minutes x 100.	--
2.1.28	processorUserCPUTime	The CPU time used by user processes on the processor since the last boot. In Solaris and AIX, the CPU time for the user processes after the start of SNMP Agent.	Hundredths of a second
2.1.29	processorNiceCPUTime	The CPU time for the user processes executed with a <code>nice</code> value of 21 or above on the processor since the last boot. In Solaris and AIX, the CPU time for the user processes executed with a <code>nice</code> value of 21 or above after the start of SNMP Agent.	Hundredths of a second
2.1.30	processorSysCPUTime	The CPU time for the user processes executed in the system (kernel) mode on the processor since the last boot. In Solaris and AIX, the CPU time for the user processes executed in the system (kernel) mode after the start of SNMP Agent.	Hundredths of a second
2.1.31	processorIdleCPUTime	The CPU time for the idle mode on the processor since the last boot. In Solaris and AIX, the CPU idle time after the start of SNMP Agent.	Hundredths of a second
2.1.32	processorWaitCPUTime	The CPU time for the wait mode on the processor since the last boot. In Solaris and AIX, the CPU wait time after the start of SNMP Agent.	Hundredths of a second

ID	Object name	Contents	Unit
2.1.33	processorBlockCPUtime	The CPU time blocked on a spinlock on this processor since the last boot.	Hundredths of a second
2.1.34	processorSwaitCPUtime	The CPU time blocked on the kernel semaphore on this processor since the last boot.	Hundredths of a second
2.1.35	processorIntrCPUtime	The CPU time for the interrupt mode on this processor since the last boot.	Hundredths of a second
2.1.36	processorSsysCPUtime	The CPU time used by kernel process on the kernel mode on this processor since the last boot.	Hundredths of a second

(6) diskInfo group

The following table describes the Hitachi enterprise-specific MIB objects in the `diskInfo` group.

Table 4–34: diskInfo group (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskInfo) (1.116.5.1.2.1.5)

ID	Object name	Contents	Units
1	diskNum	The number of disks which are being attached to system.	--
2	diskTable	The disk information table.	--
2.1	diskEntry	Each entry is distinguished from numbers associated with disk.	--
2.1.1	diskIndex	The number associated with the disk.	--
2.1.2	diskTTYMajor	The TTY device major number for the disk.	--
2.1.3	diskTTYMinor	The TTY device minor number for the disk.	--
2.1.4	diskBusyTimeTicks	The number of time ticks for device busy.	--
2.1.5	diskSeeks	The number of seeks.	--
2.1.6	diskXfers	The number of data transfers.	--
2.1.7	diskWordsXfers	The number of transfers for double bytes word data.	--
2.1.8	diskWordsWriteTime	The time for writing double bytes word data to disk.	Milliseconds/byte

(7) swapInfo group

The following table describes the Hitachi enterprise-specific MIB objects in the `swapInfo` group.

Table 4–35: swapInfo group (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.swapInfo) (1.116.5.1.2.1.6)

ID	Object name	Contents	Units
1	swapTotalSize	The total number of swap space bytes.	Bytes
2	swapTotalEnabledSize	The total number of swap space bytes enabled to use.	Bytes
3	swapTotalFreeSize	The total number of swap space free bytes.	Bytes
4	swapTotalBlockDeviceSize	The total number of swap space bytes allocated to block device.	Bytes

ID	Object name	Contents	Units
5	swapTotalBlockDeviceEnabledSize	The total number of swap space bytes enabled to use allocated to block device.	Bytes
6	swapTotalBlockDeviceFreeSize	The total number of swap space free bytes allocated to block device.	Bytes
7	swapTotalFileSystemSize	The total number of swap space bytes allocated to file system.	Bytes
8	swapTotalFileSystemEnabledSize	The total number of swap space bytes enabled to use allocated to file system.	Bytes
9	swapTotalFileSystemFreeSize	The total number of swap space free bytes allocated to file system.	Bytes
10	swapNum	The number of entries for swap space information.	--
11	swapTable	The swap space information table.	--
11.1	swapEntry	Each entry is distinguished from numbers associated with swap space.	--
11.1.1	swapIndex	The number associated with swap space.	--
11.1.2	swapPlace	The flag that indicates where swap space is being allocated. When swap space is allocated in a block device, this value is <code>swblock</code> (1); when swap space is allocated in a file system, this value is <code>swfs</code> (2).	--
11.1.3	swapFlags	When the swap space is enabled, this value is <code>enable</code> (1), when the swap space is disabled, this value is <code>disable</code> (0).	--
11.1.4	swapPriority	The priority for using swap space. Small space is used first.	--
11.1.5	swapFreeSize	The number of swap space bytes enabled to use.	Bytes
11.1.6	swapBlockDeviceMajor	Device information, only when swap space was allocated on block device fields.	--
11.1.7	swapBlockDeviceMinor	Device information, only when swap space was allocated on block device fields.	--
11.1.8	swapBlockDeviceStartNum	The block number for starting to use, only when swap space was allocated on block device fields.	--
11.1.9	swapBlockDeviceSize	The number of swap space bytes, if swap space was allocated on a block device.	Bytes
11.1.10	swapFileSystemSize	The number of swap space bytes, if swap space was allocated on a file system.	Bytes
11.1.11	swapFileSystemMinSize	The minimum number of swap space bytes that are allocated, if swap space was allocated on a file system.	Bytes
11.1.12	swapFileSystemMaxSize	The maximum number of swap space bytes which had been allocated until now, only when swap space was allocated in file system fields.	Bytes
11.1.13	swapFileSystemReservedSize	The number of swap space bytes that are currently reserved, if swap space was allocated on a file system.	Bytes
11.1.14	swapFileSystemMountPoint	The point at which a file system is mounted, if swap space was allocated on a file system.	--

(8) swapSpace group

The following table describes the Hitachi enterprise-specific MIB objects in the `swapSpace` group:

Table 4–36: swapSpace group
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.swapSpace)
(1.116.5.1.2.1.7)

ID	Object name	Contents	Units
1	swapSpaceConfig	Swap space size	Kilobytes
2	swapSpaceEnable	Available swap space size	Kilobytes
3	swapSpaceFree	Free swap space size	Kilobytes

(9) diskBusy group

The following table describes the Hitachi enterprise-specific MIB objects in the `diskBusy` group.

Table 4–37: diskBusy group (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskBusy)
(1.116.5.1.2.1.8)

ID	Object name	Contents	Units
1	diskBusyNum	Number of diskBusyTable entries.	--
2	diskBusyTable	The disk busy ratio table.	--
2.1	diskBusyEntry	Each entry is identified by diskBusyDeviceName.	--
2.1.1	diskBusyDeviceName	Disk device name. If SNMP Agent is installed on Solaris, one entry is created for each disk, not for a partition or a disk slice, which is a kind of segment. Entry examples are <code>dad0</code> and <code>sd3</code> . In this example, <code>dad</code> represents a built-in IDE disk and <code>sd</code> represents a SCSI disk. The numbers that follow are target IDs.	--
2.1.2	diskBusyUtil	Disk busy ratio for the past 5 seconds. <ul style="list-style-type: none"> Disk busy ratio The kernel passes I/O requests from application programs to the pertinent device driver. The device driver sends the received I/O request to the pertinent device. The I/O request is retained in a queue buffer provided within the device. As specified by the I/O request in the queue, the device seeks and actually reads data, and returns it via the reverse route. The disk busy ratio is the time spent by the device performing such processing divided by the total time elapsed. The disk busy ratio represents the disk busy status that exists as the disk busy ratio information is collected. You can determine the disk status by monitoring this disk busy ratio. For example, if you observe the same disk busy ratio a number of times in succession, you can ascertain that the disk remains in the same condition. Note on obtaining this object If SNMP Agent is installed on Solaris, it needs 6 seconds or more to obtain a diskBusyTable instance. Therefore, specify 6 seconds or longer as the SNMP request time-out threshold on the manager system. 	%

(10) swapSystem group

The following table describes the Hitachi enterprise-specific MIB objects in the `swapSystem` group.

Table 4–38: swapSystem group
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.swapSystem)
(1.116.5.1.2.1.9)

ID	Object name	Contents	Units
1	swapSystemTotalSize	Total swap area size	Kilobytes
2	swapSystemTotalEnableSize	Enabled swap area size	Kilobytes
3	swapSystemTotalFreeSize	Total free swap area size	Kilobytes
4	swapSystemTotalBlockDeviceSize	Total size of the swap area located in the block-based device	Kilobytes
5	swapSystemTotalBlockDeviceEnableSize	Total size of the enabled swap area located in the block-based device	Kilobytes
6	swapSystemTotalBlockDeviceFreeSize	Total size of the free swap area located in the block-based device	Kilobytes
7	swapSystemTotalFileSystemSize	Total size of the swap area located in the file system	Kilobytes
8	swapSystemTotalFileSystemEnabledSize	Total size of the enabled swap area located in the file system	Kilobytes
9	swapSystemTotalFileSystemFreeSize	Total size of the free swap area located in the file system	Kilobytes
10	swapSystemNum	Number of swap area information entries	--
11	swapSystemTable	The swap area information table	--
11.1	swapSystemEntry	Each entry is identified by the swapSystemIndex value.	--
11.1.1	swapSystemIndex	Swap area index number	--
11.1.2	swapSystemPlace	Flag that indicates where the swap area resides swblock(1): In block-based device swfs(2): In file system	--
11.1.3	swapSystemFlags	Indicates whether the swap area is enabled or disabled disable(0): Disabled enable(1): Enabled	--
11.1.4	swapSystemPriority	Swap area priority. A swap area assigned a lower value will be used sooner.	--
11.1.5	swapSystemFreeSize	Size of the free area included in the currently enabled swap area	Kilobytes
11.1.6	swapSystemBlockDeviceMajor	Contains device information (Major) if the swap area resides in a block-based device.	--
11.1.7	swapSystemBlockDeviceMinor	Contains device information (Minor) if the swap area resides in a block-based device.	--
11.1.8	swapSystemBlockDeviceStartNum	Contains the starting block number if the swap area resides in a block-based device.	--
11.1.9	swapSystemBlockDeviceSize	Swap area size if the swap area is allocated in a block-based device	Kilobytes
11.1.10	swapSystemFileSystemSize	Swap area size if the swap area is allocated in a file system	Kilobytes
11.1.11	swapSystemFileSystemMinSize	Minimum size of the area that can be allocated as the swap area if it resides in a file system	Kilobytes

ID	Object name	Contents	Units
11.1.12	swapSystemFileSystemMaxSize	Maximum size of the area that can be allocated as the swap area if it resides in a file system	Kilobytes
11.1.13	swapSystemFileSystemReservedSize	Size of the area currently reserved on the swap area if it resides in a file system	Kilobytes
11.1.14	swapSystemFileSystemMountPoint	If the swap area resides in a file system, this object indicates where the file system is mounted.	--

(11) cpuUtil group

The following table describes the Hitachi enterprise-specific MIB objects in the `cpuUtil` group.

Table 4–39: `cpuUtil` group (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.cpuUtil) (1.116.5.1.2.1.10)

ID	Object name	Contents	Units
1	cpuUtilTable	CPU utilization rate information	--
1.1	cpuUtilEntry	Each entry is distinguished by its <code>cpuUtilNum</code> value. SNMP Agent acquires the CPU utilization rate during the specified interval time (default is 5 minutes) from the OS. This value becomes the MIB value. Acquisition of the CPU utilization rate is continuously executed. The utilization rate of each CPU can be monitored by periodically collecting this MIB from the manager.	--
1.1.1	cpuUtilNum	CPU number (0 if there is only 1 CPU)	--
1.1.2	cpuUtilUser	User CPU utilization during the specified interval	%
1.1.3	cpuUtilSystem	System CPU utilization during the specified interval	%
1.1.4	cpuUtilWio	Wait CPU utilization during the specified interval	%
1.1.5	cpuUtilIdle	Idle CPU utilization during the specified interval	%
1.1.6	cpuUtilTime	The time when CPU utilization was acquired from the OS (for example, 2003/01/16 19:00:00)	--
2	cpuUtilInterval	Interval time	Minutes
3	cpuUtilTotalUser	Average user CPU utilization rate for all CPUs during the specified interval	%
4	cpuUtilTotalSystem	Average system CPU utilization rate for all CPUs during the specified interval	%
5	cpuUtilTotalWio	Average wait CPU utilization rate for all CPUs during the specified interval	%
6	cpuUtilTotalIdle	Average idle CPU utilization rate for all CPUs during the specified interval	%

Important

Note the following about the Hitachi enterprise-specific MIB objects in the `cpuUtil` group:

- You can use the `htc_monagt1` option to change the interval time (minutes).
- You can set up SNMP Agent such that it does not obtain MIB values.

- For `cpuUtilTotalUser`, `cpuUtilTotalSystem`, `cpuUtilTotalWio`, and `cpuUtilTotalIdle`, the utilization rates of individual CPUs are added and then divided by the number of CPUs. Since decimals are discarded from the resulting numbers, the total might not equal 100%.

In the SMT environment, you can use the `SNMP_HTC_AIX_CPU_SMT` environment variable to specify the CPU utilization rate for the entire machine. When you use a resource browser to obtain the CPU utilization rate from JP1/SSO, CPU information is displayed as a single instance for the entire machine even if multiple CPUs are installed in the actual environment.

- If a CPU is disabled or moved, the correct values might be temporarily unobtainable. Wait a little while and the correct values will become obtainable again.
- For notes about acquiring CPU information when the online/offline status of the CPU changes in Solaris, see [2.15 Notes about CPU information](#).
- For notes about cases in which a CPU is dynamically added or removed by DLPAR (Dynamic Logical Partition) in AIX, see [2.15 Notes about CPU information](#).

(12) virtualMemory64 group

The following table describes the Hitachi enterprise-specific MIB objects in the `virtualMemory64` group.

Table 4–40: `virtualMemory64` group
(`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.virtualMemory64`)
(1.116.5.1.2.1.13)

ID	Object name	Contents	Units
1	<code>vm64PageSize</code>	The number of bytes per page for virtual storage.	Bytes
2	<code>vm64DaemonfreePages</code>	The number of pages which were freed by the pageout daemon. This value is the average for one second and is updated every 5 seconds.	Pages
3	<code>vm64Interruptions</code>	The number of device interruptions. This value is the average for one second and is updated every 5 seconds.	--
4	<code>vm64PageInPages</code>	The number of pages which were paged in. This value is the average for one second and is updated every 5 seconds.	Pages
5	<code>vm64PageOutPages</code>	The number of pages which were paged out. This value is the average for one second and is updated every 5 seconds.	Pages
6	<code>vm64PageReclaims</code>	The total number of page reclaims. This value is the average for one second and is updated every 5 seconds.	--
7	<code>vm64TLBFlashes</code>	The number of times a TLB flush occurred. This value is the average for one second and is updated every 5 seconds.	--
8	<code>vm64DaemonScanPages</code>	The number of pages which were scanned by the pageout daemon. This value is the average for one second and is updated every 5 seconds.	Pages
9	<code>vm64ContextSwitches</code>	The number of context switches. This value is the average for one second and is updated every 5 seconds.	--
10	<code>vm64SystemCalls</code>	The number of calls to system call. This value is the average for one second and is updated every 5 seconds.	--

ID	Object name	Contents	Units
11	vm64XFileSystemFreelistPages	The number of pages that were found in free list rather than in file system. This value is the average for one second and is updated every 5 seconds.	Pages
12	vm64XSwapDeviceFreeListPages	The number of pages that were found in freelist rather than on swapdevice. This value is the average for one second and is updated every 5 seconds.	Pages
13	vm64FreeMemoryPages	The number of free memory pages. This value is the average for one second and is updated every 5 seconds.	Pages
14	vm64TotalSwapIns	The number of swapins since the last boot.	--
15	vm64TotalSwapOuts	The number of swapouts since the last boot.	--
16	vm64TotalDaemonFreePages	The number of pages that have been freed by the pageout daemon since the last boot.	Pages
17	vm64TotalDemandLoadPages	The number of pages that have been filled on demand from the executable file since the last boot.	Pages
18	vm64TotalPageFaults	The number of page faults since the last boot.	--
19	vm64TotalInterruptions	The number of device interruptions since the last boot.	--
20	vm64TotalIntransitPageFaults	The number of intransit blocking page faults since the last boot.	--
21	vm64TotalDemandLoadCreatePages	The number of created pages that have been filled on demand since the last boot.	Pages
22	vm64TotalZeroFillCreatePages	The number of new zero-filled pages created since the last boot.	Pages
23	vm64TotalFreeListReclaimedPages	The number of pages reclaimed from freelist since the last boot.	Pages
24	vm64TotalPageIns	The number of pageins since the last boot.	--
25	vm64TotalPageOuts	The number of pageouts since the last boot.	--
26	vm64TotalPageInPages	The number of pages that have been paged in since the last boot.	Pages
27	vm64TotalPageOutPages	The number of pages that have been paged out since the last boot.	Pages
28	vm64TotalSwapInPages	The number of pages that have been swapped in since the last boot.	Pages
29	vm64TotalSwapOutPages	The number of pages that have been swapped out since the last boot.	Pages
30	vm64TotalDaemonTicksNum	The number of pageouts by the pageout daemon since the last boot.	--
31	vm64TotalContextSwitches	The number of context switches since the last boot.	--
32	vm64TotalSystemCalls	The number of system calls since the last boot.	--
33	vm64TotalTraps	The number of traps since the last boot.	--
34	vm64TotalXFileSystemFreeListPages	The number of pages that were found in freelist rather than in the file system since the last boot.	Pages
35	vm64TotalXSwapDeviceFreeListPages	The number of pages that were found in freelist rather than on a swap device since the last boot.	Pages
36	vm64TotalDemandZeroFillPages	The number of pages that have been zero filled on demand since the last boot.	Pages
37	vm64TotalDaemonScanPages	The number of pages that have been scanned by the pageout daemon since the last boot.	Pages

ID	Object name	Contents	Units
38	vm64TotalReclaimedPages	The number of page reclaims since the last boot.	Pages
39	vm64TotalDeficitPages	The number of pages that are estimated to be needed for processes that will be newly swapped in.	Pages
40	vm64TotalReadChars	The number of characters that have been read from TTY devices since the last boot.	Characters
41	vm64TotalWriteChars	The number of characters that have been written to TTY devices since the last boot.	Characters
42	vm64TotalForks	The number of forks since the last boot.	--
43	vm64TotalForkPages	The number of pages that have been forked since the last boot.	Pages
44	vm64TotalDiskBlockReads	The number of disk block reads that have been issued since the last boot.	--
45	vm64TotalDiskBlockWrites	The number of disk block writes that have been issued since the last boot.	--
46	vm64TotalProcessSwapOuts	The total number of times the system swapped out executable processes. This value is updated once a second.	--
47	vm64TotalSwapOutProcesses	The total number of executable processes that have been swapped out by the system. This value is updated once a second.	--

Important

Note the following about the Hitachi enterprise-specific MIB objects in the `virtualMemory64` group:

- The syntax of objects from 14 to 47 in this group is Counter64. Objects with Counter64 syntax can only be acquired through SNMPv2c requests, not SNMPv1 requests.

(13) systemInfo64 group

The following table describes the Hitachi enterprise-specific MIB objects in the `systemInfo64` group.

Table 4–41: `systemInfo64` group
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.systemInfo64)
(1.116.5.1.2.1.14)

ID	Object name	Contents	Units
1	systemInfo64TTYMajor	The major number of the console device. If the value is $(2^{31} - 1)$ or greater, this object contains $(2^{31} - 1)$.	--
2	systemInfo64TTYMinor	The minor number of the console device. If the value is $(2^{31} - 1)$ or greater, this object contains $(2^{31} - 1)$.	--
3	systemInfo64BootTime	The time since the last boot.	Hundredths of a second
4	systemInfo64ActiveProcessors	The number of processors which are running when the system call is issued.	--
5	systemInfo64MaxProcessors	The maximum number of processors which ran at the same time since the last boot.	--

ID	Object name	Contents	Units
6	systemInfo64MaxProcesses	The maximum number of processes which were allowed to run on the system.	--
7	systemInfo64RunQueProcesses	The number of processes (except the system process) which are loaded to main storage and are waiting to run.	--
8	systemInfo64XferWaitProcesses	The number of processes which are waiting to be transferred from the disk.	--
9	systemInfo64PageInWaitProcesses	The number of processes which are waiting to be paged in from the disk.	--
10	systemInfo64SleepProcesses	The number of sleeping processes in physical memory. However, the following processes are excluded: <ul style="list-style-type: none"> • Processes that are sleeping while waiting for I/O • Processes that have slept continuously for at least 20 seconds 	--
11	systemInfo64SwapOutProcesses	The number of processes that have been swapped out and that immediately begin to wait for execution when they are swapped in.	--
12	systemInfo64PhysicalMemorySize	The maximum size of the physical memory installed in the system.	Bytes
13	systemInfo64PhysicalMemoryFreeSize	The size of free physical memory that the system assumed to be allocatable for processes.	Bytes
14	systemInfo64VirtualMemoryProcessSize	The size of virtual memory allocated for text, data and stack areas by all processes other than the system process.	Bytes
15	systemInfo64VirtualMemoryWaitProcessSize	The size of virtual memory allocated for all the processes that are waiting for execution.	Bytes
16	systemInfo64PhysicalMemoryProcessSize	The size of physical memory allocated for text, data, and stack areas by all the processes in the system.	Bytes
17	systemInfo64PhysicalMemoryWaitProcessSize	The size of physical memory allocated for all the processes that are waiting for execution.	Bytes
18	systemInfo64CPUStates	The number of CPU states that the system has.	--
19	systemInfo64OpenLogicalVolumes	The number of open logical volumes.	--
20	systemInfo64OpenLogicalVolumeGroups	The number of open logical volume groups.	--
21	systemInfo64AllocPBUFs	The number of PBUFs allocated for logical volumes.	--
22	systemInfo64UsedPBUFs	The number of PBUFs used for logical volumes.	--
23	systemInfo64MaxPBUFs	The maximum number of PBUFs which have been used for logical volumes until now.	--
24	systemInfo64ActiveProcessEntries	The number of process table entries that are currently in use.	Entries
25	systemInfo64ActiveInodeEntries	The number of inode table entries that are currently in use.	Entries
26	systemInfo64ActiveFileEntries	The number of file table entries that are currently in use.	Entries

Important

Note the following about the Hitachi enterprise-specific MIB objects in the `systemInfo64` group:

- The syntax of objects from 3 to 26 in this group is `CounterBasedGauge64`. Objects with `CounterBasedGauge64` syntax can only be acquired through SNMPv2c requests, not SNMPv1 requests.

(14) virtualMemory64Ex group

The following table describes the Hitachi enterprise-specific MIB objects in the `virtualMemory64Ex` group.

Table 4–42: virtualMemory64Ex group
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.virtualMemory64Ex)
(1.116.5.1.2.1.15)

ID	Object name	Contents	Units
1	<code>vm64ExPageSize</code>	The number of bytes per page in a virtual storage system.	Bytes
2	<code>vm64ExDaemonfreePages</code>	The number of pages that were freed by the pageout daemon. This value is the average per second, and is updated every five seconds.	Pages
3	<code>vm64ExInterruptions</code>	The number of interrupts from devices. This value is the average per second, and is updated every five seconds.	--
4	<code>vm64ExPageInPages</code>	The number of pages that were paged in. This value is the average per second, and is updated every five seconds.	Pages
5	<code>vm64ExPageOutPages</code>	The number of pages that were paged out. This value is the average per second, and is updated every five seconds.	Pages
6	<code>vm64ExPageReclaims</code>	The total number of page reclaims. This value is the average per second, and is updated every five seconds.	--
7	<code>vm64ExTLBFlashes</code>	The number of times a TLB flush occurred. This value is the average per second, and is updated every five seconds.	--
8	<code>vm64ExDaemonScanPages</code>	The number of pages that were scanned by the pageout daemon. This value is the average per second, and is updated every five seconds.	Pages
9	<code>vm64ExContextSwitches</code>	The number of times a context switch occurred. This value is the average per second, and is updated every five seconds.	--
10	<code>vm64ExSystemCalls</code>	The number of issued system calls. This value is the average per second, and is updated every five seconds.	--
11	<code>vm64ExXFileSystemFreelistPages</code>	The number of pages found in the page free list, and not in file systems. This value is the average per second, and is updated every five seconds.	Pages
12	<code>vm64ExXSwapDeviceFreeListPages</code>	The number of pages found in the page free list, and not in the swap device. This value is the average per second, and is updated every five seconds.	Pages
13	<code>vm64ExFreeMemoryPages</code>	The number of free memory pages. This value is the average per second, and is updated every five seconds.	Pages
14	<code>vm64ExTotalSwapIns</code>	The number of times a swap-in occurred after the last boot.	--
15	<code>vm64ExTotalSwapOuts</code>	The number of times a swap-out occurred after the last boot.	--
16	<code>vm64ExTotalDaemonFreePages</code>	The number of pages freed by the pageout daemon after the last boot.	Pages
17	<code>vm64ExTotalDemandLoadPages</code>	The number of pages loaded to executable files on demand after the last boot.	Pages
18	<code>vm64ExTotalPageFaults</code>	The number of times a page fault occurred after the last boot.	--
19	<code>vm64ExTotalInterruptions</code>	The number of times a device interrupt occurred after the last boot.	--
20	<code>vm64ExTotalIntransitPageFaults</code>	The number of times an intransit block page fault occurred after the last boot.	--

ID	Object name	Contents	Units
21	vm64ExTotalDemandLoadCreatePages	The number of new pages created as on-demand load pages after the last boot.	Pages
22	vm64ExTotalZeroFillCreatePages	The number of new zero-filled pages created since the last boot.	Pages
23	vm64ExTotalFreeListReclaimedPages	The number of pages reclaimed from freelist since the last boot.	Pages
24	vm64ExTotalPageIns	The number of times a page-in occurred after the last boot.	--
25	vm64ExTotalPageOuts	The number of times a page-out occurred after the last boot.	--
26	vm64ExTotalPageInPages	The number of pages that were paged in after the last boot.	Pages
27	vm64ExTotalPageOutPages	The number of pages that were paged out after the last boot.	Pages
28	vm64ExTotalSwapInPages	The number of pages that were swapped in after the last boot.	Pages
29	vm64ExTotalSwapOutPages	The number of pages that were swapped out after the last boot.	Pages
30	vm64ExTotalDaemonTicksNum	The number of pageouts by the pageout daemon since the last boot.	--
31	vm64ExTotalContextSwitches	The number of times a context switch occurred after the last boot.	--
32	vm64ExTotalSystemCalls	The number of times a system call was issued after the last boot.	--
33	vm64ExTotalTraps	The number of times a trap occurred after the last boot.	--
34	vm64ExTotalXFileSystemFreeListPages	The number of pages that were found in the free list, and not in file systems, after the last boot.	Pages
35	vm64ExTotalXSwapDeviceFreeListPages	The number of pages that were found in the free list, and not in the swap device, after the last boot.	Pages
36	vm64ExTotalDemandZeroFillPages	The number of pages that were zero-filled on demand after the last boot.	Pages
37	vm64ExTotalDaemonScanPages	The number of pages that were scanned by the pageout daemon after the last boot.	Pages
38	vm64ExTotalReclaimedPages	The number of pages reclaimed after the last boot.	Pages
39	vm64ExTotalDeficitPages	The number of pages that are likely needed for the new process that is to be swapped in.	Pages
40	vm64ExTotalReadChars	The number of characters that were read from the TTY device after the last boot.	Characters
41	vm64ExTotalWriteChars	The number of characters that were written to the TTY device after the last boot.	Characters
42	vm64ExTotalForks	The number of times a fork occurred after the last boot.	--
43	vm64ExTotalForkPages	The number of pages that forked after the last boot.	Pages
44	vm64ExTotalDiskBlockReads	The number of times that a disk-block read request was issued after the last boot.	--
45	vm64ExTotalDiskBlockWrites	The number of times that a disk-block write request was issued after the last boot.	--
46	vm64ExTotalProcessSwapOuts	The number of times that a process that can be executed by the system was swapped out. This value is updated once a second.	--
47	vm64ExTotalSwapOutProcesses	The number of processes that were swapped out by the system. This value is updated once a second.	--

! Important

Note the following about the Hitachi enterprise-specific MIB objects in the `virtualMemory64Ex` group.

- The syntax of objects from 1 to 13 in this group is `CounterBasedGauge64`. The syntax of objects from 14 to 47 in this group is `Counter64`. Objects with `CounterBasedGauge64` or `Counter64` syntax can only be acquired through SNMPv2c requests, not SNMPv1 requests.

(15) process64 group

The following table describes the Hitachi enterprise-specific MIB objects in the `process64` group.

Table 4–43: `process64` group
(`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.process64`)
(1.116.5.1.2.1.16)

ID	Object name	Contents	Units
1	<code>process64Num</code>	The number of running processes.	--
2	<code>process64Table</code>	The table containing information about running processes.	--
2.1	<code>process64Entry</code>	Each entry is identified by a process ID.	--
2.1.1	<code>process64ID</code>	The process ID.	--
2.1.2	<code>process64Index</code>	The index number of a process.	--
2.1.3	<code>process64Status</code>	The process status: sleep (1), run (2), stop (3), zombie (4), other (5), idle (6)	--
2.1.4	<code>process64StatusFlags</code>	The flags associated with a process: incore (1), sys (2), locked (4), trace (8), trace2 (16)	--
2.1.5	<code>process64UserID</code>	The process user ID.	--
2.1.6	<code>process64SavedUserID</code>	The saved process user ID.	--
2.1.7	<code>process64ParentID</code>	The parent process ID.	--
2.1.8	<code>process64GroupID</code>	The process group ID.	--
2.1.9	<code>process64CPUUtilization</code>	CPU utilization of a process.	--
2.1.10	<code>process64Priority</code>	The process priority.	--
2.1.11	<code>process64CPUNice</code>	The process CPU nice.	--
2.1.12	<code>process64Processor</code>	The processor on which this process last ran.	--
2.1.13	<code>process64StartTime</code>	The time elapsed since the process started.	Hundredths of a second
2.1.14	<code>process64PhysicalMemoryTextSize</code>	The size of physical memory allocated for text area.	Bytes
2.1.15	<code>process64PhysicalMemoryDataSize</code>	The size of physical memory allocated for data area.	Bytes
2.1.16	<code>process64PhysicalMemoryStackSize</code>	The size of physical memory allocated for stack area.	Bytes
2.1.17	<code>process64PhysicalMemorySharedMemorySize</code>	The size of physical memory allocated as shared memory.	Bytes
2.1.18	<code>process64PhysicalMemoryMemoryMappedSize</code>	The size of physical memory allocated for memory mapped files.	Bytes

ID	Object name	Contents	Units
2.1.19	process64PhysicalMemoryUserSize	The size of physical memory allocated for process data (u area).	Bytes
2.1.20	process64PhysicalMemoryIOSize	The size of physical memory allocated for I/O space.	Bytes
2.1.21	process64VirtualMemoryTextSize	The size of virtual addressing space allocated for text area.	Bytes
2.1.22	process64VirtualMemoryDataSize	The size of virtual addressing space allocated for data area.	Bytes
2.1.23	process64VirtualMemoryStackSize	The size of virtual addressing space allocated for stack area.	Bytes
2.1.24	process64VirtualMemorySharedMemorySize	The size of virtual addressing space allocated as shared memory.	Bytes
2.1.25	process64VirtualMemoryMemoryMappedSize	The size of virtual addressing space allocated for memory mapped files.	Bytes
2.1.26	process64VirtualMemoryUserSize	The size of virtual addressing space allocated for process data (u area).	Bytes
2.1.27	process64VirtualMemoryIOSize	The size of virtual addressing space allocated for I/O space.	Bytes
2.1.28	process64ResidentSize	The number of bytes that are resident in the memory being used by the process, without being paged out to a disk.	Bytes
2.1.29	process64Address	The user area address for the process. When the process has been loaded to memory, this object indicates the physical address in the physical memory. When the process has been swapped out, this object indicates the address on the disk.	--
2.1.30	process64SleepAddress	The address at which the process is sleeping. When the process is not sleeping, this object is set to 0.	--
2.1.31	process64UserTime	The time that has elapsed since the process started to run in the user mode.	Hundredths of a second
2.1.32	process64SystemTime	The time that has elapsed since the process started to run in the system mode.	Hundredths of a second
2.1.33	process64TTYMajor	If the process has a control terminal, this value indicates the TTY device major number. If both major and minor numbers are -1, the process does not have a console device. If this value is $(2^{31}-1)$ or greater, this object contains $(2^{31}-1)$.	--
2.1.34	process64TTYMinor	If the process has a control terminal, this value indicates the TTY device minor number. If both major and minor numbers are -1, the process does not have a console device. If this value is $(2^{31}-1)$ or greater, this object contains $(2^{31}-1)$.	--
2.1.35	process64Command	The command line string with which the process was started.	--
2.1.36	process64Executable	The name of the executable file for the process.	--
2.1.37	process64ResidentTime	The in-core elapsed time.	Seconds
2.1.38	process64CPUTimeTicks	The number of CPU ticks that the process spent during the current time slice.	--
2.1.39	process64TotalCPUTimeTicks	The number of CPU ticks that the process has spent since the process was generated.	--
2.1.40	process64FssID	The group ID of the fair share scheduler to which the process belongs.	--
2.1.41	process64ResidentTimeCPU	The ratio of the CPU time to the in-core elapsed time.	%
2.1.42	process64MinorFaults	The number of times this process has performed page reclaim.	--

ID	Object name	Contents	Units
2.1.43	process64MajorFaults	The number of page faults that this process caused and that required a disk access.	--
2.1.44	process64SwapOuts	The number of times this process was swapped out.	--
2.1.45	process64Signals	The number of signals that this process received.	--
2.1.46	process64ReceivedMessages	The number of messages that this process received.	--
2.1.47	process64SentMessages	The number of messages that the process sent.	--
2.1.48	process64MaxResidentSize	The maximum size of in-core memory that this process can acquire.	Bytes
2.1.49	process64User	The user who started this process.	--

Important

Note the following about the Hitachi enterprise-specific MIB objects in the `process64` group:

- The syntax of objects from 2.1.2, 2.1.5 to 2.1.7, 2.1.9, 2.1.11 to 2.1.32, 2.1.37, and 2.1.48 in this group is CounterBasedGauge64. The syntax of objects from 2.1.38 to 2.1.39 and 2.1.42 to 2.1.47 in this group is Counter64. Objects with CounterBasedGauge64 or Counter64 syntax can only be acquired through SNMPv2c requests, not SNMPv1 requests.

(16) processor64 group

The following table describes the Hitachi enterprise-specific MIB objects in the `processor64` group.

Table 4–44: `processor64` group
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.processor64)
(1.116.5.1.2.1.17)

ID	Object name	Contents	Units
1	processor64Num	The number of processor information entries.	--
2	processor64Table	The processor information table.	--
2.1	processor64Entry	Each entry is identified by the <code>processor64Index</code> value of the processor.	--
2.1.1	processor64Index	The processor index number.	--
2.1.2	processor64FileSystemReadBytes	The total number of bytes read from the file system since the last boot.	Bytes
2.1.3	processor64FileSystemWriteBytes	The total number of bytes written from the file system since the last boot.	Bytes
2.1.4	processor64DiskBlockReadRequests	The total number of read requests issued for the NFS-mounted disk block since the last boot.	--
2.1.5	processor64DiskBlockWriteRequests	The total number of write requests issued for the NFS-mounted disk block since the last boot.	--
2.1.6	processor64NFSReadBytes	The total number of bytes read from NFS since the last boot.	Bytes
2.1.7	processor64NFSWriteBytes	The total number of bytes written from NFS since the last boot.	Bytes

ID	Object name	Contents	Units
2.1.8	processor64PhysicalReads	The total number of physical reads performed by the processor from row devices since the last boot.	--
2.1.9	processor64PhysicalWrites	The total number of physical writes performed by the processor to row devices since the last boot.	--
2.1.10	processor64RunQueues	The number of times a process waited for execution on the processor since the last boot. This value is updated every second.	--
2.1.11	processor64RunQueueProcesses	The number of processes waiting for execution on the processor since the last boot. This value is updated every second.	--
2.1.12	processor64SysExecs	The total number of <code>exec</code> system calls issued on the processor since the last boot.	--
2.1.13	processor64SysReads	The total number of <code>read</code> system calls issued on the processor since the last boot.	--
2.1.14	processor64SysWrites	The total number of <code>write</code> system calls issued on the processor since the last boot.	--
2.1.15	processor64SysNamis	The total number of <code>sysnami()</code> functions issued on the processor since the last boot.	--
2.1.16	processor64SysIgets	The total number of <code>sysiget()</code> functions issued on the processor since the last boot.	--
2.1.17	processor64DirFileSystemReadBytes	The total number of file system bytes that were read on the processor during directory lookup since the last boot.	Bytes
2.1.18	processor64SemaphoreOperations	The number of System V semaphore operations executed on the processor since the last boot.	--
2.1.19	processor64MessageOperations	The number of System V message operations executed on the processor since the last boot	--
2.1.20	processor64InMUXInterruptions	The number of MUX interrupts received by the processor since the last boot.	--
2.1.21	processor64OutMUXInterruptions	The number of MUX interrupts sent by the processor since the last boot.	--
2.1.22	processor64TTYRawChars	The number of characters read by the processor via raw device read since the last boot.	Characters
2.1.23	processor64TTYCanonChars	The number of canonical characters manipulated by the processor since the last boot.	Characters
2.1.24	processor64TTYOutChars	The number of characters output by the processor since the last boot.	Characters
2.1.25	processor64CPULoadAvg1	The average load on the processor in the last 1 minute x 100. The average load means the average number of processes and threads in execution status or executable status on the processor in the last 1 minute. For example, if the average load is 1, the means that there was an average of 1 process or thread that was in execution status or executable status in the last 1 minute, and therefore it can be assumed that the CPU was always executing a process.	--
2.1.26	processor64CPULoadAvg5	The average load on the processor in the last 5 minutes x 100.	--
2.1.27	processor64CPULoadAvg15	The average load on the processor in the last 15 minutes x 100.	--
2.1.28	processor64UserCPUtime	The CPU time used by user processes on the processor since the last boot.	Hundredths of a second

ID	Object name	Contents	Units
2.1.29	processor64NiceCPUtime	The CPU time for the user processes executed with a <code>nice</code> value of 21 or above on the processor since the last boot.	Hundredths of a second
2.1.30	processor64SysCPUtime	The CPU time for the user processes executed in the system (kernel) mode on the processor since the last boot.	Hundredths of a second
2.1.31	processor64IdleCPUtime	The CPU time for the idle mode on the processor since the last boot.	Hundredths of a second
2.1.32	processor64WaitCPUtime	The CPU time for the wait mode on the processor since the last boot.	Hundredths of a second
2.1.33	processor64BlockCPUtime	The CPU time blocked on a spinlock on the processor since the last boot.	Hundredths of a second
2.1.34	processor64SwaitCPUtime	The CPU time blocked on the kernel semaphore on the processor since the last boot.	Hundredths of a second
2.1.35	processor64IntrCPUtime	The CPU time for the interrupt mode on the processor since the last boot.	Hundredths of a second
2.1.36	processor64SsysCPUtime	The CPU time used by the kernel process in the kernel mode on the processor since the last boot.	Hundredths of a second

Important

Note the following about the Hitachi enterprise-specific MIB objects in the `processor64` group:

- The syntax of objects from 2.1.2 to 2.1.24 and 2.1.28 to 2.1.36 is Counter64. Objects with Counter64 syntax can only be acquired through SNMPv2c requests, not SNMPv1 requests.
- For details about how to use SNMP Agent of HP-UX (IPF) to acquire the CPU information of enabled processors only, see [2.15 Notes about CPU information](#).

(17) diskInfo64 group

The following table describes the Hitachi enterprise-specific MIB objects in the `diskInfo64` group.

Table 4–45: `diskInfo64` group
(`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskInfo64`)
(1.116.5.1.2.1.18)

ID	Object name	Contents	Units
1	disk64Num	The number of disks attached to the system.	--
2	disk64Table	The disk information table.	--
2.1	disk64Entry	Each entry is identified by the <code>disk64Index</code> value.	--
2.1.1	disk64Index	The disk index number.	--
2.1.2	disk64TTYMajor	If the TTY device major number for the disk value is $(2^{31} - 1)$ or greater, this object contains $(2^{31} - 1)$.	--
2.1.3	disk64TTYMinor	If the TTY device minor number for the disk value is $(2^{31} - 1)$ or greater, this object contains $(2^{31} - 1)$.	--
2.1.4	disk64BusyTimeTicks	The number of time ticks for device busy.	--

ID	Object name	Contents	Units
2.1.5	disk64Seeks	The number of seeks.	--
2.1.6	disk64Xfers	The number of data transfers.	--
2.1.7	disk64WordsXfers	The number of times a two-byte word was transferred.	--
2.1.8	disk64WordsWriteTime	The time required to write a two-byte word.	Millisecond s/byte

Important

Note the following about the Hitachi enterprise-specific MIB objects in the `diskInfo64` group:

- The syntax of objects from 2.1.4 to 2.1.7 in this group is Counter64. Objects with Counter64 syntax can only be acquired through SNMPv2c requests, not SNMPv1 requests.

(18) swapSystem64 group

The following table describes the Hitachi enterprise-specific MIB objects in the `swapSystem64` group.

Table 4–46: `swapSystem64` group
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.swapSystem64)
(1.116.5.1.2.1.19)

ID	Object name	Contents	Units
1	swapSystem64TotalSize	The total swap area size.	Kilobytes
2	swapSystem64TotalEnabledSize	The total size of enabled swap area.	Kilobytes
3	swapSystem64TotalFreeSize	The total size of free swap area.	Kilobytes
4	swapSystem64TotalBlockDeviceSize	The total size of swap area allocated on the block device.	Kilobytes
5	swapSystem64TotalBlockDeviceEnabledSize	The total size of enabled swap area allocated on the block device.	Kilobytes
6	swapSystem64TotalBlockDeviceFreeSize	The total size of free swap area allocated on the block device.	Kilobytes
7	swapSystem64TotalFileSystemSize	The total size of swap area allocated on the file system.	Kilobytes
8	swapSystem64TotalFileSystemEnabledSize	The total size of enabled swap area allocated on the file system.	Kilobytes
9	swapSystem64TotalFileSystemFreeSize	The total size of free swap area allocated on the file system.	Kilobytes
10	swapSystem64Num	The number of entries in the information about the swap area.	--
11	swapSystem64Table	The swap area information table.	--
11.1	swapSystem64Entry	Each entry is identified by the <code>swapSystem64Index</code> value.	--
11.1.1	swapSystem64Index	The swap area index number.	--
11.1.2	swapSystem64Place	The flag that indicates where the swap area is allocated: swblock (1): On the block-based device swfs (2): On the file system	--

ID	Object name	Contents	Units
11.1.3	swapSystem64Flags	The flag that indicates whether the swap area is enabled or disabled: disable(0): Disabled enable(1): Enabled	--
11.1.4	swapSystem64Priority	The swap area priority. A swap area that is assigned a lower value will be used sooner.	--
11.1.5	swapSystem64FreeSize	The size of the free area included in the currently enabled swap area.	Kilobytes
11.1.6	swapSystem64BlockDeviceMajor	Contains device information (Major) if swap area is allocated in a block-based device. If the value is $(2^{31} - 1)$ or greater, this object contains $(2^{31} - 1)$.	--
11.1.7	swapSystem64BlockDeviceMinor	Contains device information (Minor) when swap area is allocated in a block-based device. If the value is $(2^{31} - 1)$ or greater, this object contains $(2^{31} - 1)$.	--
11.1.8	swapSystem64BlockDeviceStartNum	Contains the starting block number when swap area is allocated in a block-based device.	--
11.1.9	swapSystem64BlockDeviceSize	The size of the allocated swap area when swap area is allocated in a block-based device.	Kilobytes
11.1.10	swapSystem64FileSystemSize	The size of the allocated swap area when swap area is allocated in a file system.	Kilobytes
11.1.11	swapSystem64FileSystemMinSize	The minimum swap area size of the allocated swap area when swap area is allocated in a file system.	Kilobytes
11.1.12	swapSystem64FileSystemMaxSize	The maximum swap area size that has been allocated in a file system when swap area is allocated in a file system.	Kilobytes
11.1.13	swapSystem64FileSystemReservedSize	The size of the currently reserved swap area when swap area is allocated in a file system.	Kilobytes
11.1.14	swapSystem64FileSystemMountPoint	If the swap area resides in a file system, this object indicates where the file system is mounted.	--

Important

Note the following about the Hitachi enterprise-specific MIB objects in the `swapSystem64` group:

- The syntax of objects from 1 to 9, 11.1.4 to 1.1.5, and 11.1.8 to 11.1.13 in this group is `CounterBasedGauge64`. Objects with `CounterBasedGauge64` syntax can only be acquired through SNMPv2c requests, not SNMPv1 requests.

(19) diskBusyInfo group

The following table describes the Hitachi enterprise-specific MIB objects in the `diskBusyInfo` group.

Table 4–47: `diskBusyInfo` group
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskBusyInfo)
(1.116.5.1.2.1.20)

ID	Object name	Contents	Units
1	diskBusyInfoTable	The table for indicating the disk busy times.	--

ID	Object name	Contents	Units
1.1	diskBusyInfoEntry	Each entry is identified by <code>diskBusyInfoDeviceName</code> and <code>diskBusyInfoDeviceIndex</code> .	--
1.1.1	diskBusyInfoDeviceName	Disk device name. One entry is created for each disk, not for a partition or disk slice, which is a kind of segment.	--
1.1.2	diskBusyInfoDeviceIndex	An index value assigned by SNMP Agent. When the information acquired from the OS includes the same device name, the value is incremented.	--
1.1.3	diskBusyInfoTime	The accumulated time from the start of SNMP Agent.	Hundredths of a second
2	diskBusyInfoNum	The number of <code>diskBusyInfoTable</code> entries.	--

(20) fileSystem64 group

The following table describes the Hitachi enterprise-specific MIB objects in the `fileSystem64` group.

Table 4–48: `fileSystem64` group
(`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.fileSystem64`)
(1.116.5.1.2.1.21)

ID	Object name	Contents	Units
1	<code>fileSystem64Mounted</code>	The number of file systems mounted	--
2	<code>fileSystem64Table</code>	File system information table	--
2.1	<code>fileSystem64Entry</code>	Each entry is identified by <code>fileSystem64HighID1</code> , <code>fileSystem64LowID1</code> , or <code>fileSystem64ID2</code> .	--
2.1.1	<code>fileSystem64HighID1</code>	High bit of the file system ID	--
2.1.2	<code>fileSystem64LowID1</code>	Low bit of the file system ID	--
2.1.3	<code>fileSystem64ID2</code>	The value is incremented when <code>fileSystem64HighID1</code> and <code>fileSystem64LowID1</code> obtained from the OS are the same.	--
2.1.4	<code>fileSystem64Name</code>	The name of the mounted file system	--
2.1.5	<code>fileSystem64Block</code>	The number of blocks in the file system	Blocks
2.1.6	<code>fileSystem64Bfree</code>	The number of free blocks in the file system	Blocks
2.1.7	<code>fileSystem64Bavail</code>	The number of free blocks available to a non-superuser	Blocks
2.1.8	<code>fileSystem64Bsize</code>	The fundamental file system block size	Bytes
2.1.9	<code>fileSystem64Files</code>	The total number of inodes in the file system	--
2.1.10	<code>fileSystem64Ffree</code>	The total number of free inodes in the file system	--
2.1.11	<code>fileSystem64Dir</code>	The file system path	--

Important

Note the following about the Hewlett-Packard enterprise-specific MIB objects in the `fileSystem64` group:

- Objects with a `fileSystem64` group ID from 2.1.5 to 2.1.10 can only be acquired through SNMPv2c requests, not SNMPv1 requests.
- For details about how to prevent SNMP Agent from returning information on the target file system if it is not mounted in AIX and Linux, see [2.12.2 Settings for suppressing an invalid shared disk capacity response \(for AIX and Linux\)](#).
- For details about how to prevent unnecessary file system information from being returned in Linux, see [2.16 Settings to prevent responses with information about file systems for which a response is not required \(for Linux\)](#).

Important

The file system types that can be monitored in the `fileSystem64` group are described below. To monitor NFS, set the NFS server to monitor file systems.

- For HP-UX (IPF):
File systems described in `/etc/mnttab`
Note that file systems of type `swap`, `ignore`, or `nfs` cannot be monitored.
- For Solaris:
File systems described in `/etc/mnttab`
Note that file systems of the type `nfs` or `swap` cannot be monitored.
- For AIX:
File systems described in `/etc/filesystems`
- For Linux
File systems in `/etc/fstab` or `/etc/SnmpAgent.d/esafstab.conf`.
Note that for the file type `swap`, responses are not issued, even if the file system is in `/etc/fstab`, `/etc/SnmpAgent.d/esafstab.conf`.
This is because the system calls issued by the SNMP agent to acquire the file system cannot respond with this file type.

(21) diskBusyAvail group

The following table describes the Hitachi enterprise-specific MIB objects in the `diskBusyAvail` group:

Table 4–49: `diskBusyAvail` group
(`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskBusyAvail`)
(1.116.5.1.2.1.22)

ID	Object name	Contents	Units
1	<code>diskBusyAvailTable</code>	The disk busy ratio table	--
1.1	<code>diskBusyAvailEntry</code>	Each entry is identified by <code>diskBusyAvailDeviceName</code> or <code>diskBusyAvailDeviceIndex</code> .	--
1.1.1	<code>diskBusyAvailDeviceName</code>	Disk device name. One entry is created for each disk, not for a partition or disk slice, which is a kind of segment.	--

ID	Object name	Contents	Units
1.1.2	diskBusyAvailDeviceIndex	An index value assigned by SNMP Agent. When the information acquired from the OS includes the same device name, the value is incremented.	--
1.1.3	diskBusyAvailDiskBusy	The disk busy ratio within the specified time interval	%
1.1.4	diskBusyAvailTime	The time at which the disk busy ratio was obtained from the OS <i>Example:</i> 2004/11/11 14:47:00	--
2	diskBusyAvailNum	Number of diskBusyAvailTable entries	--
3	diskBusyAvailInterval	Interval time [#]	Minutes

[#]: The interval time (minutes) can be changed using the `htc_monagt1` option.

(22) disk64Ex group

The following table describes the Hitachi enterprise-specific MIB objects in the `disk64Ex` group.

Table 4–50: `disk64Ex` group (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.disk64Ex) (1.116.5.1.2.1.23)

ID	Object name	Contents	Units
1	disk64ExNum	The number of disk information entries.	--
2	disk64ExTable	The disk information table.	--
2.1	disk64ExEntry	Each entry is identified by <code>disk64ExDeviceName</code> and <code>disk64ExDeviceIndex</code> .	--
2.1.1	disk64ExDeviceName	The disk device name. [#]	--
2.1.2	disk64ExDeviceIndex	An index value assigned by SNMP Agent. If information acquired from the OS includes the same device name, the value is incremented. If it does not include the same device name, the value will be 1.	--
2.1.3	disk64ExDiskBusyTime	The disk busy time since machine startup.	Milliseconds
2.1.4	disk64ExRead	Amount of data read from the disk since machine startup.	Kilobytes
2.1.5	disk64ExWrite	Amount of data written to the disk since machine startup.	Kilobytes
2.1.6	disk64ExXfers	The number of data transfers (number of times I/O processing has occurred) since machine startup.	Number of times

[#]: In Linux, the disk devices listed in `/proc/diskstats` are retrieved, except for any disk device information that is specified in the `esadisk.conf` file to be excluded.

For details about how to set up `esadisk.conf`, see *Disk definition file (esadisk.conf)* in *Chapter 6. Definition Files*.

(23) computerSystem64 group

The following table describes the Hitachi enterprise-specific MIB objects in the `computerSystem64` group.

**Table 4–51: computerSystem64 group
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.computerSystem64)
(1.116.5.1.2.1.24)**

ID	Object name	Contents	Units
1	computerSystem64UpTime	Time elapsed since system startup	Hundredths of a second
7	computerSystem64FreeMemory ^{#1}	Amount of free space in physical memory	Kilobytes
8	computerSystem64PhysMemory	Physical memory size	Kilobytes
10	computerSystem64SwapConfig ^{#2, #3}	Size of device swap space	Kilobytes
11	computerSystem64EnabledSwap ^{#3}	Available size of disk swap space	Kilobytes
12	computerSystem64FreeSwap ^{#2}	Size of the actual free swap area	Kilobytes
13	computerSystem64UserCPU ^{#4}	CPU time used in the user mode with a nice value of 21 or above. In Solaris and AIX, this is the CPU time used in the user mode after startup of SNMP Agent.	Hundredths of a second
14	computerSystem64SysCPU ^{#4}	CPU time used in the kernel mode. In Solaris and AIX, this is the CPU time used in the kernel mode after startup of SNMP Agent.	Hundredths of a second
15	computerSystem64IdleCPU ^{#4}	CPU idle time. In Solaris and AIX, this is the CPU idle time after startup of SNMP Agent.	Hundredths of a second
16	computerSystem64NiceCPU ^{#4}	CPU time used in the user mode with a nice value of 20 or smaller. In Solaris and AIX, this is the CPU time used in the user mode after startup of SNMP Agent.	Hundredths of a second

#1

Note the following about `computerSystem64FreeMemory`:

Time required to obtain

Six seconds or longer is required to obtain the `computerSystem64FreeMemory` value in Solaris. Therefore, if the manager system issues SNMP requests addressed to SNMP Agent, specify a time-out value of 6 seconds or longer in the manager system.

For the amount of free space in physical memory in Solaris, AIX, and Linux, see [2.13 Notes about the amount of free space in physical memory](#).

#2

The following indicates whether the objects `computerSystem64SwapConfig` and `computerSystem64FreeSwap` of each OS include physical memory.

HP-UX (IPF) and Linux

Physical memory is not included.

Solaris

Physical memory is included.

AIX

In AIX, the actual paging space usage status is returned. Physical memory is not included.

#3

Note the following about the swap space in Solaris:

In the swap space in Solaris, the swap area on the disk contains real memory that is not used. In real memory, a virtual storage area is dynamically allocated. Therefore, the `computerSystem64SwapConfig` and

computerSystem64EnabledSwap values vary dynamically. For details about the swap space in Solaris, see [2.14 Notes about swap space size](#).

#4

For details about CPU information, see [2.15 Notes about CPU information](#).

(24) system group (in AIX)

The following table describes the Hitachi enterprise-specific MIB objects in the `system` group (in AIX).

Table 4–52: system group (enterprises.hitachi.systemExMib.cometMibs.systems.aix.system) (1.116.5.1.2.3.1)

ID	Object name	Contents	Units
1	systemCPUUtilUser	User CPU utilization (average over the past 1 second)	%
2	systemCPUUtilSystem	System CPU utilization (average over the past 1 second)	%
3	systemCPUUtilIdle	Idle CPU utilization (average over the past 1 second)	%
4	systemCPUUtilWait	Wait CPU utilization (average over the past 1 second)	%
5	systemActVirtualPage	The number of active virtual pages	Pages

(25) disk group (in AIX)

The following table describes the Hitachi enterprise-specific MIB objects in the `disk` group (in AIX).

Table 4–53: disk group (enterprises.hitachi.systemExMib.cometMibs.systems.aix.disk) (1.116.5.1.2.3.2)

ID	Object name	Contents	Units
1	diskNum	Number of disk information entries	--
2	diskTable	Disk input-output information	--
2.1	diskTableEntry	Each entry is distinguished from <code>diskIndex</code> .	--
2.1.1	diskIndex	The disk index	--
2.1.2	diskName	The disk name	--
2.1.3	diskRead	Amount of data read from the disk	Kilobytes
2.1.4	diskWrite	Amount of data written to the disk	Kilobytes

(26) page group (in AIX)

The following table describes the Hitachi enterprise-specific MIB objects in the `page` group (in AIX).

Table 4–54: page group (enterprises.hitachi.systemExMib.cometMibs.systems.aix.page) (1.116.5.1.2.3.3)

ID	Object name	Contents	Units
1	pageNum	Number of paging space entries	--
2	pageTable	Paging space organization and utilization rate	--
2.1	pageTableEntry	Each entry is distinguished from <code>pageIndex</code> .	--

ID	Object name	Contents	Units
2.1.1	pageIndex	Index number	--
2.1.2	pageSpaceName	Paging space name	--
2.1.3	pagePhysicalVol	The Physical Volume Name of paging space	--
2.1.4	pageVolGroup	The volume group name of paging space	--
2.1.5	pageSize	Paging space size	Megabytes
2.1.6	pageUsed	Paging space utilization rate	%
2.1.7	pageActive	The status of paging space 1: inactive (not being used) 2: active (being used)	--
2.1.8	pageAuto	Use of paging space at OS boot 1: not auto (not used) 2: auto (used)	--
2.1.9	pageType	Paging space type 1: lv (logical volume) 2: nfs (NFS volume)	--

(27) system group (in Solaris)

The following table describes the Hitachi enterprise-specific MIB objects in the `system` group (in Solaris).

Table 4–55: `system` group (enterprises.hitachi.systemExMib.cometMibs.systems.solaris.system)
(1.116.5.1.2.4.1)

ID	Object name	Contents	Units
1	systemCPUUtilUser	User CPU utilization (average value over the past 5 second)	%
2	systemCPUUtilSystem	System CPU utilization (average value over the past 5 second)	%
3	systemCPUUtilIdle	Idle CPU utilization (average value over the past 5 second)	%
4	systemCPUUtilWait	Wait CPU utilization (average value over the past 5 second)	%

(28) linuxSystem group (in Linux)

The following table describes Hitachi enterprise-specific MIB objects in the `linuxSystem` group (in Linux).

Table 4–56: `linuxSystem` group
(enterprises.hitachi.systemExMib.cometMibs.systems.linux.linuxsystem)
(1.116.5.1.2.5.1)

ID	Object name	Contents	Units
1	linuxSystemCPUUtilUser	User CPU utilization (average value over the past 1 second)	%
2	linuxSystemCPUUtilSystem	System CPU utilization (average value over the past 1 second)	%
3	linuxSystemCPUUtilIdle	Idle CPU utilization (average value over the past 1 second)	%

4.3.3 Implementation of Hitachi enterprise-specific MIB objects

Different groups of Hitachi enterprise-specific MIB objects are used depending on the system on which SNMP Agent is running. The following table lists the applicable OSs and the referenced location for each group of Hitachi enterprise-specific MIB objects.

Table 4–57: Implementation status and referenced location by OS for Hitachi enterprise-specific MIB objects

Group name (higher node)	Applicable OS				Referencing destination for MIB object implementation status
	HP-UX (IPF)	Solaris	AIX	Linux	
opConf (cometOP)	Y	Y	Y	Y	Table 4-58
systemInfo (hiux)	--	--	Y	--	Table 4-59
virtualMemory (hiux)	--	--	Y	--	Table 4-60
process (hiux)	--	--	Y	--	Table 4-61
processor (hiux)	--	Y	Y	Y	Table 4-62
diskInfo (hiux)	--	--	--	--	Table 4-63
swapInfo (hiux)	--	--	--	--	Table 4-64
swapSpace (hiux)	--	--	Y	--	Table 4-65
diskBusy (hiux)	--	Y	--	--	Table 4-66
swapSystem (hiux)	--	--	--	--	Table 4-67
cpuUtil (hiux)	--	Y	Y	Y	Table 4-68
virtualMemory64 (hiux)	--	--	Y	--	Table 4-69
systemInfo64 (hiux)	Y	--	--	--	Table 4-70
virtualMemory64Ex (hiux)	Y	--	--	--	Table 4-71
process64 (hiux)	Y	--	--	--	Table 4-72
processor64 (hiux)	Y	--	--	--	Table 4-73
diskInfo64 (hiux)	Y	--	--	--	Table 4-74
swapSystem64 (hiux)	Y	--	--	--	Table 4-75
diskBusyInfo (hiux)	--	Y	Y	--	Table 4-76
fileSystem64 (hiux)	Y	Y	Y	Y	Table 4-77
diskBusyAvail (hiux)	--	Y	Y	--	Table 4-78
disk64Ex (hiux)	--	--	Y	Y	Table 4-79
computerSystem64 (hiux)	Y	Y	Y	Y	Table 4-80
system (aix)	--	--	Y	--	Table 4-81
disk (aix)	--	--	Y	--	
page (aix)	--	--	Y	--	
system (solaris)	--	Y	--	--	Table 4-82

Group name (higher node)	Applicable OS				Referencing destination for MIB object implementation status
	HP-UX (IPF)	Solaris	AIX	Linux	
linuxSystem (linux)	--	--	--	Y	Table 4-83

Legends:

Y: Applicable

--: Not applicable

Note: The swapSystem (hiux) groups are not implemented in the MIB operations of Hitachi enterprise-specific MIB objects.

This section describes the implementation status of Hitachi enterprise-specific MIB objects in each group. These tables use the following legends:

Legends:

Y: A get or set operation can get or set the value of this MIB object.

N: A get or set operation cannot get or set the value of this MIB object (a noSuchName error is returned).

F (*value*): The object returns a fixed value indicated by *value*.

--: No access permission. A noSuchName error is returned.

(1) opConf group

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the opConf group.

**Table 4–58: Implementation of Hitachi enterprise-specific MIB objects (opConf group)
(enterprises.hitachi.systemExMib.cometMibs.subSystems.cometOP.opConf)
(1.116.5.1.1.2.10)**

Object name	MIB operation							
	HP-UX (IPF)		Solaris		AIX		Linux	
	get	set	get	set	get	set	get	set
opConfCharCode	Y	--	Y	--	Y	--	Y	--

(2) systemInfo group

SNMP Agent supports the systemInfo group under AIX.

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the systemInfo group.

**Table 4–59: Implementation of Hitachi enterprise-specific MIB objects (systemInfo group)
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.systemInfo)
(1.116.5.1.2.1.1)**

Object name	MIB operation			
	Solaris		AIX	
	get	set	get	set
systemInfoTTYMajor	N	--	--	--
systemInfoTTYMinor	N	--	--	--

Object name	MIB operation			
	Solaris		AIX	
	get	set	get	set
systemInfoBootTime	N	--	Y	--
systemInfoActiveProcessors	N	--	--	--
systemInfoMaxProcessors	N	--	--	--
systemInfoMaxProcesses	N	--	--	--
systemInfoRunQueProcesses	N	--	--	--
systemInfoXferWaitProcesses	N	--	--	--
systemInfoPageInWaitProcesses	N	--	--	--
systemInfoSleepProcesses	N	--	--	--
systemInfoSwapOutProcesses	N	--	--	--
systemInfoPhysicalMemorySize	N	--	--	--
systemInfoPhysicalMemoryFreeSize	N	--	--	--
systemInfoVirtualMemoryProcessSize	N	--	--	--
systemInfoVirtualMemoryWaitProcessSize	N	--	--	--
systemInfoPhysicalMemoryProcessSize	N	--	--	--
systemInfoPhysicalMemoryWaitProcessSize	N	--	--	--
systemInfoCPUStates	N	--	--	--
systemInfoOpenLogicalVolumes	N	--	--	--
systemInfoOpenLogicalVolumeGrps	N	--	--	--
systemInfoAllocPBUFs	N	--	--	--
systemInfoUsedPBUFs	N	--	--	--
systemInfoMaxPBUFs	N	--	--	--
systemInfoActiveProcessEntries	N	--	--	--
systemInfoActiveInodeEntries	N	--	--	--
systemInfoActiveFileEntries	N	--	--	--

(3) virtualMemory group

SNMP Agent supports the virtualMemory group under AIX.

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the virtualMemory group.

**Table 4–60: Implementation of Hitachi enterprise-specific MIB objects (virtualMemory group)
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.virtualMemory)
(1.116.5.1.2.1.2)**

Object name	MIB operation			
	Solaris		AIX	
	get	set	get	set
vmPageSize	N	--	Y	--
vmDaemonFreePages	N	--	Y	--
vmInterruptions	N	--	Y	--
vmPageInPages	N	--	Y	--
vmPageOutPages	N	--	Y	--
vmPageReclaims	N	--	Y	--
vmTLBFlashes	N	--	--	--
vmDaemonScanPages	N	--	--	--
vmContextSwitches	N	--	Y	--
vmSystemCalls	N	--	Y	--
vmXfileSystemFreeListPages	N	--	--	--
vmXSwapDeviceFreeListPages	N	--	--	--
vmFreeMemoryPages	N	--	--	--
vmTotalSwapIns	N	--	--	--
vmTotalSwapOuts	N	--	--	--
vmTotalDaemonFreePages	N	--	--	--
vmTotalDemandLoadPages	N	--	--	--
vmTotalPageFaults	N	--	--	--
vmTotalInterruptions	N	--	Y	--
vmTotalIntransitPageFaults	N	--	--	--
vmTotalDemandLoadCreatePages	N	--	--	--
vmTotalZeroFillCreatePages	N	--	--	--
vmTotalFreeListReclaimedPages	N	--	--	--
vmTotalPageIns	N	--	Y	--
vmTotalPageOuts	N	--	Y	--
vmTotalPageInPages	N	--	--	--
vmTotalPageOutPages	N	--	--	--
vmTotalSwapInPages	N	--	--	--
vmTotalSwapOutPages	N	--	--	--
vmTotalDaemonTicksNum	N	--	--	--
vmTotalContextSwitches	N	--	Y	--

Object name	MIB operation			
	Solaris		AIX	
	get	set	get	set
vmTotalSystemCalls	N	--	Y	--
vmTotalTraps	N	--	Y	--
vmTotalXfileSystemFreeListPages	N	--	--	--
vmTotalXSwapDeviceFreeListPages	N	--	--	--
vmTotalDemandZeroFillPages	N	--	--	--
vmTotalDaemonScanPages	N	--	--	--
vmTotalReclaimedPages	N	--	--	--
vmTotalDeficitPages	N	--	--	--
vmTotalReadChars	N	--	--	--
vmTotalWriteChars	N	--	--	--
vmTotalForks	N	--	Y	--
vmTotalForkPages	N	--	--	--
vmTotalDiskBlockReads	N	--	--	--
vmTotalDiskBlockWrites	N	--	--	--
vmTotalProcessSwapOuts	N	--	--	--
vmTotalswapOutProcesses	N	--	--	--

(4) process group

SNMP Agent supports the `process` group under AIX.

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the `process` group.

Table 4–61: Implementation of Hitachi enterprise-specific MIB objects (process group)
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.process) (1.116.5.1.2.1.3)

Object name	MIB operation				
	Solaris		AIX		
	get	set	get	set	
processNum	N	--	Y	--	
processTable	processID	N	--	Y	--
processEntry	processIndex	N	--	--	--
.~	processStatus	N	--	--	--
	processStateFlags	N	--	--	--
	processUserID	N	--	--	--
	processSavedUserID	N	--	--	--

Object name		MIB operation			
		Solaris		AIX	
		get	set	get	set
	processParentID	N	--	Y	--
	processGroupID	N	--	--	--
	processCPUUtilization	N	--	--	--
	processPriority	N	--	Y	--
	processCPUNice	N	--	Y	--
	processProcessor	N	--	--	--
	processStartTime	N	--	--	--
	processPhysicalMemoryTextSize	N	--	--	--
	processPhysicalMemoryDataSize	N	--	--	--
	processPhysicalMemoryStackSize	N	--	--	--
	processPhysicalMemorySharedMemorySize	N	--	--	--
	processPhysicalMemoryMemoryMappedSize	N	--	--	--
	processPhysicalMemoryUserSize	N	--	--	--
	processPhysicalMemoryIOSize	N	--	--	--
	processVirtualMemoryTextSize	N	--	--	--
	processVirtualMemoryDataSize	N	--	--	--
	processVirtualMemoryStackSize	N	--	--	--
	processVirtualMemorySharedMemorySize	N	--	--	--
	processVirtualMemoryMemoryMappedSize	N	--	--	--
	processVirtualMemoryUserSize	N	--	--	--
	processVirtualMemoryIOSize	N	--	--	--
	processResidentSize	N	--	--	--
	processAddress	N	--	--	--
	processSleepAddress	N	--	--	--
	processUserTime	N	--	--	--
	processSystemTime	N	--	--	--

Object name		MIB operation			
		Solaris		AIX	
		get	set	get	set
	processTTYMajor	N	--	--	--
	processTTYMinor	N	--	--	--
	processCommand	N	--	Y	--
	processExecutable	N	--	--	--
	processResidentTime	N	--	--	--
	processCPUTimeTicks	N	--	--	--
	processTotalCPUTimeTicks	N	--	Y	--
	processFssID	N	--	--	--
	processResidentTimeCPU	N	--	--	--
	processMinorFaults	N	--	--	--
	processMajorFaults	N	--	--	--
	processSwapOuts	N	--	--	--
	processSignals	N	--	--	--
	processReceivedMessages	N	--	--	--
	processSentMessages	N	--	--	--
	processMaxResidentSize	N	--	--	--
	processUser	N	--	Y	--

(5) processor group

SNMP Agent does not support the `processor` group under HP-UX (IPF).

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the `processor` group.

Table 4–62: Implementation of Hitachi enterprise-specific MIB objects (processor group)
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.pocessor)
(1.116.5.1.2.1.4)

Object name		MIB operation					
		Solaris		AIX		Linux	
		get	set	get	set	get	set
	processorNum	Y#	--	Y#	--	Y#	--
processorTable. processorEntry.~	processorIndex	Y#	--	Y#	--	Y#	--
	processorFileSystemReadBytes	N	--	--	--	--	--
	processorFileSystemWriteBytes	N	--	--	--	--	--
	processorDiskBlockReadRequests	N	--	--	--	--	--

Object name		MIB operation					
		Solaris		AIX		Linux	
		get	set	get	set	get	set
processorDiskBlockWriteRequests	N	--	--	--	--	--	
processorNFSReadBytes	N	--	--	--	--	--	
processorNFSWriteBytes	N	--	--	--	--	--	
processorPhysicalReads	N	--	--	--	--	--	
processorPhysicalWrites	N	--	--	--	--	--	
processorRunQueues	N	--	--	--	--	--	
processorRunQueueProcesses	N	--	--	--	--	--	
processorSysExecs	N	--	--	--	--	--	
processorSysReads	N	--	--	--	--	--	
processorSysWrites	N	--	--	--	--	--	
processorSysNamis	N	--	--	--	--	--	
processorSysIgets	N	--	--	--	--	--	
processorDirFileSystemReadBytes	N	--	--	--	--	--	
processorSemaphoreOperations	N	--	--	--	--	--	
processorMessageOperations	N	--	--	--	--	--	
processorInMUXInterruptions	N	--	--	--	--	--	
processorOutMUXInterruptions	N	--	--	--	--	--	
processorTTYRawChars	N	--	--	--	--	--	
processorTTYCanonChars	N	--	--	--	--	--	
processorTTYOutChars	N	--	--	--	--	--	
processorCPULoadAvg1	N	--	--	--	--	--	
processorCPULoadAvg5	N	--	--	--	--	--	
processorCPULoadAvg15	N	--	--	--	--	--	
processorUserCPUTime	Y#	--	Y#	--	Y#	--	
processorNiceCPUTime	--	--	--	--	Y#	--	
processorSysCPUTime	Y#	--	Y#	--	Y#	--	
processorIdleCPUTime	Y#	--	Y#	--	Y#	--	
processorWaitCPUTime	Y#	--	Y#	--	--	--	
processorBlockCPUTime	N	--	--	--	--	--	
processorSwaitCPUTime	N	--	--	--	--	--	
processorIntrCPUTime	N	--	--	--	--	--	
processorSsysCPUTime	N	--	--	--	--	--	

#

See the note about CPU information in [4.2.2 \(1\) computerSystem group](#).

(6) diskInfo group

SNMP Agent does not support the `diskInfo` group.

Table 4–63: Implementation of Hitachi enterprise-specific MIB objects (`diskInfo` group)
(`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskInfo`) (1.116.5.1.2.1.5)

Object name	MIB operation	
	get	set
<code>diskNum</code>	Not implemented.	Not implemented.
<code>diskTable. diskEntry.~</code>		
<code>diskIndex</code>		
<code>diskTTYMajor</code>		
<code>diskTTYMinor</code>		
<code>diskBusyTimeTicks</code>		
<code>diskSeeks</code>		
<code>diskXfers</code>		
<code>diskWordsXfers</code>		
<code>diskWordsWriteTime</code>		

(7) swapInfo group

SNMP Agent does not support the `swapInfo` group.

Table 4–64: Implementation of Hitachi enterprise-specific MIB objects (`swapInfo` group)
(`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.swapInfo`)
(1.116.5.1.2.1.6)

Object name	MIB operation	
	get	set
<code>swapTotalSize</code>	Not implemented.	Not implemented.
<code>swapTotalEnabledSize</code>		
<code>swapTotalFreeSize</code>		
<code>swapTotalBlockDeviceSize</code>		
<code>swapTotalBlockDeviceEnabledSize</code>		
<code>swapTotalBlockDeviceFreeSize</code>		
<code>swapTotalFreeSystemSize</code>		
<code>swapTotalFileSystemEnabledSize</code>		
<code>swapTotalFileSystemFreeSize</code>		
<code>swapNum</code>		
<code>swapTable.</code>	<code>swapIndex</code>	

Object name		MIB operation	
		get	set
swapEntry.~	swapPlace		
	swapFlags		
	swapPriority		
	swapFreeSize		
	swapBlockDeviceMajor		
	swapBlockDeviceMinor		
	swapBlockDeviceStartNum		
	swapFileSystemSize		
	swapFileSystemMinSize		
	swapFileSystemMaxSize		
	swapFileSystemReservedSize		
	swapFileSystemMountPoint		

(8) swapSpace group

SNMP Agent supports the `swapSpace` group under AIX.

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the `swapSpace` group:

Table 4–65: Implementation of Hitachi enterprise-specific MIB objects (swapSpace group) (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.swapSpace) (1.116.5.1.2.1.7)

Object name	MIB operation					
	Solaris		AIX		Linux	
	get	set	get	set	get	set
swapSpaceConfig	--	--	Y	--	--	--
swapSpaceEnable	--	--	Y	--	--	--
swapSpaceFree	--	--	Y	--	--	--

(9) diskBusy group

SNMP Agent supports the `diskBusy` group in Solaris.

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the `diskBusy` group:

**Table 4–66: Implementation of Hitachi enterprise-specific MIB objects (diskBusy group)
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskBusy)
(1.116.5.1.2.1.8)**

Object name		MIB operation	
		Solaris	
		get	set
diskBusyNum		Y	--
diskBusyTable. diskBusyEntry~	diskBusyDeviceName	Y	--
	diskBusyUtil	Y	--

(10) swapSystem group

SNMP Agent does not support the swapSystem group.

**Table 4–67: Implementation of Hitachi enterprise-specific MIB objects (swapSystem group)
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.swapSystem)
(1.116.5.1.2.1.9)**

Object name		MIB operation	
		get	set
swapSystemTotalSize		Not implemented.	Not implemented.
swapSystemTotalEnableSize			
swapSystemTotalFreeSize			
swapSystemTotalBlockDeviceSize			
swapSystemTotalBlockDeviceEnabledSize			
swapSystemTotalBlockDeviceFreeSize			
swapSystemTotalFileSystemSize			
swapSystemTotalFileSystemEnabledSize			
swapSystemTotalFileSystemFreeSize			
swapSystemNum			
swapSystemTable. swapSystemEntry~	swapSystemIndex		
	swapSystemPlace		
	swapSystemFlags		
	swapSystemPriority		
	swapSystemFreeSize		
	swapSystemBlockDeviceMajor		
	swapSystemBlockDeviceMinor		
	swapSystemBlockDeviceStartNum		
	swapSystemBlockDeviceSize		
swapSystemFileSystemSize			

Object name		MIB operation	
		get	set
	swapSystemFileSystemMinSize		
	swapSystemFileSystemMaxSize		
	swapSystemFileSystemReservedSize		
	swapSystemFileSystemMountPoint		

(11) cpuUtil group

SNMP Agent does not support the `cpuUtil` group under HP-UX (IPF).

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the `cpuUtil` group.

Table 4–68: Implementation of Hitachi enterprise-specific MIB objects (`cpuUtil` group)
(`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.cpuUtil`) (1.116.5.1.2.1.10)

Object name		MIB operation					
		Solaris		AIX		Linux	
		get	set	get	set	get	set
<code>cpuUtilTable.cpuUtilEntry</code>	<code>cpuUtilNum</code>	Y	--	Y	--	Y	--
	<code>cpuUtilUser</code>	Y	--	Y	--	Y	--
	<code>cpuUtilSystem</code>	Y	--	Y	--	Y	--
	<code>cpuUtilWio</code>	Y	--	Y	--	Y	--
	<code>cpuUtilIdle</code>	Y	--	Y	--	Y	--
	<code>cpuUtilTime</code>	Y	--	Y	--	Y	--
<code>cpuUtilInterval</code>		Y	--	Y	--	Y	--
<code>cpuUtilTotalUser</code>		Y	--	Y	--	Y	--
<code>cpuUtilTotalSystem</code>		Y	--	Y	--	Y	--
<code>cpuUtilTotalWio</code>		Y	--	Y	--	Y	--
<code>cpuUtilTotalIdle</code>		Y	--	Y	--	Y	--

(12) virtualMemory64 group

SNMP Agent supports the `virtualMemory64` group under AIX.

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the `virtualMemory64` group.

Table 4–69: Implementation of Hitachi enterprise-specific MIB objects (virtualMemory64 group) (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.virtualMemory64) (1.116.5.1.2.1.13)

Object name	MIB operation	
	AIX	
	get	set
vm64PageSize	Y	--
vm64DaemonfreePages	Y	--
vm64Interruptions	Y	--
vm64PageInPages	Y	--
vm64PageOutPages	Y	--
vm64PageReclaims	Y	--
vm64TLBFlashes	--	--
vm64DaemonScanPages	--	--
vm64ContextSwitches	Y	--
vm64SystemCalls	Y	--
vm64XFileSystemFreelistPages	--	--
vm64XSwapDeviceFreeListPages	--	--
vm64FreeMemoryPages	--	--
vm64TotalSwapIns	--	--
vm64TotalSwapOuts	--	--
vm64TotalDaemonFreePages	--	--
vm64TotalDemandLoadPages	--	--
vm64TotalPageFaults	--	--
vm64TotalInterruptions	Y	--
vm64TotalIntransitPageFaults	--	--
vm64TotalDemandLoadCreatePages	--	--
vm64TotalZeroFillCreatePages	--	--
vm64TotalFreeListReclaimedPages	--	--
vm64TotalPageIns	Y	--
vm64TotalPageOuts	Y	--
vm64TotalPageInPages	--	--
vm64TotalPageOutPages	--	--
vm64TotalSwapInPages	--	--
vm64TotalSwapOutPages	--	--
vm64TotalDaemonTicksNum	--	--
vm64TotalContextSwitches	Y	--

Object name	MIB operation	
	AIX	
	get	set
vm64TotalSystemCalls	Y	--
vm64TotalTraps	Y	--
vm64TotalXFileSystemFreeListPages	--	--
vm64TotalXSwapDeviceFreeListPages	--	--
vm64TotalDemandZeroFillPages	--	--
vm64TotalDaemonScanPages	--	--
vm64TotalReclaimedPages	--	--
vm64TotalDeficitPages	--	--
vm64TotalReadChars	--	--
vm64TotalWriteChars	--	--
vm64TotalForks	Y	--
vm64TotalForkPages	--	--
vm64TotalDiskBlockReads	--	--
vm64TotalDiskBlockWrites	--	--
vm64TotalProcessSwapOuts	--	--
vm64TotalSwapOutProcesses	--	--

(13) systemInfo64 group

SNMP Agent supports the `systemInfo64` group under HP-UX (IPF).

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the `systemInfo64` group.

Table 4–70: Implementation of Hitachi enterprise-specific MIB objects (systemInfo64 group) (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.systemInfo64) (1.116.5.1.2.1.14)

Object name	MIB operation	
	HP-UX (IPF)	
	get	set
systemInfo64TTYMajor	Y	--
systemInfo64TTYMinor	Y	--
systemInfo64BootTime	Y	--
systemInfo64ActiveProcessors	Y	--
systemInfo64MaxProcessors	Y	--
systemInfo64MaxProcesses	Y	--

Object name	MIB operation	
	HP-UX (IPF)	
	get	set
systemInfo64RunQueProcesses	Y	--
systemInfo64XferWaitProcesses	Y	--
systemInfo64PageInWaitProcesses	Y	--
systemInfo64SleepProcesses	Y	--
systemInfo64SwapOutProcesses	Y	--
systemInfo64PhysicalMemorySize	Y	--
systemInfo64PhysicalMemoryFreeSize	Y	--
systemInfo64VirtualMemoryProcessSize	Y	--
systemInfo64VirtualMemoryWaitProcessSize	Y	--
systemInfo64PhysicalMemoryProcessSize	Y	--
systemInfo64PhysicalMemoryWaitProcessSize	Y	--
systemInfo64CPUStates	Y	--
systemInfo64OpenLogicalVolumes	Y	--
systemInfo64OpenLogicalVolumeGrps	Y	--
systemInfo64AllocPBUFs	Y	--
systemInfo64UsedPBUFs	Y	--
systemInfo64MaxPBUFs	Y	--
systemInfo64ActiveProcessEntries	Y	--
systemInfo64ActiveInodeEntries	Y	--
systemInfo64ActiveFileEntries	Y	--

(14) virtualMemory64Ex group

SNMP Agent supports the `virtualMemory64Ex` group under HP-UX (IPF).

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the `virtualMemory64Ex` group.

Table 4–71: Implementation of Hitachi enterprise-specific MIB objects (virtualMemory64Ex group) (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.virtualMemory64Ex) (1.116.5.1.2.1.15)

Object name	MIB operation	
	HP-UX (IPF)	
	get	set
vm64ExPageSize	Y	--
vm64ExDaemonfreePages	Y	--

Object name	MIB operation	
	HP-UX (IPF)	
	get	set
vm64ExInterruptions	Y	--
vm64ExPageInPages	Y	--
vm64ExPageOutPages	Y	--
vm64ExPageReclaims	Y	--
vm64ExTLBFlashes	Y	--
vm64ExDaemonScanPages	Y	--
vm64ExContextSwitches	Y	--
vm64ExSystemCalls	Y	--
vm64ExXFileSystemFreelistPages	Y	--
vm64ExXSwapDeviceFreeLlistPages	Y	--
vm64ExFreeMemoryPages	Y	--
vm64ExTotalSwapIns	Y	--
vm64ExTotalSwapOuts	Y	--
vm64ExTotalDaemonFreePages	Y	--
vm64ExTotalDemandLoadPages	Y	--
vm64ExTotalPageFaults	Y	--
vm64ExTotalInterruptions	Y	--
vm64ExTotalIntransitPageFaults	Y	--
vm64ExTotalDemandLoadCreatePages	Y	--
vm64ExTotalZeroFillCreatePages	Y	--
vm64ExTotalFreeListReclaimedPages	Y	--
vm64ExTotalPageIns	Y	--
vm64ExTotalPageOuts	Y	--
vm64ExTotalPageInPages	Y	--
vm64ExTotalPageOutPages	Y	--
vm64ExTotalSwapInPages	Y	--
vm64ExTotalSwapOutPages	Y	--
vm64ExTotalDaemonTicksNum	Y	--
vm64ExTotalContextSwitches	Y	--
vm64ExTotalSystemCalls	Y	--
vm64ExTotalTraps	Y	--
vm64ExTotalXFileSystemFreeListPages	Y	--
vm64ExTotalXSwapDeviceFreeListPages	Y	--

Object name	MIB operation	
	HP-UX (IPF)	
	get	set
vm64ExTotalDemandZeroFillPages	Y	--
vm64ExTotalDaemonScanPages	Y	--
vm64ExTotalReclaimedPages	Y	--
vm64ExTotalDeficitPages	Y	--
vm64ExTotalReadChars	Y	--
vm64ExTotalWriteChars	Y	--
vm64ExTotalForks	Y	--
vm64ExTotalForkPages	Y	--
vm64ExTotalDiskBlockReads	Y	--
vm64ExTotalDiskBlockWrites	Y	--
vm64ExTotalProcessSwapOuts	Y	--
vm64ExTotalSwapOutProcesses	Y	--

(15) process64 group

SNMP Agent supports the `process64` group under HP-UX (IPF).

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the `process64` group.

Table 4–72: Implementation of Hitachi enterprise-specific MIB objects (process64 group)
 (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.process64)
 (1.116.5.1.2.1.16)

Object name	MIB operation		
	HP-UX (IPF)		
	get	set	
process64Num	Y	--	
process64Table.process64Entry.~	process64ID	Y	--
	process64Index	Y	--
	process64Status	Y	--
	process64StatusFlags	Y	--
	process64UserID	Y	--
	process64SavedUserID	Y	--
	process64ParentID	Y	--
	process64GroupID	Y	--
process64CPUUtilization	Y	--	

Object name		MIB operation	
		HP-UX (IPF)	
		get	set
	process64Priority	Y	--
	process64CPUNice	Y	--
	process64Processor	Y	--
	process64StartTime	Y	--
	process64PhysicalMemoryTextSize	Y	--
	process64PhysicalMemoryDataSize	Y	--
	process64PhysicalMemoryStackSize	Y	--
	process64PhysicalMemorySharedMemorySize	Y	--
	process64PhysicalMemoryMemoryMappedSize	Y	--
	process64PhysicalMemoryUserSize	Y	--
	process64PhysicalMemoryIOSize	Y	--
	process64VirtualMemoryTextSize	Y	--
	process64VirtualMemoryDataSize	Y	--
	process64VirtualMemoryStackSize	Y	--
	process64VirtualMemorySharedMemorySize	Y	--
	process64VirtualMemoryMemoryMappedSize	Y	--
	process64VirtualMemoryUserSize	Y	--
	process64VirtualMemoryIOSize	Y	--
	process64ResidentSize	Y	--
	process64Address	Y	--
	process64SleepAddress	Y	--
	process64UserTime	Y	--
	process64SystemTime	Y	--
	process64TTYMajor	Y	--
	process64TTYMinor	Y	--
	process64Command	Y	--
	process64Executable	Y	--
	process64ResidentTime	Y	--
	process64CPUTimeTicks	Y	--
	process64TotalCPUTimeTicks	Y	--
	process64FssID	Y	--
	process64ResidentTimeCPU	Y	--
	process64MinorFaults	Y	--

Object name		MIB operation	
		HP-UX (IPF)	
		get	set
	process64MajorFaults	Y	--
	process64SwapOuts	Y	--
	process64Signals	Y	--
	process64ReceivedMessages	Y	--
	process64SentMessages	Y	--
	process64MaxResidentSize	Y	--
	process64User	Y	--

(16) processor64 group

SNMP Agent supports the `processor64` group under HP-UX (IPF).

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the `processor64` group.

Table 4–73: Implementation of Hitachi enterprise-specific MIB objects (processor64 group) (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.processor64) (1.116.5.1.2.1.17)

Object name		MIB operation	
		HP-UX (IPF)	
		get	set
processor64Num		Y	--
processor64Table.processor64Entry.~	processor64Index	Y	--
	processor64FileSystemReadBytes	Y	--
	processor64FileSystemWriteBytes	Y	--
	processor64DiskBlockReadRequests	Y	--
	processor64DiskBlockWriteRequests	Y	--
	processor64NFSReadBytes	Y	--
	processor64NFSWriteBytes	Y	--
	processor64PhysicalReads	Y	--
	processor64PhysicalWrites	Y	--
	processor64RunQueues	Y	--
	processor64RunQueueProcesses	Y	--
	processor64SysExecs	Y	--
	processor64SysReads	Y	--
processor64SysWrites	Y	--	

Object name		MIB operation	
		HP-UX (IPF)	
		get	set
	processor64SysNamis	Y	--
	processor64SysIgets	Y	--
	processor64DirFileSystemReadBytes	Y	--
	processor64SemaphoreOperations	Y	--
	processor64MessageOperations	Y	--
	processor64InMUXInterruptions	Y	--
	processor64OutMUXInterruptions	Y	--
	processor64TTYRawChars	Y	--
	processor64TTYCanonChars	Y	--
	processor64TTYOutChars	Y	--
	processor64CPULoadAvg1	Y	--
	processor64CPULoadAvg5	Y	--
	processor64CPULoadAvg15	Y	--
	processor64UserCPUtime	Y	--
	processor64NiceCPUtime	Y	--
	processor64SysCPUtime	Y	--
	processor64IdleCPUtime	Y	--
	processor64WaitCPUtime	Y	--
	processor64BlockCPUtime	Y	--
	processor64SwaitCPUtime	Y	--
	processor64IntrCPUtime	Y	--
	processor64SsysCPUtime	Y	--

(17) diskInfo64 group

SNMP Agent supports the `diskInfo64` group under HP-UX (IPF).

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the `diskInfo64` group.

**Table 4–74: Implementation of Hitachi enterprise-specific MIB objects (diskInfo64 group)
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskInfo64)
(1.116.5.1.2.1.18)**

Object name		MIB operation	
		HP-UX (IPF)	
		get	set
disk64Num		Y	--
disk64Table.disk64Entry.~	disk64Index	Y	--
	disk64TTYMajor	Y	--
	disk64TTYMinor	Y	--
	disk64BusyTimeTicks	Y	--
	disk64Seeks	Y	--
	disk64Xfers	Y	--
	disk64WordsXfers	Y	--
	disk64WordsWriteTime	Y	--

(18) swapSystem64 group

SNMP Agent supports the `swapSystem64` group under HP-UX (IPF).

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the `swapSystem64` group.

**Table 4–75: Implementation of Hitachi enterprise-specific MIB objects (swapSystem64 group)
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.swapSystem64)
(1.116.5.1.2.1.19)**

Object name		MIB operation	
		HP-UX (IPF)	
		get	set
swapSystem64TotalSize		Y	--
swapSystem64TotalEnabledSize		Y	--
swapSystem64TotalFreeSize		Y	--
swapSystem64TotalBlockDeviceSize		Y	--
swapSystem64TotalBlockDeviceEnabledSize		Y	--
swapSystem64TotalBlockDeviceFreeSize		Y	--
swapSystem64TotalFileSystemSize		Y	--
swapSystem64TotalFileSystemEnabledSize		Y	--
swapSystem64TotalFileSystemFreeSize		Y	--
swapSystem64Num		Y	--
swapSystem64Table.swapSystem64Entry.~	swapSystem64Index	Y	--

Object name		MIB operation	
		HP-UX (IPF)	
		get	set
	swapSystem64Place	Y	--
	swapSystem64Flags	Y	--
	swapSystem64Priority	Y	--
	swapSystem64FreeSize	Y	--
	swapSystem64BlockDeviceMajor	Y	--
	swapSystem64BlockDeviceMinor	Y	--
	swapSystem64BlockDeviceStartNum	Y	--
	swapSystem64BlockDeviceSize	Y	--
	swapSystem64FileSystemSize	Y	--
	swapSystem64FileSystemMinSize	Y	--
	swapSystem64FileSystemMaxSize	Y	--
	swapSystem64FileSystemReservedSize	Y	--
swapSystem64FileSystemMountPoint	Y	--	

(19) diskBusyInfo group

SNMP Agent does not support the `diskBusyInfo` group under HP-UX (IPF) and Linux.

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects in the `diskBusyInfo` group.

Table 4–76: Implementation of Hitachi enterprise-specific MIB objects (diskBusyInfo group) (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskBusyInfo) (1.116.5.1.2.1.20)

Object name		MIB operation			
		Solaris		AIX	
		get	set	get	set
diskBusyInfoTable.diskBusyInfoEntry~	diskBusyInfoDeviceName	Y	--	Y	--
	diskBusyInfoDeviceIndex	Y	--	Y	--
	diskBusyInfoTime	Y	--	Y	--
diskBusyInfoNum		Y	--	Y	--

Important

All MIB values in the above table are updated at every interval for acquiring disk busy time (default: 5 minutes) that is set in SNMP Agent in Solaris or AIX. Thus, to collect disk busy time in Solaris or AIX, set a collection interval that is longer than the acquisition interval for disk busy time. Furthermore, in Solaris or AIX, during the period from the start of SNMP Agent until the first acquisition of disk busy time, all the

MIB values of disk busy time are returned as a `noSuchName` error. The interval time (minutes) can be changed using the `htc_monagt1` option.

(20) fileSystem64 group

The following table describes the Hitachi enterprise-specific MIB objects in the `fileSystem64` group.

Table 4–77: Implementation of Hitachi enterprise-specific MIB objects (fileSystem64 group)
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.fileSystem64)
(1.116.5.1.2.1.21)

Object name	MIB operation							
	HP-UX (IPF)		Solaris		AIX		Linux	
	get	set	get	set	get	set	get	set
<code>fileSystem64Mounted</code>	Y	--	Y	--	Y	--	Y	--
<code>fileSystem64Table</code>	--	--	--	--	--	--	--	--
<code>fileSystem64Entry</code>	Y	--	Y	--	Y	--	Y	--
<code>fileSystem64HighID1</code>	Y	--	Y	--	Y	--	Y	--
<code>fileSystem64LowID1</code>	Y	--	Y	--	Y	--	Y	--
<code>fileSystem64ID2</code>	Y	--	Y	--	Y	--	Y	--
<code>fileSystem64Name</code>	Y	--	Y	--	Y	--	Y	--
<code>fileSystem64Block</code>	Y	--	Y	--	Y	--	Y	--
<code>fileSystem64Bfree</code>	Y	--	Y	--	Y	--	Y	--
<code>fileSystem64Bavail</code>	Y	--	Y	--	Y	--	Y	--
<code>fileSystem64Bsize</code>	Y	--	Y	--	Y	--	Y	--
<code>fileSystem64Files</code>	Y	--	Y	--	Y	--	Y	--
<code>fileSystem64Ffree</code>	Y	--	Y	--	Y	--	Y	--
<code>fileSystem64Dir</code>	Y	--	Y	--	Y	--	Y	--

(21) diskBusyAvail group

The following table describes the Hitachi enterprise-specific MIB objects in the `diskBusyAvail` group.

Table 4–78: Implementation of Hitachi enterprise-specific MIB objects (diskBusyAvail group)
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskBusyAvail)
(1.116.5.1.2.1.22)

Object name	MIB operation							
	HP-UX (IPF)		Solaris		AIX		Linux	
	get	set	get	set	get	set	get	set
<code>diskBusyAvailTable</code>	--	--	Y	--	Y	--	--	--
<code>diskBusyAvailEntry</code>	--	--	--	--	--	--	--	--

Object name	MIB operation							
	HP-UX (IPF)		Solaris		AIX		Linux	
	get	set	get	set	get	set	get	set
diskBusyAvailDeviceName	--	--	Y	--	Y	--	--	--
diskBusyAvailDeviceIndex	--	--	Y	--	Y	--	--	--
diskBusyAvailDiskBusy	--	--	Y	--	Y	--	--	--
diskBusyAvailTime	--	--	Y	--	Y	--	--	--
diskBusyAvailNum	--	--	Y	--	Y	--	--	--
diskBusyAvailInterval	--	--	Y	--	Y	--	--	--

(22) disk64Ex group

The following table describes the Hitachi enterprise-specific MIB objects in the `disk64Ex` group.

Table 4–79: Implementation of Hitachi enterprise-specific MIB objects (disk64Ex group) (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.disk64Ex) (1.116.5.1.2.1.23)

Object name	MIB operation							
	HP-UX (IPF)		Solaris		AIX		Linux	
	get	set	get	set	get	set	get	set
disk64ExNum	--	--	--	--	Y	--	Y	--
disk64ExTable	--	--	--	--	Y	--	Y	--
disk64ExEntry	--	--	--	--	Y	--	Y	--
disk64ExDeviceName	--	--	--	--	Y	--	Y	--
disk64ExDeviceIndex	--	--	--	--	Y	--	Y	--
disk64ExDiskBusyTime	--	--	--	--	--	--	Y	--
disk64ExRead	--	--	--	--	Y	--	Y	--
disk64ExWrite	--	--	--	--	Y	--	Y	--
disk64ExXfers	--	--	--	--	--	--	Y	--

(23) computerSystem64 group

The following table describes the Hitachi enterprise-specific MIB objects in the `computerSystem64` group.

Table 4–80: Implementation of Hitachi enterprise-specific MIB objects (computerSystem64 group) (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.computerSystem64) (1.116.5.1.2.1.24)

Object name	MIB operation							
	HP-UX (IPF)		Solaris		AIX		Linux	
	get	set	get	set	get	set	get	set
computerSystem64UpTime	Y	--	Y	--	Y	--	Y	--
computerSystem64FreeMemory	Y	--	Y	--	Y	--	Y	--
computerSystem64PhysMemory	Y	--	Y	--	Y	--	Y	--
computerSystem64SwapConfig	--	--	Y	--	Y	--	Y	--
computerSystem64EnabledSwap	--	--	Y	--	Y	--	Y	--
computerSystem64FreeSwap	--	--	Y	--	Y	--	Y	--
computerSystem64UserCPU	Y	--	Y	--	Y	--	Y	--
computerSystem64SysCPU	Y	--	Y	--	Y	--	Y	--
computerSystem64IdleCPU	Y	--	Y	--	Y	--	Y	--
computerSystem64NiceCPU	Y	--	--	--	--	--	Y	--

(24) Groups specific to AIX

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects specific to AIX.

Table 4–81: Implementation of Hitachi enterprise-specific MIB objects (groups specific to AIX) (enterprises.hitachi.systemExMib.cometMibs.systems.aix) (1.116.5.1.2.3)

Object name		MIB operation		
		get	set	
system group (1)	systemCPUUtilUser	Y	--	
	systemCPUUtilSystem	Y	--	
	systemCPUUtilIdle	Y	--	
	systemCPUUtilWait	Y	--	
	systemActVirtualPage	Y	--	
disk group (2)	diskNum	Y	--	
	diskTable. diskTableEntry.~	diskIndex	Y	--
		diskName	Y	--
		diskRead	Y	--
	diskWrite	Y	--	
page group (3)	pageNum	Y	--	
	pageTable. pageTableEntry.~	pageIndex	Y	--
		pageSpaceName	Y	--

Object name		MIB operation	
		get	set
	pagePhysicalVol	Y	--
	pageSize	Y	--
	pageUsed	Y	--
	pageActive	Y	--
	pageAuto	Y	--
	pageType	Y	--

(25) Groups specific to Solaris

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects specific to Solaris.

Table 4–82: Implementation of Hitachi enterprise-specific MIB objects (groups specific to Solaris) (enterprises.hitachi.systemExMib.cometMibs.systems.solaris) (1.116.5.1.2.4)

Object name		MIB operation	
		get	set
system group (1)	systemCPUUtilUser	Y	--
	systemCPUUtilSystem	Y	--
	systemCPUUtilIdle	Y	--
	systemCPUUtilWait	Y	--

(26) Groups specific to Linux

The following table shows the implementation status of the Hitachi enterprise-specific MIB objects specific to Linux.

Table 4–83: Implementation of Hitachi enterprise-specific MIB objects (groups specific to Linux) (enterprises.hitachi.systemExMib.cometMibs.systems.linux) (1.116.5.1.2.5)

Object name		MIB operation	
		get	set
linuxSystem group (1)	linuxSystemCPUUtilUser	Y	--
	linuxSystemCPUUtilSystem	Y	--
	linuxSystemCPUUtilIdle	Y	--

5

Commands and Processes

This chapter explains the functions and syntax of commands used with SNMP Agent. It also explains the functions and startup options of the SNMP Agent processes.

Commands

This section explains the functions and syntax of commands used with SNMP Agent.

List of commands

The following table lists the SNMP Agent commands.

Table 5–1: SNMP Agent commands

Command name	Function
<code>jpllesalog.sh.def</code> [#]	Collects system information from a machine on which a problem occurred.
<code>snmpcheck</code>	Displays the operating status (running or not running) of the master agent, subagents, and native agent.
<code>snmpcmdchk</code>	Checks the installation status of the OS commands needed by SNMP Agent to generate MIB values.
<code>snmpstart</code> [#]	Starts SNMP Agent.
<code>snmpstop</code> [#]	Stops SNMP Agent.
<code>snmptrap</code>	Issues an SNMP trap.
<code>systemtrap</code>	Issues a system trap.
<code>trapsend</code>	Issues an SNMP trap.

[#]: Execute this command as a superuser.

Details of commands

This section explains the SNMP Agent commands in accordance with the common format consisting of the next items. There is, however, no assurance that all items exist.

Syntax

Includes a diagram that summarizes the syntax of the command.

Description

Provides a brief description of the function of the command.

Location

Lists the directory in which the command is stored.

Arguments

Lists and explains the command's arguments, if any.

Who can execute the command

Describes who is authorized to execute the command.

Return values

Describes the return values of the command.

Customizable items

Describes the items that can be customized using the command (shell script).

Execution example

Gives an example of executing the command.

Notes

Provides notes on using the command.

The rest of this section explains SNMP Agent commands in alphabetical order.

jp1esalog.sh.def

Syntax

```
jp1esalog.sh.def
```

Description

The `jp1esalog.sh.def` command is a command to be executed immediately after occurrence of an error.

First, this command uses the `tar` command to archive the directories or files that contain log information immediately under the root directory. Then, this command uses the `compress` command to compress the created archive file. By default, this command creates the `/tmp/jplesa/jplesa.log.tar.Z` file. If this file already exists, the existing file is overwritten.

You can execute this command while SNMP Agent is running.

Location

- Systems other than AIX: `/opt/CM2/ESA/bin`
- AIX: `/usr/CM2/ESA/bin`

Arguments

None

Who can execute the command

A superuser can execute this command.

Return values

0: Normal termination

8: One of the following runtime errors occurred (an error message is output):

- The command was not executed as a superuser.
- The work directory could not be created.
- The archive file could not be created.
- The directory for outputting files did not allow writes.
- The directory for output files could not be created.

Customizable items

This command is a shell script. The following describes what you can customize when using this command.

Changing the default output file

To change the default output file, customize the following line:

```
# Log output file defaultname
OUTFILE="/tmp/jplesa/jplesa.log"
```

When a file name is specified, the following file is created:

```
specified-file-name.Z
```

Changing the work directory

To change the work directory, customize the following line:

```
# Working directory
WORKDIR="/tmp/jplesa/work"
```

When this command is executed, it creates a temporary work directory in which it temporarily saves the files it creates. This means that the command requires a certain amount of free disk space to operate. By default, this directory is created in the directory that is used to store the output files. Edit the above line if you want to change the location of the work directory.

Collecting other information such as user-created files

To collect other information, customize the following line:

```
# User Additional files
ADDFILE=""
```

Note that when you specify additional files in the `ADDFILE` variable, always prefix a period (.) to the full path name of each file, and separate each entry with a space.

Example: `ADDFILE="./var/tmp/user_log ./etc/opt/sample/conf"`

Notes

- This command is provided as the `jplesalog.sh.def` file. This file is overwritten with a new file every time the product is installed. To retain the file you customized, copy the file to any directory you like, and customize and use the file there.
- When transferring the collected information via FTP, use the binary mode.
- When the `compress` command is not installed in the machine where information is to be collected, this command creates the `/tmp/jplesa/jplesa.log.tar` file without performing compression.
- The following table lists the data collected by the `jplesalog.sh.def` command.

Table 5–2: List of data collected by the `jplesalog.sh.def` command

Type	Collected information
Common information	<code>/etc/hosts</code>
	<code>/etc/services</code>
	<code>/etc/nsswitch.conf</code> (Linux)
	<code>/etc/resolv.conf</code> (Linux, Solaris, AIX)
	<code>/etc/redhat-release</code> (Linux)
	<code>/etc/environment</code> (AIX)
	<code>/proc/stat</code> (Linux)
	<code>/proc/diskstats</code> (Linux)
	Mount configuration files <ul style="list-style-type: none">• HP-UX (IPF) and Solaris <code>/etc/mnttab</code>• AIX <code>/etc/filesystems</code>

Type	Collected information
	<ul style="list-style-type: none"> Linux /etc/fstab
	System log files <ul style="list-style-type: none"> HP-UX (IPF) /var/adm/syslog/syslog.log /var/adm/syslog/OLDSyslo.log Solaris /var/adm/messages Linux /var/log/messages The results of journalctl (RHEL 7, CentOS 7, Oracle Linux 7, SUSE Linux 12)
	The results of ps -ef (for Solaris, the results of ps -lf -z zone-name)
	The results of ps -e (AIX only)
	The results of netstat (with -a, -i, -rv, and -an) The result of netstat -anp (Linux)
	The result of ss -anp (Linux)
	The result of ifconfig -a (Linux)
	The result of ip addr (Linux)
	The result of lsb_release -a (Linux)
	The results of uname -a
	The results of oslevel -s (AIX)
	The results of hostname
	The results of id
	The results of env
	The results of bdf -l or df -k
	/etc/.hitachi/pplstd/pplstd
	The results of swlist -l product (HP-UX (IPF))
	The results of what /usr/sbin/snmpdm (HP-UX (IPF))
	The results of what /usr/sbin/mib2agt (HP-UX (IPF))
	The results of what /usr/sbin/ipv6agt (HP-UX (IPF))
	The results of what /usr/sbin/hp_unixagt (HP-UX (IPF))
	The result of ls -alR /tmp
	The result of ls -alR /opt/CM2/ESA (except AIX)
	The result of ls -alR /usr/CM2/ESA (AIX)
	The result of ls -alR /etc
	The result of rpm -qa (Linux)
	The results of systemctl show jp1_esa.service (RHEL 7, CentOS 7, Oracle Linux 7, SUSE Linux 12)

Type	Collected information
	The results of <code>systemctl list-dependencies jpl_esa.service --after</code> (RHEL 7, CentOS 7, Oracle Linux 7, SUSE Linux 12)
	The results of <code>systemctl list-dependencies jpl_esa.service --before</code> (RHEL 7, CentOS 7, Oracle Linux 7, SUSE Linux 12)
	The results of <code>systemctl list-unit-files</code> (RHEL 7, CentOS 7, Oracle Linux 7, SUSE Linux 12)
	The results of <code>systemctl status jpl_esa.service</code> (RHEL 7, CentOS 7, Oracle Linux 7, SUSE Linux 12)
	<p>Configuration files and log files for the native agent</p> <p>Solaris:</p> <ul style="list-style-type: none"> • /etc/release • /var/log/snmpd.log • /var/opt/CM2/ESA/log/initdesa.log.err • /var/opt/CM2/ESA/log/snmpstart.log.err • /var/svc/log/milestone-multi-user:default.log • /etc/net-snmp/snmp/snmpd.conf • /lib/svc/method/svc-net-snmp • /var/svc/log/application-management-net-snmp:default.log • The results of <code>/usr/bin/svcs -l svc:/application/management/net-snmp:default</code> <p>AIX:</p> <ul style="list-style-type: none"> • /etc/snmpd.conf • /usr/tmp/snmpd.log • /etc/snmpdv3.conf • /usr/tmp/snmpdv3.log • /etc/aixmibd.conf • /usr/tmp/aixmibd.log • /etc/hostmibd.conf • /usr/tmp/hostmibd.log • /etc/snmpmibd.conf • /usr/tmp/snmpmibd.log <p>Linux:</p> <ul style="list-style-type: none"> • /etc/snmp/snmpd.conf • /var/log/snmpd.log <p>For RHEL 7, CentOS 7, Oracle Linux 7, and SUSE Linux 12, the following additional files are collected:</p> <ul style="list-style-type: none"> • /usr/lib/systemd/system/snmpd.service • /etc/systemd/system/snmpd.service (if present)
SNMP Agent information	<p>The following files under /var/adm/:</p> <ul style="list-style-type: none"> • snmpd.log^{#1, #2} <p><i>n</i>: Number of log files snmpd.logⁿ files under SR_LOG_DIR if SR_LOG_DIR is set.</p>
	/tmp/jplesa/work/jplesalog_err.log
	<ul style="list-style-type: none"> • /var/opt/CM2/ESA/log/htc_monagt1.log (Solaris, Linux) • /usr/CM2/ESA/log/htc_monagt1.log (AIX)
	The results of <code>/opt/CM2/ESA/bin/snmpcheck</code>
	Data under <code>/etc/SnmpAgent.d</code>
	Data under <code>/etc/srconf/agt</code>

Type	Collected information
	Data under <code>/etc/srconf/mgr</code>
	<ul style="list-style-type: none"> • Data under <code>/opt/CM2/ESA/opt</code> (HP-UX (IPF), Linux) • Files whose names begin with <code>/etc/rc.config.d/Snmp</code> (Solaris) • Data under <code>/usr/CM2/ESA/opt</code> (AIX only)
	<ul style="list-style-type: none"> • Data under <code>/opt/CM2/ESA/ext</code> (systems other than AIX) • Data under <code>/usr/CM2/ESA/ext</code> (AIX only)
	RHEL 7, CentOS 7, Oracle Linux 7, and SUSE Linux 12: <ul style="list-style-type: none"> • <code>/usr/lib/systemd/system/jpl_esa.service</code> • <code>/etc/systemd/system/jpl_esa.service</code> (if present)
	<code>/core</code> (if present)
	<code>/root/core</code> (for Linux and only if present)

#1: Log information, hexadecimal dumps, and VarBind lists are output to the `snmpd.logn` files. The size and number of `snmpd.logn` files are specified with the following environment values in the `SnmpMaster` file:

- File size: `SNMP_HTC_SNMPD_LOG_SIZE`

The following shows a specification example for the `SNMP_HTC_SNMPD_LOG_SIZE` environment variable. The unit is megabytes. In the example, 10 megabytes is specified as the file size.

Example:

```
SNMP_HTC_SNMPD_LOG_SIZE=10
export SNMP_HTC_SNMPD_LOG_SIZE
```

- Number of files: `SNMP_HTC_SNMPD_LOG_CNT`

The following shows a specification example for the `SNMP_HTC_SNMPD_LOG_CNT` environment variable. The unit is the number of files. In the example, 10 is specified as the number of files.

Example:

```
SNMP_HTC_SNMPD_LOG_CNT=10
export SNMP_HTC_SNMPD_LOG_CNT
```

For details about environment variables, see [snmpdm](#).

By default, the size of each file is set to 10 megabytes, which means that in order to create 10 files, an area that can accommodate 100 megabytes must be available for the directory in which the `snmpd.logn` files are stored.

#2: The following processes send their output to the `snmpd.logn` files:

```
snmpdm, naaagt, hp_unixagt, extsubagt, trapdestagt, htc_unixagt1, htc_unixagt2,
htc_unixagt3, htc_unixagt4
```

snmpcheck

Syntax

```
snmpcheck
```

Description

The `snmpcheck` command displays the operating status (running or not running) of the master agent, subagents, and native agent.

Output Example

The following example shows the execution of the `snmpcheck` command in Solaris 11.

```
#/opt/CM2/ESA/bin/snmpcheck
snmpdm  running pid=15128
hp_unixagt      running pid=15170
trapdestagt    running pid=15189
extsubagt      not running
htc_unixagt1   running pid=15209
htc_unixagt3   running pid=15229
htc_monagt1    running pid=15248
htc_unixagt4   running pid=15250
naaagt  running pid=15151
snmpd  running pid=15132
```

running indicates that the process is active.

not running indicates that the process is inactive.

pid indicates the process ID.

Location

- Systems other than AIX: `/opt/CM2/ESA/bin`
- AIX: `/usr/CM2/ESA/bin`

Arguments

None

snmpcmdchk

Syntax

```
snmpcmdchk
```

Description

The `snmpcmdchk` command checks whether the OS commands needed by SNMP Agent to generate MIB values are installed. Execute this command after you install SNMP Agent, and then install any command that is not installed.

Specification Example

The following example shows the execution of the `snmpcmdchk` command in Solaris 11.

```
#/opt/CM2/ESA/bin/snmpcmdchk
/usr/sbin/prtconf      installed.
/usr/bin/sar           installed.
/usr/sbin/swap         installed.
/usr/bin/pagesize      installed.
/usr/bin/mpstat        Not installed.
```

`installed` indicates that the applicable command is installed.

`Not installed` indicates that the applicable command is not installed.

Location

- Systems other than AIX: `/opt/CM2/ESA/bin`
- AIX: `/usr/CM2/ESA/bin`

Arguments

None.

snmpstart

Syntax

```
snmpstart [-e] [-n]
```

Description

The `snmpstart` command starts SNMP Agent. If SNMP Agent is running when the `snmpstart` command is executed, the `snmpstart` command stops and then restarts SNMP Agent.

Location

- Systems other than AIX: `/opt/CM2/ESA/bin`
- AIX: `/usr/CM2/ESA/bin`

Arguments

`-e`

Specify this option when you want to load the extended MIB definition files that are placed under the `/opt/CM2/ESA/ext` directory and have not been loaded yet.

`-n`

Only the SNMP Agent process starts or restarts. The native agent process does not start or restart. This argument can be used in AIX and Solaris.

None

If no options are specified, the behavior is as follows.

For HP-UX (IPF), Solaris, and AIX

In addition to the SNMP Agent process, the native agent process also starts or restarts.

For Linux

The `snmpstart` command starts (or restarts) only SNMP agent processes, and does not start (or restart) native agent processes.

Who can execute the command

A superuser can execute this command.

snmpstop

Syntax

```
snmpstop [-n]
```

Description

The `snmpstop` command stops SNMP Agent.

Location

- Systems other than AIX: `/opt/CM2/ESA/bin`
- AIX: `/usr/CM2/ESA/bin`

Arguments

`-n`

Only the SNMP Agent process stops. The native agent process does not stop. This argument can be used in AIX and Solaris.

None

If no options are specified, the behavior is as follows.

For HP-UX (IPF), Solaris, and AIX

In addition to the SNMP Agent process, the native agent process also stops.

For Linux

The `snmpstop` command stops only SNMP agent processes, and does not stop native agent processes.

Who can execute the command

A superuser can execute this command.

snmptrap

Syntax

```
snmptrap [-d] [-p port-number] [-c community-name] node-name  
         enterprise-ID agent-address standard-trap-number  
         enterprise-specific-trap-number time-stamp  
         [object-identifier value-type value...]
```

Description

The `snmptrap` command issues an SNMP trap to a specified node.

Location

- Systems other than AIX: `/opt/OV/bin`
- AIX: `/usr/OV/bin`

Arguments

`-d`

Specify this option if you want to output the SNMP packets to the standard output in the hexadecimal format and decoded ASN.1 format.

`-p` *port-number*

Specify the port number of the sending manager. If omitted, the value 162 is assumed.

`-c` *community-name*

Specify the community name. If omitted, the value `public` is assumed.

node-name

Specify the IP address or host name of the destination node.

enterprise-ID

Specify a `sysObjectID` in the `A.B.C.D...` format, where `A`, `B`, `C`, and `D` are subidentifiers in decimal notation. If the object identifier is in the `A.B.C.D...` format and begins with `1.3.6.1.2.1`, you can omit `1.3.6.1.2.1`. If you specify a NULL string (`' '`), the command assumes SNMP Agent's `sysObjectID`. For details about the `sysObjectID` function of SNMP Agent, see [4.3.1 Organization of Hitachi enterprise-specific MIB objects](#).

agent-address

Specify the IP address or the host name of the agent. If you specify a NULL string (`' '`), the command applies the value acquired by using the appropriate OS functions to first acquire the host name, and then to convert the acquired host name to its IP address.

If you wish to use a particular IP address as the agent address that is output in trap messages issued by the `snmptrap` command, specify that IP address as the agent address.

standard-trap-number

Specify a standard trap number as an integer from 0 to 6. Specify 6 to issue an enterprise-specific trap.

enterprise-specific-trap-number

Specify an enterprise-specific trap number as a 32-bit integer. If the standard trap number is not set to 6, this number is ignored, and the argument is filled with 0s. Valid values are positive integers, negative integers, hexadecimal integers (beginning with `0x`), and octet integers (beginning with `0`).

time-stamp

Specify a time as an integer of value 0 or greater. If you specify a NULL string (' ') the command assumes the value in `timeticks`, which is the number of ticks counted since the system started.

object-identifier

For the `snmptrap` command, you can specify more than one *object-identifier value-type value* tuple. For example, if one tuple is 256 bytes in length, you can specify up to 20 tuples. *object-identifier* must be in the *A.B.C.D...* format, where *A*, *B*, and *C* are subidentifiers in decimal notation. If the object identifier is in the *A.B.C.D...* format and begins with `1.3.6.1.2.1`, you can omit `1.3.6.1.2.1`.

value-type

Specify one of the following values:

- `integer` (from -2^{31} to $2^{31}-1$)
- `octetstring`
- `objectidentifier` (if the object identifier is in the *A.B.C.D...* format and begins with `1.3.6.1.2.1`, you can omit `1.3.6.1.2.1`)
- `null` (the command unconditionally ignores the *value* argument following `null`)
- `ipaddress`
- `counter` (from 0 to 4294967295)
- `gauge`
- `timeticks`
- `opaque`

For details, see RFC 1155.

You can also specify the following special `octetstring` values:

- `octetstringhex` (string of hexadecimal pairs from 00 to FF; example: `01FF`)
- `octetstringoctal` (string of octal triples from 000 to 377; example: `001377`)
- `octetstringascii` (ASCII character string)

You can also specify the following special `opaque` values:

- `opaquehex` (string of hexadecimal pairs from 00 to FF; example: `01FF`)
- `opaqueoctal` (string of octal triples from 000 to 377; example: `001377`)
- `opaqueascii` (ASCII character string)

value

Specify a value of the specified value type.

Return values

0: Normal termination

Since SNMP traps are transmitted via UDP, whether the transmission succeeded is not checked. Thus, the remote node might not be notified even when the command terminated normally.

1: Run-time error

An error message is output.

Note

In AIX, if the `snmptrap` command is executed as an extension of a shell script or program started from `cron` or `/etc/inittab`, the command might fail with the following message:

```
snmptrap:cannot set locale($LANG="Ja_JP")
```

If this message is output, in the `LC_ALL` environment variable, set the language you want to use.

The following shows an example of setting `C` as the language code for the `B` shell.

```
LC_ALL=C
export LC_ALL
snmptrap flcndmak .1.3.6.1.4.1.4242 15.6.71.223 6 2 0
```

systemtrap

Syntax

```
systemtrap
```

Description

The `systemtrap` command issues a system trap to a specified node.

The `systemtrap` command is a monitoring program that is provided for backward compatibility when using JP1/SSO. For details about the `systemtrap` command, see the manual of JP1/SSO.

trapsend

Syntax

The syntax for sending an SNMPv1 trap is:

```
trapsend -v1 [-ipv6] [-port port-number]  
node-name community-name standard-trap-number  
[enterprise-specific-trap-number enterprise-ID  
  [time-stamp  
    [object-identifier value-type value]  
    [object-identifier value-type value]  
    ...  
  ]  
]
```

The syntax for sending an SNMPv2c trap is:

```
trapsend -v2c [-ipv6] [-port port-number]  
node-name community-name trap-OID  
[time-stamp  
  [object-identifier value-type value]  
  [object-identifier value-type value]  
  ...  
]
```

Description

The `trapsend` command sends an SNMPv1 trap or SNMPv2c trap to the specified IPv4 node or IPv6 node.

Location

- Systems other than AIX: `/opt/CM2/ESA/bin`
- AIX: `/usr/CM2/ESA/bin`

Arguments

- `-v1`
Send an SNMPv1 trap.
- `-v2c`
Send an SNMPv2c trap.
- `-ipv6`
Send the trap to an IPv6 node. If omitted, the trap is sent to an IPv4 node.
- `-port port-number`
Specify the trap destination port number as an integer from 1 to 65535. If omitted, 162 is assumed. If an out-of-range numeric value or a non-numeric value is specified, a message is output, and processing continues assuming a value of 162.
- `node-name`
Specify an IP address or host name.

When `-ipv6` is specified, specify an IPv6 node (an IPv6 address or host name that can be resolved to an IPv6 address), otherwise specify an IPv4 node (an IPv4 address or host name that can be resolved to an IPv4 address).

community-name

Specify the community name.

standard-trap-number

Specify a standard trap number as an integer from 0 to 6. Specify 6 to issue an enterprise-specific trap.

If a non-numeric value is entered, processing continues using 0 (`coldStart`).

enterprise-specific-trap-number

Specify an enterprise-specific trap number as a 32-bit signed integer in decimal notation. This cannot be omitted when issuing an enterprise-specific trap. If a non-numeric value is specified, a message is output and processing continues using the value 0.

enterprise-ID

Specify a `sysObjectID`. This cannot be omitted when issuing an enterprise-specific trap.

The input format follows the numeric object identifier notation (see below). If a format other than this is specified, an error message is output and the program terminates.

If this argument is omitted, the following `sysObjectID` is assumed:

OS	sysObjectID
HP-UX (IPF)	1.3.6.1.4.1.116.3.9.1.3
Solaris	1.3.6.1.4.1.116.3.8.1.3
AIX	1.3.6.1.4.1.116.3.13.1.3
Linux	1.3.6.1.4.1.116.3.14.1.3

trap-OID

Specify the trap OID. The specified value follows the numeric object identifier notation (see below).

time-stamp

Specify a time as an integer of value 0 or greater. If omitted, the value 0 is used. If a non-numeric value is entered, a message is output and processing continues using the value 0.

object-identifier, value-type, value

The `trapsend` command permits the specification of more than one *object-identifier value-type value* tuple.

object-identifier

The format of the value follows the numeric object identifier notation (see below).

value-type

Select from the following types:

Specified value	Type	Description
<code>-i</code>	INTEGER	Specifies a 32-bit signed integer value as a decimal or hexadecimal (0x . . .) string.
<code>-o</code>	OCTET STRING	Specifies pairs of hexadecimal numbers separated by a colon or space (Examples: "0A B1 B3", "00:BB:F0").
<code>-d</code>	OBJECT IDENTIFIER	Format follows the numeric object identifier notation (see below).
<code>-a</code>	IpAddress	IPv4 address format
<code>-c</code>	Counter32	32-bit unsigned integer value

Specified value	Type	Description
-g	Gauge32	32-bit unsigned integer value
-t	TimeTicks	32-bit unsigned integer value
-D	OCTET STRING	Character string of NVT ASCII code set values
-N	NULL	Specifies a numeric value. However, the value that is entered is ignored.

value

Specify a value of the correct value type.

Numeric object identifier notation

In the `trapsend` command, when an object identifier is specified as a number, it must conform to the following format:

```
0-2.0-39 [.0-4294967295 [.0-4294967295 . . . ]]
```

Return values

0: Normal termination

Because the SNMP trap is sent over UDP, delivery to the destination node is not confirmed. As a result, the destination node might not be notified even in the case of normal termination. Also, even if a message is output to the standard output or standard error when the `trapsend` command is executed, if the return value is 0 then the SNMP trap is sent.

Other than 0: Runtime error

The SNMP trap is not sent.

Execution examples

These examples show transmission of SNMPv1 traps.

SNMPv1 trap transmission example 1

In this example, a `coldStart` trap (standard trap number 0) is sent to the node with IPv4 address `192.168.1.5` using the community name `sendtrap`.

```
trapsend -v1 192.168.1.5 sendtrap 0
```

SNMPv1 trap transmission example 2

In this example, an `enterpriseSpecific` trap (standard trap number 6) and enterprise-specific trap number 5, with an enterprise ID of `1.3.6.1.4.1.116.3.14.1.3`, is sent to port 22162 of the node with IPv6 address `fec0::1111:2222:3333:4444:5555` using the community name `sendtrap`.

```
trapsend -v1 -port 22162 -ipv6 fec0::1111:2222:3333:4444:5555 sendtrap 6
5 1.3.6.1.4.1.116.3.14.1.3
```

SNMPv1 trap transmission example 3

In this example, a `linkDown` trap (standard trap number 2) with enterprise ID `1.3.6.1.4.1.116.3.14.1.3` is sent to the hostname `esaipv6` (which can be resolved to an IPv6 address) using the community name `sendtrap`. The time stamp is 10000. One *object-identifier value-type value* tuple is assigned (object identifier: `1.3.6.1.2.1.2.2.1.1.3`, value type: Integer, value: 3).

```
trapsend -v1 -ipv6 esaipv6 sendtrap 2 0 1.3.6.1.4.1.116.3.14.1.3 10000
1.3.6.1.2.1.2.2.1.1.3 -i 3
```


These examples illustrate transmission of SNMPv2c traps.

SNMPv2c trap transmission example 1

In this example, a `warmStart` trap (object identifier: 1.3.6.1.6.3.1.1.5.2) is sent to the node with IPv4 address 192.168.1.5 using the community name `sendtrap`.

```
trapsend -v2c 192.168.1.5 sendtrap 1.3.6.1.6.3.1.1.5.2
```

SNMPv2c trap transmission example 2

In this example, a `linkUp` trap (object identifier: 1.3.6.1.6.3.1.1.5.4) is sent to the node with IPv6 address `fec0::1111:2222:3333:4444:5555` with the time stamp 79000 and the community name `sendtrap`. One *object-identifier value-type value* tuple is assigned (object identifier: 1.3.6.1.2.1.2.2.1.1.3, value type: Integer, value: 3).

```
trapsend -v2c -ipv6 fec0::1111:2222:3333:4444:5555 sendtrap  
1.3.6.1.6.3.1.1.5.4 79000 1.3.6.1.2.1.2.2.1.1.3 -i 3
```

Notes

- Unlike the `snmptrap` command, you cannot specify the `Agent Address` field of SNMPv1 trap packets.
- If the trap destination node is an IPv4 node, the IP address of the agent system (the value of the host name obtained by the OS functions of the relevant system and converted to an IP address using OS functions) is entered in the `Agent Address` field.
- If the trap destination node is an IPv6 node, `0.0.0.0` is entered in the `Agent Address` field. However, if the destination node is the IPv6 loopback address (`::1`), `127.0.0.1` is entered. If the trap destination node is an IPv4-mapped address, the IPv4 address of the agent system is entered.

Processes

This section explains the functions and startup options of the SNMP Agent processes. It also describes the environment variables used by these processes.

List of processes

SNMP Agent consists of a master agent and subagents. It also includes an information collection daemon for collecting information from the OS.

The following tables list the processes used in SNMP Agent operations.

- Process used in master agent operations

Table 5–3: SNMP Agent process (master agent)

Processes name	Description
snmpdm	Handles communications for the master agent.

- Processes used in subagent operations and the information collection daemon

Table 5–4: SNMP Agent processes (subagents and the information collection daemon)

Processes name	Description
extsubagt	Subagent that provides extended MIB objects.
hp_unixagt	Subagent that provides Hewlett-Packard enterprise-specific MIB objects.
htc_monagt1	Daemon process that periodically collects information on the CPU utilization rate, CPU utilization time, and disk busy time.
htc_unixagt1	Subagent that provides Hitachi enterprise-specific MIB objects.
htc_unixagt2	Subagent that provides Hitachi enterprise-specific MIB objects.
htc_unixagt3	Subagent that provides Hitachi enterprise-specific MIB objects.
naaagt	Subagent that provides the native language adapter function.
trapdestagt	Subagent that provides Hewlett-Packard enterprise-specific MIB objects (trap group).

Detailed process descriptions

The following process description sections use an organization consisting of the subsections indicated here to describe the SNMP Agent processes. Sections do not necessarily include all subsections.

Syntax

Provides the startup format of the process.

Description

Describes the functioning of the process.

Location

Lists the directory in which the process is stored.

Arguments

Explains the process's startup options.

External influences

Describes the applicable environment variables.

Notes

Provides notes about the process.

snmpdm

Syntax

```
snmpdm [-authfail] [-Contact system-contact] [-help]
        [-Location system-location]
        [-mask logmask] [-n] [-tcplocal]
        [-ip_proto [ipv4 | ipv4_ipv6 | ipv6]]
        [-sysDescr description] [-hexdump] [-vbdump]
```

Description

The `snmpdm` process handles communications for the master agent. The master agent listens for SNMP requests from the SNMP manager, and sends each message it receives to a subagent. The subagent returns a response to the master agent, and the master agent passes this response back to the SNMP manager.

The master agent provides the following MIB groups:

- System group (`internet.mgmt.mib-2.system`)
- SNMP group (`internet.mgmt.mib-2.snmp`)
- `snmpdConf` group (`enterprises.hp.nm.snmp.snmpdConf`)

Location

- HP-UX (IPF), Solaris: `/opt/CM2/ESA/bin`
- AIX, Linux: `/usr/sbin`

Arguments

`-authfail`

Abbreviation of `-authfail`: `-a`

This option suppresses the sending of authentication failure traps from the master agent.

Usually do not use this option because it is provided for compatibility with SNMP Agents of the previous version.

To inhibit authentication failure traps, specify 2 for `snmpEnableAuthenTraps` in `/etc/srconf/agt/snmpd.cnf` and reactivate the master agent.

`-Contact system-contact`

Abbreviation of `-Contact`: `-C`

This option sets *system-contact* as the system contact of the master agent.

`-help`

Abbreviation of `-help`: `-h`

This option displays information about the command options of the master agent.

`-Location system-location`

Abbreviation of `-Location`: `-L`

This option sets *system-location* as the system location of the master agent.

`-logfile log-file-name`

Abbreviation of `-logfile`: `-l`

This option sets *log-file-name* as the name of the logfile for master agent logs.

`-mask logmask`

Abbreviation of `-mask`: `-m`

This option sets the master agent's logmask value to *logmask*. The logmask value can be a character string, decimal number, or hexadecimal number. The table below lists logmask values, followed by coding samples.

Type of logmask value	Log suppression	Trace log output	Warning log output	Error log output
Character string	-	FACTORY_TRACE	FACTORY_WARN	FACTORY_ERROR
Decimal number	0	8388608	268435456	536870912
Hexadecimal number	0x0	0x00800000	0x10000000	0x20000000

Legend:

-: Not applicable

Examples:

```
snmpdm -m FACTORY_TRACE
snmpdm -m 8388608
snmpdm -m 0x00800000
snmpdm -m FACTORY_TRACE FACTORY_WARN FACTORY_ERROR
```

If you specify the `-m` option with a character string in combination with other options, note that the `-m` option is the last option specified.

`-n`

This option specifies that the master agent does not act as a daemon.

`-tcplocal`

This option enables acceptance of TCP connections from the subagents.

`-ip_proto [ipv4 | ipv4_ipv6 | ipv6]`

Specifies the IP version of the SNMP request reception port. If `-ip_proto` is not specified, an IPv4 and IPv6 (`ipv4_ipv6`) SNMP request reception port is used.

`ipv4`

Only an IPv4 SNMP request reception port is used.

`ipv4_ipv6`

Both IPv4 and IPv6 SNMP request reception ports are used.

`ipv6`

Only an IPv6 SNMP request reception port is used.

`-sysDescr description`

Abbreviation of `-sysDescr`: `-sys`

This option sets *description* as the description of the master agent.

`-hexdump`

Displays a hexadecimal dump of the contents of an SNMP packet. For details about how to use this argument, see [7.4.1 Acquiring a master agent send/receive packet dump](#).

`-vbdump`

This option displays the contents of the VarBind lists in the SNMP packets. For details about how to use this option, see [7.4.1 Acquiring a master agent send/receive packet dump](#).

`-apverbose`

This option outputs a verbose log.

External influences

Environment variables

Specify the following environment variables in the `SnmpMaster` file.

`SR_SNMP_TEST_PORT`

This environment variable sets the master agent's SNMP reception port. If this environment variable is not specified, the master agent uses the value set in the `snmp` line of the `/etc/services` file as the SNMP reception port. Normally, you do not need to specify this environment variable. You must specify this environment variable only if you want to change the master agent's SNMP reception port.

`SNMP_HTC_AUTH_LOG`

This environment variable specifies whether unauthorized community names are collected.

For details about how to set the `SNMP_HTC_AUTH_LOG` environment variable, see [7.4.4 Collecting logs of unauthorized community names](#).

`SR_TRAP_TEST_PORT`

This environment variable sets the master agent's SNMP trap notification port.

`SNMP_HTC_INIT_WAIT_TIME`

This environment variable specifies a value from 0 to 300 (in seconds) as the time period from when the `snmpdm` process completes startup processing until a `coldStart` trap is sent. SNMP requests received during this period are discarded. The default is 15 seconds.

`SNMP_HTC_SNMPD_LOG_SIZE`

This environment variable specifies the size of the output files for logs, hexadecimal dumps, and VarBind lists as a value from 0 to 50 (in megabytes). If 0 is specified, logs, hexadecimal dumps, and VarBind lists are not output. The default is 10 megabytes.

`SNMP_HTC_SNMPD_LOG_CNT`

This environment variable specifies the number of output files for logs, hexadecimal dumps, and VarBind lists as a value from 1 to 10. The default is 10 files.

`SR_LOG_DIR`

This environment variable specifies the output directory for logs, hexadecimal dumps, and VarBind lists.

Note

- Specifying a value of less than 15 seconds as the time set in the `SNMP_HTC_INIT_WAIT_TIME` environment variable increases the probability that a `noSuchName` error will be returned in response to a request from the SNMP manager.

extsubagt

Syntax

```
extsubagt [-e extended-MIB-definition-file] [-p] [-E priority]  
          [-aperror] [-apwarn] [-aptrace] [-apconfig]  
          [-appacket] [-aptrap] [-apaccess] [-apemanate]  
          [-apverbose] [-apuser] [-apall] [-help] [-retry N]  
          [-fcmdguard N] [-pipeguard N] [-invokeid]
```

Description

The `extsubagt` process provides extended MIB objects.

Location

- HP-UX (IPF), Solaris: `/opt/CM2/ESA/bin`
- AIX, Linux: `/usr/sbin`

Arguments

`-e extended-MIB-definition-file`

Specify an extended MIB definition file. The default definition file is `/etc/SnmpAgent.d/snmpd.extend`.

`-p`

Specify this option if you want to check the syntax of the extended MIB definition file.

`-E priority`

Specify the priority of the subagent.

`-aperror`

Specify this option if you want to collect error logs.

`-apwarn`

Specify this option if you want to collect error and warning logs.

`-aptrace`

Specify this option if you want to collect trace logs.

`-apconfig`

Specify this option if you want to collect logs related to the configuration file.

`-appacket`

Specify this option if you want to collect logs related to packet assembly and analysis.

`-aptrap`

Specify this option if you want to collect logs related to trap messages.

`-apaccess`

Specify this option if you want to collect logs related to agent processing.

`-apemanate`

Specify this option if you want to collect logs related to the master agent and subagents.

`-apverbose`

Specify this option if you want to collect verbose logs.

-apuser

Specify this option if you want to collect user logs.

-apall

Specify this option if you want to collect all types of logs.

-help

Specify this option if you want to display the command syntax.

-retry *N*

Specify this option if you want the subagent to attempt to establish a connection with the master agent at *N*-second intervals.

-fcmdguard *N*

Specify the *file_command* execution response monitoring time in seconds. The specified value *N* must be $1 \leq N \leq 90$.

-pipeguard *N*

Specify the monitoring period in seconds from the time SNMP Agent writes data into *pipe_out_name* to the time processing results are written. The specified value *N* must be $1 \leq N \leq 90$.

-invokeid

Specify this option to add an identification number as the first argument written to *pipe_out_name*. The identification number is in the form of *xxxxxxx.yyyyyy*, where *xxxxxxx* indicates the number of elapsed seconds and *yyyyyy* indicates the fraction of the current second in microseconds.

You can configure settings so that `-fcmdguard`, `-pipeguard`, and `-invokeid` are enabled at startup or when the `snmpstart` command is executed. For details, see *Environment variable definition file (SnmpExtAgt)* in [Chapter 6. Definition Files](#).

External influences

Environment variables

Specify the following environment variable in the `SnmpExtagt` file.

`SR_SNMP_TEST_PORT`

This environment variable sets the master agent's SNMP reception port. If this environment variable is specified for a subagent, it is used as data for connecting to the master agent. This means that the value set for the subagent must be the same as the port number specified for the master agent. If this environment variable is not specified, the value in the `snmp` line of the `/etc/services` file is used. Normally, you do not need to specify this environment variable. You must specify this environment variable only if you want to change the master agent's SNMP reception port.

hp_unixagt

Syntax

```
hp_unixagt [-aperror] [-apwarn] [-aptrace] [-apconfig]
           [-appacket] [-aptrap] [-apaccess] [-apemanate]
           [-apverbose] [-apuser] [-apall] [-help]
           [-retry N]
```

Description

The `hp_unixagt` process provides the following Hewlett-Packard enterprise-specific MIB groups:

- `computerSystem` group (`enterprises.hp.nm.system.general.computerSystem`)
- `fileSystem` group (`enterprises.hp.nm.system.general.fileSystem`)
- `processes` group (`enterprises.hp.nm.system.general.processes`)
- `icmp` group (`enterprises.hp.nm.icmp`)

Location

- HP-UX (IPF), AIX, Linux: `/usr/sbin`
- Solaris: `/opt/CM2/ESA/bin`

Arguments

`-aperror`

Specify this option if you want to collect error logs.

`-apwarn`

Specify this option if you want to collect error and warning logs.

`-aptrace`

Specify this option if you want to collect trace logs.

`-apconfig`

Specify this option if you want to collect logs related to the configuration file.

`-appacket`

Specify this option if you want to collect logs related to packet assembly and analysis.

`-aptrap`

Specify this option if you want to collect logs related to trap messages.

`-apaccess`

Specify this option if you want to collect logs related to agent processing.

`-apemanate`

Specify this option if you want to collect logs related to the master agent and subagents.

`-apverbose`

Specify this option if you want to collect verbose logs.

`-apuser`

Specify this option if you want to collect user logs.

-apall

Specify this option if you want to collect all types of logs.

-help

Specify this option if you want to display the command syntax.

-retry *N*

Specify this option if you want the subagent to attempt to establish a connection with the master agent at *N*-second intervals.

External influences

Environment variables

Specify the following environment variables in the `SnmpHpunix` file.

`SR_SNMP_TEST_PORT`

This environment variable sets the master agent's SNMP reception port. If this environment variable is specified for a subagent, it is used as data for connecting to the master agent. This means that the value set for the subagent must be the same as the port number specified for the master agent. If this environment variable is not specified, the value in the `snmp` line of the `/etc/services` file is used. Normally, you do not need to specify this environment variable. You must specify this environment variable only if you want to change the master agent's SNMP reception port.

`SNMP_HTC_SOLARIS_SWAP_RESERVED` (Solaris)

This environment variable specifies whether the size of the reserved space is included as part of the device swap space size.

Y: The size of the reserved space is included.

Value other than *Y*: The size of the reserved space is not included.

The default is a value other than *Y*.

`SNMP_HTC_AIX_EXCEPT_FILECACHE` (AIX)

This environment variable specifies whether the file cache is excluded from the amount of physical memory currently in use.

Y: The file cache is excluded from the amount of the physical memory currently in use.

Value other than *Y*: The file cache is not excluded from the amount of physical memory currently in use.

The default is a value other than *Y*.

`SNMP_HTC_LINUX_INACTIVE_MEM` (Linux)

This environment variable specifies how to calculate the MIB value for the amount of free memory.

Y: The sum of the amount of free memory, inactive buffer memory, and inactive cache memory.

Value other than *Y*: The sum of the amount of free memory, buffer memory, and cache memory.

The default is a value other than *Y*.

htc_monagt1

Syntax

```
htc_monagt1 [-i CPU-utilization-acquisition-interval-time]
[-s CPU-utilization-time-information-acquisition-interval-time (valid for
Solaris and AIX only)]
[-d disk-busy-time-acquisition-interval-time (valid for Solaris and AIX
only)]
[-t trace-mask-value]
[-k]
[-T trace-mask-value]
```

Description

The `htc_monagt1` process periodically collects the CPU utilization rate, CPU utilization time, and disk busy time. This command is a daemon process.

The following table lists the options and the supported MIB groups.

Option	MIB
-i (CPU utilization rate)	hitachi.systemExMib.cometMibs.system.hiux.cpuUtil group
-s (CPU utilization time)	Solaris and AIX hp.nm.system.general.computerSystem group <ul style="list-style-type: none">computerSystemUserCPUcomputerSystemSysCPUcomputerSystemIdleCPUcomputerSystemNiceCPU hitachi.systemExMib.cometMibs.system.hiux.processor group <ul style="list-style-type: none">processorUserCPUtimeprocessorNiceCPUtimeprocessorSysCPUtimeprocessorIdleCPUtimeprocessorWaitCPUtime hitachi.systemExMib.cometMibs.system.hiux.computerSystem64 group <ul style="list-style-type: none">computerSystem64UserCPUcomputerSystem64SysCPUcomputerSystem64IdleCPU
-d (disk busy time)	hitachi.systemExMib.cometMibs.system.hiux.diskBusyInfo group hitachi.systemExMib.cometMibs.system.hiux.diskBusyAvail group

Location

- Solaris: `/opt/CM2/ESA/bin`
- AIX, Linux: `/usr/sbin`

Arguments

-i *CPU-utilization-acquisition-interval-time* ((0-1440))<<5>>

Specifies in minutes the interval for acquiring the CPU utilization rate. If you specify 0, CPU utilization rate information is not acquired.

-s *CPU-utilization-time-information-acquisition-interval-time* ((0-1440))<<5>>

Specifies in minutes the interval for acquiring CPU utilization time information. If you specify 0, CPU utilization time information is not acquired.

-d *disk-busy-time-acquisition-interval-time* ((0-1440))<<5>>

Specifies in minutes the interval for acquiring disk busy time information. If you specify 0, disk busy time information is not acquired.

-t *trace-mask-value*

Specifies to change the `htc_monagt1` trace mask value to the specified trace mask value.

If this argument is not specified, the trace mask value is 0.

The following table lists the trace mask values.

Trace mask	Acquisition content
0	Trace stop
1	Trace start

Log and trace are acquired at `/var/opt/CM2/ESA/log/htc_monagt1.log`. When this file reaches or exceeds 4 megabytes in size, the contents of `/var/opt/CM2/ESA/log/htc_monagt1.log` are copied to `/var/opt/CM2/ESA/log/htc_monagt1.log.old`, and `/var/opt/CM2/ESA/log/htc_monagt1.log` is overwritten.

-k

Specify this option if you want to send an end request for the currently running `htc_monagt1`.

-T *trace-mask-value*

Specify this option if you want to report changes to the trace mask for the currently running `htc_monagt1`.

External influences

Environment variables

Specify the following environment variables in the `SnmpHtcmonagt1` file.

`SNMP_HTCMONAGT1_START`

This environment variable specifies whether `htc_monagt1` starts when SNMP Agent starts.

Y: Starts.

N: Does not start.

`SNMP_HTC_AIX_CPU_SMT` (AIX)

This environment variable specifies the method used to acquire the CPU utilization rate information.

Y: Acquires the CPU utilization rate for the entire machine.

Value other than Y: Acquires the CPU utilization rate of each CPU.

The default is a value other than Y.

Notes

- If the `-i`, `-s`, and `-d` options are all set to 0, `htc_monagt1` does not start.
- If the `SNMP_HTC_AIX_CPU_SMT` environment variable is set to Y, you cannot monitor the CPU utilization rate of each CPU.

htc_unixagt1

Syntax

```
htc_unixagt1 [-aperror] [-apwarn] [-aptrace] [-apconfig]
             [-appacket] [-aptrap] [-apaccess]
             [-apemanate] [-apverbose] [-apuser] [-apall]
             [-help] [-retry N]
```

Description

The `htc_unixagt1` process provides the following Hitachi enterprise-specific MIB groups:

- **systemInfo group**
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.systemInfo)
- **virtualMemory group**
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.virtualMemory)
- **process group** (enterprises.hitachi.systemExMib.cometMibs.systems.hiux.process)
- **swapSpace group**
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.swapSpace)
- **diskBusy group**
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskBusy)
- **systemInfo64 group**
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.systemInfo64)
- **virtualMemory64Ex group**
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.virtualMemory64Ex)
- **process64 group**
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.process64)
- **fileSystem64 group**
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.fileSystem64)
- **system group** (enterprises.hitachi.systemExMib.cometMibs.systems.aix.system)
- **disk group** (enterprises.hitachi.systemExMib.cometMibs.systems.aix.disk)
- **page group** (enterprises.hitachi.systemExMib.cometMibs.systems.aix.page)
- **system group** (enterprises.hitachi.systemExMib.cometMibs.systems.solaris.system)
- **linuxSystem group**
(enterprises.hitachi.systemExMib.cometMibs.systems.linux.linuxsystem)
- **opConf group**
(enterprises.hitachi.systemExMib.cometMibs.subSystems.cometOP.opConf)

Location

- HP-UX (IPF), Solaris: `/opt/CM2/ESA/bin`
- AIX, Linux: `/usr/sbin`

Arguments

-aerror

Specify this option if you want to collect error logs.

-apwarn

Specify this option if you want to collect error and warning logs.

-aptrace

Specify this option if you want to collect trace logs.

-apconfig

Specify this option if you want to collect logs related to the configuration file.

-appacket

Specify this option if you want to collect logs related to packet assembly and analysis.

-aptrap

Specify this option if you want to collect logs related to trap messages.

-apaccess

Specify this option if you want to collect logs related to agent processing.

-apemanate

Specify this option if you want to collect logs related to the master agent and subagents.

-apverbose

Specify this option if you want to collect verbose logs.

-apuser

Specify this option if you want to collect user logs.

-apall

Specify this option if you want to collect all types of logs.

-help

Specify this option if you want to display the command syntax.

-retry *N*

Specify this option if you want the subagent to attempt to establish a connection with the master agent at *N*-second intervals.

External influences

Environment variables

Specify the following environment variables in the `SnmpHtcunix1` file.

`SR_SNMP_TEST_PORT`

This environment variable sets the master agent's SNMP reception port. If this environment variable is specified for a subagent, it is used as data for connecting to the master agent. This means that the value set for the subagent must be the same as the port number specified for the master agent. If this environment variable is not specified, the value in the `snmp` line of the `/etc/services` file is used. Normally, you do not need to specify this environment variable. You must specify this environment variable only if you want to change the master agent's SNMP reception port.

SNMP_HTC_FILE_EXTEND

In AIX or Linux, specify this environment variable to use SNMP Agent in an environment in which the total number of inodes in the file system exceeds $2^{32} - 1$. In a new installation, this environment variable is specified by default.

htc_unixagt2

Syntax

```
htc_unixagt2 [-aperror] [-apwarn] [-aptrace][-apconfig]
              [-appacket] [-aptrap] [-apaccess]
              [-apemanate] [-apverbose] [-apuser] [-apall]
              [-help] [-retry N]
```

Description

The `htc_unixagt2` process provides the following Hitachi enterprise-specific MIB groups:

- **processor group**
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.processor)
- **diskInfo group**
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskInfo)
- **swapInfo group**
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.swapInfo)
- **processor64 group**
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.processor64)
- **diskInfo64 group**
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskInfo64)
- **swapSystem64 group**
(enterprises.hitachi.systemExMib.cometMibs.systems.hiux.swapSystem64)

Location

- HP-UX (IPF): /opt/CM2/ESA/bin

Arguments

`-aperror`

Specify this option if you want to collect error logs.

`-apwarn`

Specify this option if you want to collect error and warning logs.

`-aptrace`

Specify this option if you want to collect trace logs.

`-apconfig`

Specify this option if you want to collect logs related to the configuration file.

`-appacket`

Specify this option if you want to collect logs related to packet assembly and analysis.

`-aptrap`

Specify this option if you want to collect logs related to trap messages.

`-apaccess`

Specify this option if you want to collect logs related to agent processing.

-apemanate

Specify this option if you want to collect logs related to the master agent and subagents.

-apverbose

Specify this option if you want to collect verbose logs.

-apuser

Specify this option if you want to collect user logs.

-apall

Specify this option if you want to collect all types of logs.

-help

Specify this option if you want to display the command syntax.

-retry *N*

Specify this option if you want the subagent to attempt to establish a connection with the master agent at *N*-second intervals.

External influences

Environment variables

Specify the following environment variable in the `SnmpHtcunix2` file.

`SR_SNMP_TEST_PORT`

This environment variable sets the master agent's SNMP reception port. If this environment variable is specified for a subagent, it is used as data for connecting to the master agent. This means that the value set for the subagent must be the same as the port number specified for the master agent. If this environment variable is not specified, the value in the `snmp` line of the `/etc/services` file is used. Normally, you do not need to specify this environment variable. You must specify this environment variable only if you want to change the master agent's SNMP reception port.

`SNMP_HTC_HPUX_ENABLE_PROCESSOR` (HP-UX (IPF))

This environment variable specifies how processor information is acquired.

`Y`: Acquire information only for processors that are enabled.

Value other than `Y`: Acquire information from the OS on all processors, regardless of whether they are enabled or disabled.

The default is a value other than `Y`.

htc_unixagt3

Syntax

```
htc_unixagt3 [-aperror] [-apwarn] [-aptrace] [-apconfig]
              [-appacket] [-aptrap] [-apaccess]
              [-apemanate] [-apverbose] [apuser] [-apall]
              [-help] [-retry N]
```

Description

The `htc_unixagt3` process provides the following Hitachi enterprise-specific MIB groups:

- `cpuUtil` group (`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.cpuUtil`)
- `virtualMemory64` group (`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.virtualMemory64`)
- `diskBusyInfo` group (`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskBusyInfo`)
- `diskBusyAvail` group (`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.diskBusyAvail`)
- `disk64Ex` group (`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.disk64Ex`)

Location

- Solaris: `/opt/CM2/ESA/bin`
- AIX, Linux: `/usr/sbin`

Arguments

`-aperror`

Specify this option if you want to collect error logs.

`-apwarn`

Specify this option if you want to collect error and warning logs.

`-aptrace`

Specify this option if you want to collect trace logs.

`-apconfig`

Specify this option if you want to collect logs related to the configuration file.

`-appacket`

Specify this option if you want to collect logs related to packet assembly and analysis.

`-aptrap`

Specify this option if you want to collect log related to trap messages.

`-apaccess`

Specify this option if you want to collect log related to agent processing.

`-apemanate`

Specify this option if you want to collect log related to the master agent and subagents.

-apverbose

Specify this option if you want to collect verbose logs.

-apuser

Specify this option if you want to collect user logs.

-apall

Specify this option if you want to collect all types of logs.

-help

Specify this option if you want to display the command syntax.

-retry *N*

Specify this option if you want the subagent to attempt to establish a connection with the master agent at *N*-second intervals.

External influences

Environment variables

Specify the following environment variable in the `SnmpHtcunix3` file.

`SR_SNMP_TEST_PORT`

This environment variable `SR_SNMP_TEST_PORT` specifies the SNMP reception port on the master agent. If this environment variable is specified on a subagent, it will be used as data required for connection to the master agent. Therefore, the value assigned to this environment variable must be equal to the port number specified on the master agent. If this environment variable is not specified, the pertinent value on the `snmp` line in the `/etc/services` file will be used. In general, you do not need to specify this environment variable. You only need to specify it if you want to change the SNMP reception port on the master agent.

htc_unixagt4

Syntax

```
htc_unixagt4 [-aperror] [-apwarn] [-aptrace] [-apconfig]
              [-appacket] [-aptrap] [-apaccess]
              [-apemanate] [-apverbose] [-apuser] [-apall]
              [-help] [-retry N]
```

Description

The `htc_unixagt4` process provides the following Hitachi enterprise-specific MIB group:

- `computerSystem64` group
(`enterprises.hitachi.systemExMib.cometMibs.systems.hiux.computerSystem64`)

Location

- HP-UX (IPF) and Solaris: `/opt/CM2/ESA/bin`
- AIX and Linux: `/usr/sbin`

Arguments

`-aperror`

Specify this option if you want to collect error logs.

`-apwarn`

Specify this option if you want to collect error and warning logs.

`-aptrace`

Specify this option if you want to collect trace logs.

`-apconfig`

Specify this option if you want to collect logs related to the configuration file.

`-appacket`

Specify this option if you want to collect logs related to packet assembly and analysis.

`-aptrap`

Specify this option if you want to collect logs related to trap messages.

`-apaccess`

Specify this option if you want to collect logs related to agent processing.

`-apemanate`

Specify this option if you want to collect logs related to the master agent and subagents.

`-apverbose`

Specify this option if you want to collect verbose logs.

`-apuser`

Specify this option if you want to collect user logs.

`-apall`

Specify this option if you want to collect all types of logs.

-help

Specify this option if you want to display the command syntax.

-retry N

Specify this option if you want the subagent to attempt to establish a connection with the master agent at *N*-second intervals.

External influences

Environment variables

SR_SNMP_TEST_PORT

The environment variable `SR_SNMP_TEST_PORT` specifies the SNMP reception port on the master agent. If this environment variable is specified on a subagent, it will be used as data required for connecting to the master agent. Therefore, the value assigned to this environment variable must be equal to the port number specified on the master agent. If this environment variable is not specified, the pertinent value on the `snmp` line in the `/etc/services` file is used. In general, you do not need to specify this environment variable. You only need to specify it if you want to change the SNMP reception port on the master agent.

naaagt

Syntax

```
naaagt [-aperror] [-apwarn] [-aptrace] [-apverbose]
        [-apall] [-help] [-port port-number]
        [-readcomm community-name] [-timeout time-out-value]
        [-writecomm community-name]
        [-hexdump] [-vbdump] [-n]
        [-v1 | -v2c]
```

Description

The `naaagt` process activates the native agent adapter function.

Location

- Solaris: `/opt/CM2/ESA/bin`
- AIX, Linux: `/usr/sbin`

Arguments

`-aperror`

Specify this option if you want to collect error logs.

`-apwarn`

Specify this option if you want to output error and warning logs.

`-aptrace`

Specify this option if you want to collect trace logs.

`-apverbose`

Specify this option if you want to output verbose logs.

`-apall`

Specify this option if you want to collect all log types.

`-help`

Specify this option if you want to look at the command syntax.

`-port port-number`

Specify the UDP *port-number* of the UDP port to be connected to the native agent. The port number you specify here is the destination port number. If omitted, 8161 is assumed.

`-readcomm community-name`

Specify the *community-name* that the `naaagt` process will use when it sends a GET request to the native agent. The default value is `public`.

Note the following when specifying the community name to be used in a GET request:

- Insert a single-byte space between the `readcomm` tag and *community-name*.
- The maximum length of *community-name* is 60 characters.
- Specify only a single `readcomm` tag in the `naa.cnf` definition file.
- When using the `readcomm` tag, do not specify the `-readcomm` option of the `naaagt` process.

`-timeout` *time-out-value*

Specify the *time-out-value* in seconds during which the `naaagt` process will wait for a GET response from the native agent. The default value is 4.

`-writecomm` *community-name*

Specify the *community-name* that the `naaagt` process will use when it sends a SET request to the native agent. The default value is `public`.

Note the following when specifying the community name to be used in a SET request.

- Insert a single-byte space between the `writecomm` tag and *community-name*.
- The maximum length of *community-name* is 60 characters.
- Specify only a single `writecomm` tag in the `naa.cnf` definition file.
- When using the `writecomm` tag, do not specify the `-writecomm` option of the `naaagt` process.

`-hexdump`

Displays, in hexadecimal dump to the standard output, the contents of SNMP packets sent or received by the `naaagt` process.

`-vbdump`

Displays, to the standard output, the contents of the `VarBind` list for SNMP packets sent or received by the `naaagt` process.

`-n`

The `naaagt` process is not used as a daemon.

`-v1`

Specify this option if you want the `naaagt` process to use the SNMPv1 protocol when it sends an SNMP request to the native agent. The SNMPv1 protocol is assumed when neither `-v1` nor `-v2c` is specified.

`-v2c`

Specify this option if you want the `naaagt` process to use the SNMPv2c protocol when it sends an SNMP request to the native agent.

External influences

Environment variables

Specify the following environment variable in the `SnmpNaa` file.

`SR_SNMP_TEST_PORT`

The environment variable `SR_SNMP_TEST_PORT` specifies the SNMP reception port on the master agent. If this environment variable is specified on a subagent, it will be used as data required for connection to the master agent. Therefore, the value assigned to this environment variable must be equal to the port number specified on the master agent. If this environment variable is not specified, the pertinent value on the `snmp` line in the `/etc/services` file will be used. In general, you do not need to specify this environment variable. You only need to specify it if you want to change the SNMP reception port on the master agent.

trapdestagt

Syntax

```
trapdestagt [-aperror] [-apwarn] [-aptrace] [-apconfig]
            [-appacket] [-aptrap] [-apaccess]
            [-apemanate] [-apverbose] [-apuser] [-apall]
            [-help] [-retry N]
```

Description

The `trapdestagt` process provides the `Trap` group from the Hewlett-Packard enterprise-specific MIB (`enterprises.hp.nm.snmp.trap`).

Location

- HP-UX (IPF), Solaris: `/opt/CM2/ESA/bin`
- AIX, Linux: `/usr/sbin`

Arguments

`-aperror`

Specify this option if you want to collect error logs.

`-apwarn`

Specify this option if you want to collect error and warning logs.

`-aptrace`

Specify this option if you want to collect trace logs.

`-apconfig`

Specify this option if you want to collect logs related to the configuration file.

`-appacket`

Specify this option if you want to collect logs related to packet assembly and analysis.

`-aptrap`

Specify this option if you want to collect logs related to trap messages.

`-apaccess`

Specify this option if you want to collect logs related to agent processing.

`-apemanate`

Specify this option if you want to collect logs related to the master agent and subagents.

`-apverbose`

Specify this option if you want to collect verbose logs.

`-apuser`

Specify this option if you want to collect user logs.

`-apall`

Specify this option if you want to collect all types of logs.

`-help`

Specify this option if you want to display the command syntax.

`-retry N`

Specify this option if you want the subagent to attempt to establish a connection with the master agent at *N*-second intervals.

External influences

Environment variables

Specify the following environment variable in the `SnmpTrpDst` file.

`SR_SNMP_TEST_PORT`

The environment variable `SR_SNMP_TEST_PORT` specifies the SNMP reception port on the master agent. If this environment variable is specified on a subagent, it will be used as data required for connection to the master agent. Therefore, the value assigned to this environment variable must be equal to the port number specified on the master agent. If this environment variable is not specified, the pertinent value on the `snmp` line in the `/etc/services` file will be used. In general, you do not need to specify this environment variable. You only need to specify it if you want to change the SNMP reception port on the master agent.

6

Definition Files

This chapter explains the definition files used by SNMP Agent.

About definition files

Definition files define SNMP Agent information. The following types of definition files are used:

Type of definition files

- Configuration file
- Environment variable definition file
- Operating locale definition file
- File system definition file
- Disk definition file

When a definition takes effect

Stop SNMP Agent before you edit a definition file. After the definition file has been edited, the new setting takes effect when SNMP Agent starts.

The following table lists the definition files.

Table 6–1: Definition files

Type	Settings file	Description	Applicable OS
Configuration file	<code>snmpd.conf</code>	This file defines the following settings: <ul style="list-style-type: none"> • System contact and system location • Community name • IPv4 trap destination 	HP-UX (IPF), Solaris, AIX, and Linux
	<code>snmpd.cnf</code>	This file defines the following settings: <ul style="list-style-type: none"> • IPv6 trap destination • Settings for the sending of authentication failure traps • Maximum number of connected subagents • Maximum number of threads concurrently generated by the master agent • <code>sysName</code> value 	HP-UX (IPF), Solaris, AIX, and Linux
	<code>naa.cnf</code>	This file defines the following settings: <ul style="list-style-type: none"> • MIB objects to be acquired from or set to the native agent • Community name of the SNMP request to be sent by the native agent adapter to the native agent 	Solaris, AIX, and Linux
Environment variable definition file	<code>SnpMaster</code>	This file defines the following environment variables: <ul style="list-style-type: none"> • <code>SNMP_MASTER_OPTIONS</code> • <code>SR_SNMP_TEST_PORT</code> • <code>SR_TRAP_TEST_PORT</code> • <code>SNMP_HTC_AUTH_LOG</code> • <code>SNMP_HTC_INIT_WAIT_TIME</code> • <code>SR_LOG_DIR</code> • <code>SNMP_HTC_SNMPD_LOG_SIZE</code> • <code>SNMP_HTC_SNMPD_LOG_CNT</code> • <code>PSALLOC</code> • <code>NODISCLAIM</code> 	HP-UX (IPF), Solaris, AIX, and Linux
	<code>SnpNaa</code>	This file defines the following environment variables: <ul style="list-style-type: none"> • <code>SNMP_NAA_OPTIONS</code> 	Solaris, AIX, and Linux

Type	Settings file	Description	Applicable OS
		<ul style="list-style-type: none"> SR_SNMP_TEST_PORT 	
	SnmpNative	<p>This file defines the following environment variables:</p> <ul style="list-style-type: none"> SNMP_NATIVE_OPTIONS SNMP_SNMPMIBD_OPTIONS SNMP_HOSTMIBD_OPTIONS SNMP_AIXMIBD_OPTIONS 	AIX
	SnmpHpunix	<p>This file defines the following environment variables:</p> <ul style="list-style-type: none"> SNMP_HPUNIX_OPTIONS SR_SNMP_TEST_PORT SNMP_HTC_SOLARIS_SWAP_RESERVED SNMP_HTC_AIX_EXCEPT_FILECACHE SNMP_HTC_LINUX_INACTIVE_MEM 	Solaris, AIX, and Linux
	SnmpTrpDst	<p>This file defines the following environment variables:</p> <ul style="list-style-type: none"> SNMP_TRAPDEST_OPTIONS SR_SNMP_TEST_PORT 	HP-UX (IPF), Solaris, AIX, and Linux
	SnmpHtcunix1	<p>This file defines the following environment variables:</p> <ul style="list-style-type: none"> SNMP_HTCUNIX1_OPTIONS SR_SNMP_TEST_PORT SNMP_HTC_FILE_EXTEND 	HP-UX (IPF), Solaris, AIX, and Linux
	SnmpHtcunix2	<p>This file defines the following environment variables:</p> <ul style="list-style-type: none"> SNMP_HTCUNIX2_OPTIONS SR_SNMP_TEST_PORT SNMP_HTC_HPUX_ENABLE_PROCESSOR 	HP-UX (IPF)
	SnmpHtcunix3	<p>This file defines the following environment variables:</p> <ul style="list-style-type: none"> SNMP_HTCUNIX3_OPTIONS SR_SNMP_TEST_PORT 	Solaris, AIX, and Linux
	SnmpHtcunix4	<p>This file defines the following environment variables:</p> <ul style="list-style-type: none"> SNMP_HTCUNIX4_OPTIONS SR_SNMP_TEST_PORT 	HP-UX (IPF), Solaris, AIX, and Linux
	SnmpHtcmonagt1	<p>This file defines the following environment variables:</p> <ul style="list-style-type: none"> SNMP_HTCMONAGT1_OPTIONS SNMP_HTCMONAGT1_START SNMP_HTC_AIX_CPU_SMT 	Solaris, AIX, and Linux
	SnmpExtAgt	<p>This file defines the following environment variables:</p> <ul style="list-style-type: none"> SNMP_EXTAGT_OPTIONS SR_SNMP_TEST_PORT 	HP-UX (IPF), Solaris, AIX, and Linux
Operating locale definition file	esalocale.conf	<p>This file defines the following setting:</p> <ul style="list-style-type: none"> Operating locale of this product 	HP-UX (IPF), Solaris, AIX, and Linux
File system definition file	esafilesys.conf	<p>This file defines the following settings:</p> <ul style="list-style-type: none"> Setting for checking whether the monitoring-target file system is mounted Setting for the file systems to be excluded from monitoring 	HP-UX (IPF), Solaris, AIX, and Linux
Adding file system target definition file	esafstab.conf	<p>This file defines the following setting:</p> <ul style="list-style-type: none"> Setting for the file systems to be collected as fileSystem64 group 	Linux

Type	Settings file	Description	Applicable OS
Disk definition file	esadisk.conf	This file defines the following setting: <ul style="list-style-type: none">• Setting for the disk devices to be excluded from monitoring	Linux

Definition file description format

Definition files are described in the format shown below. Not all of the items shown are explained for every definition file. Information unique to specific definition files is provided in some cases, in addition to the items listed below.

Format

Describes the input format for the definition file.

Overview

Provides an overview of the definition file.

Location

Lists the directory in which the definition file is stored.

Detailed description

Explains in detail what is set in the definition file.

Notes

Provides notes about editing the definition file.

Definition examples

Provides definition file definition examples.

Configuration file (snmpd.conf)

Format

```
contact: system-contact
location: system-location
get-community-name: get-community-name options
set-community-name: set-community-name options
trap-dest: IPv4-trap-destination
```

Overview

This file defines the following settings:

- System contact and system location
- Community names
- IPv4 trap destination

Location

HP-UX (IPF), Solaris, AIX, and Linux: `/etc/SnmpAgent.d/snmpd.conf`

Detailed description

`contact: system-contact`

Describe the system contact using an ASCII character string after the `contact:` label.

The maximum length is 255 characters.

`location: system-location`

Describe the system location using an ASCII character string after the `location:` label.

The maximum length is 255 characters.

`get-community-name: get-community-name options`

Describe the *get* community name of SNMP Agent using an ASCII character string.

Match this *get* community name to the *get* community name of the SNMP manager.

You can specify multiple *get* community names by adding lines.

- *get-community-name*

This is the password for GetRequest.

- *options*

You can specify `IP:` or `VIEW:` as an option.

If you omit both options, the community name permits access requests from any IP address. In addition, you can access any MIB supported by SNMP Agent.

`IP:`

The community name specified in the SNMP request restricts the IP addresses that can access MIBs. When you specify IP addresses that can access MIBs, separate each address with a space. No host name is allowed. Place at least one space between the community name and `IP:`, and at least one space between `IP:` and the IP address.

VIEW:

The specified community name restricts accessible MIBs. Specify object IDs representing accessible subtrees (1.3.6.1.2.1 for `mib-2`, for example), separated by a space. If you add a hyphen (-) before an object ID, the subtree represented by the object ID will be inaccessible. Place at least one space between the community name and `VIEW:`, and at least one space between `VIEW:` and the object ID. Also, place one space before a hyphen (-).

`set-community-name:` *set-community-name options*

Describe the *set* community name of SNMP Agent using an ASCII character string.

Match this *set* community name to the *set* community name of the SNMP manager.

You can specify multiple *set* community names by adding lines.

- *set-community-name*

This is the password for both `GetRequest` and `SetRequest`.

- *options*

See *options* for `get-community-name:`.

`trap-dest:`

- IPv4 trap destination

To send a trap to NNMI or a desired manager, specify a trap destination. Enter the host name or IP address of the manager to which you want SNMP Agent to send IPv4 traps.

Notes

- Specifying community names

To specify the same name for the *get* community name and *set* community name, specify the name only in the `set-community-name:` label. To specify different names for the *get* community name and *set* community name, specify those community names separately in the `get-community-name:` label and `set-community-name:` label.

Definition examples

- This example registers a system contact and a system location:

```
contact: Bob Jones (Phone 555-2000)
location: 1st Floor near Mens Room
```

- This example registers *get* community names:

```
get-community-name: public
get-community-name: private
```

- This example registers *set* community names:

```
set-community-name: private
set-community-name: point
```

- The example below registers a *get* community name with the `IP:` option specified.

If the community name specified in an SNMP request is `public`, SNMP Agent responds to the SNMP request as long as the request comes from `172.16.45.17` or `172.16.45.18`.

```
get-community-name: public IP: 172.16.45.17 172.16.45.18
```

- The example below registers a *get* community name with the `VIEW:` option specified.

If the community name specified in an SNMP request is `public`, SNMP Agent permits access to MIBs under `1.3.6.1.2.1` with the exception of `1.3.6.1.2.1.1`.

```
get-community-name: public VIEW: 1.3.6.1.2.1 -1.3.6.1.2.1.1
```

- This example specifies an IPv4 trap destination (for NNMi or a desired manager):

```
trap-dest: 15.2.113.223
```

Configuration file (snmpd.cnf)

Format

```
snmpTargetAddrEntry <CONFIG_NAME> transportDomainUdpIpv6 \  
  [<IPv6_ADDRESS>]:0 0 0 TrapConf <v1TrapParams | v2cTrapParams> readOnly \  
  [ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff]:0 2048  
  
snmpEnableAuthenTraps {1 | 2}  
  
MAX_SUBAGENTS maximum-number-of-connected-subagents  
  
MAX_THREADS maximum-number-of-threads-concurrently-generated-by-master-agent  
  
sysName local-host-name
```

Overview

This file defines the following settings:

- IPv6 trap destination
- Settings for the sending of authentication failure traps
- Maximum number of connected subagents
- Maximum number of threads concurrently generated by the master agent
- `sysName` value

Location

HP-UX (IPF), Solaris, AIX, and Linux: `/etc/srconf/agt/snmpd.cnf`

Detailed description

`snmpTargetAddrEntry`

This definition specifies an IPv6 trap destination.

- `CONFIG_NAME`: Specify any name as the configuration name.
If multiple IPv6 trap destinations are set, give each a unique configuration name. The configuration name cannot exceed 32 characters, and can contain only alphanumeric characters, including underscores.
- `transportDomainUdpIpv6`: To specify an IPv6 address without a scope ID, specify `transportDomainUdpIpv6`.
To specify an IPv6 address with a scope ID, enter `transportDomainUdpIpv6z`.
- `IPv6_ADDRESS`: Specify an IPv6 address for the IPv6 trap destination. A host name cannot be specified. For both SNMPv1 traps and SNMPv2c traps, you need to enclose the IPv6 address in square brackets ([]). Be careful not to omit these.
- `v1TrapParams | v2cTrapParams`: Specify the protocol version for the SNMP trap.

For SNMPv1 traps

Specify `v1TrapParams`.

For SNMPv2c traps

Specify `v2cTrapParams`.

snmpEnableAuthenTraps {1 | 2}

Indicates whether the SNMP Agent process is permitted to send authentication-failure traps. The default value is 1.

- 1: Sends authentication failure traps.
- 2: Sending of authentication failure traps is disabled.

MAX_SUBAGENTS *maximum-number-of-connected-subagents*

This value is the maximum number of connected subagents. The minimum value and the default value of MAX_SUBAGENTS is 22. The maximum value depends on the maximum number of threads that the OS can generate in a process.

MAX_THREADS *maximum-number-of-threads-concurrently-generated-by-master-agent*

This value is the maximum number of threads that can be concurrently generated by the master agent. The minimum value and the default value of MAX_THREADS is 22. The maximum value depends on the maximum number of threads that the OS can generate in a process.

sysName *local-host-name*

This option specifies the local host name that becomes the sysName value. By default, no value is set.

Definition examples

- Below is a sample configuration for sending SNMPv1 traps to interface number 1 at IP address fec0::1111:2222:3333:4444:5555. The configuration name is Trapsend_SNMPv1_IPv6. Note that when a %scope-ID is specified, you must write transportDomainUdpIpv6z.

```
snmpTargetAddrEntry Trapsend_SNMPv1_IPv6 transportDomainUdpIpv6z \  
[fec0::1111:2222:3333:4444:5555%1]:0 0 0 TrapConf v1TrapParams readOnly \  
[ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff]:0 2048
```

- Below is a sample configuration for sending SNMPv2c traps to IP address fec0::1111:2222:3333:4444:5555, with no %scope-ID specified. The configuration name is Trapsend_SNMPv2c_IPv6.

```
snmpTargetAddrEntry Trapsend_SNMPv2c_IPv6 transportDomainUdpIpv6 \  
[fec0::1111:2222:3333:4444:5555]:0 0 0 TrapConf v2cTrapParams readOnly \  
[ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff]:0 2048
```

- Below is a sample configuration for sending SNMPv2c traps to IP address fec0::1111:2222:3333:4444:5555 and IP address fec0::aaaa:bbbb:cccc:dddd:eeee, with no %scope-ID specified. The configuration names are NNM_1 and NNM_2. Note that when multiple trap destinations are set, each must be given a unique configuration name.

```
snmpTargetAddrEntry NNM_1 transportDomainUdpIpv6 \  
[fec0::1111:2222:3333:4444:5555]:0 0 0 TrapConf v2cTrapParams readOnly \  
[ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff]:0 2048  
snmpTargetAddrEntry NNM_2 transportDomainUdpIpv6 \  
[fec0::aaaa:bbbb:cccc:dddd:eeee]:0 0 0 TrapConf v2cTrapParams readOnly \  
[ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff]:0 2048
```

- To suppress the sending of authentication failure traps:

```
snmpEnableAuthenTraps 2
```

- Changing the maximum number of connected subagents

For details about how to change the maximum number of connected subagents, see [3.5 Changing the maximum number of connected subagents](#).

- Procedure for changing the `sysName` host name

For details about how to use a new host name as the `sysName` value after you have installed SNMP Agent and renamed the host, see [3.7.4 Notes about renaming a host](#).

Configuration file (naa.cnf)

Format

```
read object-ID
write object-ID
readcomm community-name
writecomm community-name
```

Overview

This file defines the following settings:

- MIB objects to be acquired from or set to the native agent
 - Adding read-only MIB objects (`read`)
 - Adding read-write MIB objects (`write`)
- Community name of the SNMP request to be sent by the native agent adapter to the native agent
 - Specifying the community name used in GET requests (`readcomm`)
 - Specifying the community name used in SET requests (`writecomm`)

`naa.cnf` is loaded when the native agent adapter starts. MIB groups to be acquired from the native agent are determined by `naa.cnf`. SNMP requests regarding the MIB objects defined in `naa.cnf` are issued from NNM or NNMi.

Location

Solaris, AIX, and Linux: `/etc/srconf/agt/naa.cnf`

Detailed description

`read object-ID`

Adds read-only MIB objects when MIB objects are registered on the native agent.

To define MIB subtrees or individual MIB objects as read-only, add `read` tag lines.

This object-ID is the object identifier of a MIB subtree or an individual MIB object. It must be specified in a numeric format. You cannot use object names here. It is not necessary to specify a suffix to the object ID; however, doing so will not result in an error. The object ID must not begin with a dot (.).

- *object-ID*: The table below shows the default object IDs of MIB objects defined in `naa.cnf`.

`write object-ID`

Adds read-write MIB objects when MIB objects are registered on the native agent.

To specify individual MIB objects as read-write objects, add `write` tag lines.

This object-ID is the object identifier of an individual MIB object. It must be specified in a numeric format. You cannot use object names here. It is not necessary to specify a suffix to the object ID; however, doing so will not result in an error. The object ID must not begin with a dot (.).

Although it is possible to register MIB subtrees as read-write, this is not recommended.

- *object-ID*: The following table shows the default object IDs of MIB objects defined in `naa.cnf`.

Table 6–2: Default object IDs of MIB objects defined in the `naa.cnf` definition file

MIB object ID	read/write	Applicable OS		
		Solaris	AIX	Linux
.1.3.6.1.2.1.2	read	Y	Y	Y
.1.3.6.1.2.1.3	read	Y	Y	Y
.1.3.6.1.2.1.4	read	Y	Y	Y
.1.3.6.1.2.1.5	read	Y	Y	Y
.1.3.6.1.2.1.6	read	Y	Y	Y
.1.3.6.1.2.1.7	read	Y	Y	Y
.1.3.6.1.2.1.10	read	N	Y	N
.1.3.6.1.2.1.12	read	N	Y	N
.1.3.6.1.2.1.25	read	N#	Y	Y
.1.3.6.1.2.1.31	read	N	N	Y
.1.3.6.1.2.1.55	read	N	N	Y
.1.3.6.1.4.1.2	read	N	Y	N
.1.3.6.1.4.1.4	read	N	Y	N
.1.3.6.1.4.1.42	read	Y	N	N
.1.3.6.1.4.1.2021	read	N#	N	Y

Legend:

Y: Defined.

N: Not defined.

#

If you need to acquire the MIB group `.1.3.6.1.2.1.25` or `.1.3.6.1.4.1.2021` from the native agent, add it to the default MIB objects defined in the `naa.cnf` configuration file.

`readcomm` *community-name*

This option specifies the community name used in a GET request sent to the native agent.

To specify a community name (default is `public`) used in a GET request that is sent from the native agent adapter to the native agent, add a line with the `readcomm` tag.

Match this community name to the GET community name of the native agent.

The following are notes about specifying community names used in GET requests:

- Place one space between the `readcomm` tag and the community name.
- The maximum length of a community name is 60 characters.
- In the `naa.cnf` definition file, specify only one `readcomm` tag.
- If you use the `readcomm` tag, do not specify the `-readcomm` option in the `naaagt` process.

`writcomm` *community-name*

This option specifies the community name used in a SET request sent to the native agent.

To specify a community name (default is `public`) in a SET request that is sent from the native agent adapter to the native agent, add a line with the `writcomm` tag.

Match this community name to the SET community name of the native agent.

The following are notes about specifying community names used in SET requests:

- Place one space between the `writcomm` tag and the community name.
- The maximum length of a community name is 60 characters.
- In the `naa.cnf` definition file, specify only one `writcomm` tag.
- If you use the `writcomm` tag, do not specify the `-writcomm` option in the `naaagt` process.

Notes

- Notes about deleting the `naa.cnf` definition file

Do not start the native agent adapter after deleting the `naa.cnf` definition file.

If you start the native agent adapter after deleting the `naa.cnf` definition file, only MIB-II information is acquired from the native agent. Therefore, specify the MIB objects to be acquired in the `naa.cnf` definition file.

- Notes about `naa.cnf` definition file specifications (in Solaris)

The path names and definition specifications are different for two different `naa.cnf` configuration files: the one referenced by the `naaagt` process of the Solaris edition of SNMP Agent, and the one referenced by the `naaagt` process provided by NNM. The `naa.cnf` configuration file provided by NNM cannot be used as is by SNMP Agent.

Definition examples

- This example specifies `.1.3.6.1.2.1.2` to define MIB subtrees or individual MIB objects as read-only:

```
read 1.3.6.1.2.1.2
```

- If the `naa.cnf` configuration file contains the following definitions, the native agent adapter will attempt to retrieve MIB-II interfaces, `at`, `ip`, `icmp`, `tcp`, `udp` and `host` groups from the native agent:

```
read 1.3.6.1.2.1.2
read 1.3.6.1.2.1.3
read 1.3.6.1.2.1.4
read 1.3.6.1.2.1.5
read 1.3.6.1.2.1.6
read 1.3.6.1.2.1.7
read 1.3.6.1.2.1.25
```

- This example specifies `.1.3.6.1.4.1.116` to define individual MIB objects as read-only:

```
write 1.3.6.1.4.1.116
```

- This example specifies `snmpread` as the community name in a GET request:

```
readcomm snmpread
```

- This example specifies `snmpwrite` as the community name in a SET request:

```
writcomm snmpwrite
```


Environment variable definition file (SnmpMaster)

Format

```
SNMP_MASTER_OPTIONS="[-aperror] [-apwarn] [-apverbose] [-authfail]
                    [-Contact system-contact] [-hexdump]
                    [-ip_proto [ipv4 | ipv4_ipv6 | ipv6]]
                    [-Location system-location] [-mask log-mask-value]
                    [-sysDescr description] [-tcplocal] [-vbdump]"

SR_SNMP_TEST_PORT=SNMP-reception-port-number

SR_TRAP_TEST_PORT=SNMP-trap-transmission-port-number

SNMP_HTC_AUTH_LOG

SNMP_HTC_INIT_WAIT_TIME=trap-transmission-wait-time (in seconds)

SR_LOG_DIR=output-directory

SNMP_HTC_SNMPD_LOG_SIZE=snmpd.logn-file-size

SNMP_HTC_SNMPD_LOG_CNT=snmpd.logn-file-count

PSALLOC=early

NODISCLAIM=true
```

Overview

The environment variable definition file (SnmpMaster) defines the following environment variables:

Environment variable name	Description
SNMP_MASTER_OPTIONS	This environment variable specifies startup options for the snmpdm process.
SR_SNMP_TEST_PORT	This environment variable changes SNMP Agent's SNMP reception port.
SR_TRAP_TEST_PORT	This environment variable sets an SNMP trap transmission port number.
SNMP_HTC_AUTH_LOG	This environment variable specifies whether unauthorized community names are collected.
SNMP_HTC_INIT_WAIT_TIME	This environment variable specifies the time period from when the snmpdm process completes startup processing until a coldStart trap is sent.
SR_LOG_DIR	This environment variable specifies the output directory for logs.
SNMP_HTC_SNMPD_LOG_SIZE	This environment variable specifies the size of the snmpd.logn file.
SNMP_HTC_SNMPD_LOG_CNT	This environment variable specifies the number of snmpd.logn files.
PSALLOC	This environment variable is specified to perform early page space allocation in AIX.
NODISCLAIM	This environment variable is specified to suppress memory area release in AIX.

Location

- HP-UX (IPF) and Linux: /opt/CM2/ESA/opt/SnmpMaster
- Solaris: /etc/rc.config.d/SnmpMaster

- AIX: `/usr/CM2/ESA/opt/SnmpMaster`

Detailed description

SNMP_MASTER_OPTIONS

This environment variable specifies startup options for the `snmpdm` process.

- `-aperror`: Specify this option if you want to collect error logs.
- `-apwarn`: Specify this option if you want to collect error and warning logs.
- `-apverbose`: Specify this option if you want to collect verbose logs.
- `-authfail` (abbreviation of `-authfail: -a`): This option suppresses the sending of authentication failure traps from the master agent.

This option is rarely needed, because it is provided for compatibility with earlier versions of SNMP Agent.

To inhibit authentication failure traps, specify 2 for `snmpEnableAuthenTraps` in `/etc/srconf/agt/snmpd.cnf`, and then reactivate the master agent.

For details about the `snmpd.cnf` file, see *Configuration file (snmpd.cnf)* in *Chapter 6. Definition Files*.

When you activate the master agent with the `-a` option specified, the master agent changes the setting of `snmpEnableAuthenTraps` in `/etc/srconf/agt/snmpd.cnf` to 2. If you later realize that you need no longer inhibit authentication failure traps, specify 1 for `snmpEnableAuthenTraps` in `/etc/srconf/agt/snmpd.cnf`, and then reactivate the master agent.

- `-Contact system-contact` (abbreviation of `-Contact: -C`): This option changes the system contact of the master agent.
- `-hexdump`: This option displays a hexadecimal dump of the contents of an SNMP packet. For details about how to use this option, see *7.4.1 Acquiring a master agent send/receive packet dump*.
- `-ip_proto [ipv4 | ipv4_ipv6 | ipv6]`: Specifies the IP version of the SNMP request reception port. If `-ip_proto` is not specified, an IPv4 and IPv6 (`ipv4_ipv6`) SNMP request reception port is used.
 - `ipv4`: Only an IPv4 SNMP request reception port is used.
 - `ipv4_ipv6`: Both IPv4 and IPv6 SNMP request reception ports are used.
 - `ipv6`: Only an IPv6 SNMP request reception port is used.

- `-Location system-location` (abbreviation of `-Location: -L`)

This option changes the system location of the master agent.

- `-mask logmast-value` (abbreviation `-mask: -m`)

This option changes the master agent's logmask value to the specified logmask value.

The logmask value can be a character string, decimal number, or hexadecimal number. The following table lists logmask values, followed by code samples:

Type of logmask value	Log suppression	Trace log output	Warning log output	Error log output
Character string	--	FACTORY_TRACE	FACTORY_WARN	FACTORY_ERROR
Decimal number	0	8388608	268435456	536870912
Hexadecimal number	0x0	0x00800000	0x10000000	0x20000000

Legend:

--: Not applicable

- `-sysDescr description` (abbreviation of `-sysDescr: -sys`): This option changes the description of the master agent.

- `-tcplocal`: This option enables acceptance of TCP connections from the subagents.
- `-vbdump`: This option displays the contents of the VarBind lists in the SNMP packets. For details about how to use this option, see [7.4.1 Acquiring a master agent send/receive packet dump](#).

SR_SNMP_TEST_PORT

This environment variable changes SNMP Agent's SNMP reception port.

- *SNMP-reception-port-number* after the change

This environment variable sets the master agent's SNMP reception port. If this environment variable is not specified, the value in the `snmp` line of the `/etc/services` file is used. Normally, you do not need to specify this environment variable. You must specify this environment variable only if you want to change the master agent's SNMP reception port.

By default, the following values are set:

- Solaris or AIX: 161
- Linux: 22161

For details about how to change the SNMP reception port on SNMP Agent, see [3.4.1 Changing the SNMP reception port on SNMP Agent](#).

SR_TRAP_TEST_PORT

This environment variable sets an SNMP trap transmission port number.

- *SNMP trap transmission port number*

This option specifies the master agent's SNMP trap notification port number. The default is 162.

SNMP_HTC_AUTH_LOG

To collect logs of unauthorized community names, specify 1. By default, this environment variable is not specified.

SNMP_HTC_INIT_WAIT_TIME

This environment variable specifies the time period from when the `snmpd` process completes startup processing until a `coldStart` trap is sent.

- *trap-transmission-wait-time* (in seconds)

Specify a value from 0 to 300 (in seconds) as the time period from when the `snmpd` process completes startup processing until a `coldStart` trap is sent. SNMP requests received during this period are discarded. The default is 15 seconds.

Specifying a value of less than 15 seconds as the time set in the `SNMP_HTC_INIT_WAIT_TIME` environment variable increases the probability that a `noSuchName` error will be returned in response to a request from the SNMP manager.

SR_LOG_DIR

This environment variable specifies the output directory for logs.

- *output-directory*

This option specifies the output directory for logs.

SNMP_HTC_SNMPD_LOG_SIZE

This environment variable specifies the size of the `snmpd.logn` file.

- *snmpd.logn-file-size*

This option specifies the size of the output files for logs, hexadecimal dumps, and VarBind lists as a value from 0 to 50 (in megabytes). If 0 is specified, logs, hexadecimal dumps, and VarBind lists are not output. The default is 10 megabytes.

SNMP_HTC_SNMPD_LOG_CNT

This environment variable specifies the number of `snmpd.logn` files.

- *snmpd.logn-file-count*

This option specifies the number of output files for logs, hexadecimal dumps, and VarBind lists as a value from 1 to 10. The default is 10 files.

```
PSALLOC=early
```

In the event of a shortage of OS memory in AIX, SIGKILL is issued and the process might terminate. You can avoid this by specifying PSALLOC=early.

```
NODISCLAIM=true
```

If you set early in the PSALLOC environment variable, also set the NODISCLAIM=true environment variable.

Notes

The following are notes about backing up the environment variable definition files:

- When you back up an environment variable definition file, verify that the name of the backup file does not begin with Snmp.

The following shows an example of a name for a backup file.

```
Example: Backup file of /opt/CM2/ESA/opt/SnmpMaster
/opt/CM2/ESA/opt/Bak.SnmpMaster
```

- If your OS is Solaris, do not create backup environment variable definition files under /etc/rc.config.d.

The following are notes about sending coldStart traps when the OS starts up:

- By default, the snmpdm process of the master agent sends a coldStart trap 15 seconds after it starts up. No response is sent to the manager's request during this time, because the process sends a coldStart trap without checking whether startup processing is complete for other subagents. Normally, 15 seconds are sufficient for subagents to complete their startup processing; however, some subagents might require more time depending on the environment. If this is the case, adjust the timing of coldStart transmission by specifying the appropriate time (in seconds) that elapses before the coldStart trap is sent in the SNMP_HTC_INIT_WAIT_TIME environment variable in the SnmpMaster file.

Definition examples

- This example specifies 162 for the SNMP trap transmission port number (SR_TRAP_TEST_PORT environment variable):

```
SR_TRAP_TEST_PORT=162
export SR_TRAP_TEST_PORT
```

- In the following example, the SNMP manager and SNMP Agent communicate using only IPv6:

```
SNMP_MASTER_OPTIONS="-ip_proto ipv6 -tcplocal -aperror -apwarn -apverbose
-hexdump -vbdump"
export SNMP_MASTER_OPTIONS
```

- This example specifies the time (in seconds) that elapses before the coldStart trap is sent:

```
SNMP_HTC_INIT_WAIT_TIME=15
export SNMP_HTC_INIT_WAIT_TIME
```

- This example prevents SIGKILL from being issued when a shortage of OS memory occurs in AIX:

```
PSALLOC=early
export PSALLOC
```

```
NODISCLAIM=true
export NODISCLAIM
```

- A specification example of the *SNMP_HTC_SNMPD_LOG_SIZE* environment variable is shown below. The unit is megabytes. The example specifies 10 megabytes.

```
SNMP_HTC_SNMPD_LOG_SIZE=10
export SNMP_HTC_SNMPD_LOG_SIZE
```

- A specification example of the *SNMP_HTC_SNMPD_LOG_CNT* environment variable is shown below. The unit is files. The example specifies 10 files.

```
SNMP_HTC_SNMPD_LOG_CNT=10
export SNMP_HTC_SNMPD_LOG_CNT
```

- This example specifies a log output destination:

```
SR_LOG_DIR=/tmp/esalog
export SR_LOG_DIR
```

- This example specifies the acquisition of the sending source IP address and community name for SNMP requests when the community name is invalid:

```
SNMP_HTC_AUTH_LOG=1
export SNMP_HTC_AUTH_LOG
```

Environment variable definition file (SnmpNaa)

Format

```
SNMP_NAA_OPTIONS="[-aperror] [-apwarn] [-port port-number]  
                 [-readcomm community-name] [-writecomm community-name]  
                 [-timeout time-out-value] [-apverbose] [-hexdump]  
                 [-vbdump] [-v1 | -v2c]"  
  
SR_SNMP_TEST_PORT=SNMP reception port-number
```

Overview

The environment variable definition file (SnmpNaa) defines the following environment variables:

Environment variable name	Description
SNMP_NAA_OPTIONS	This environment variable specifies startup options for the <code>naaagt</code> process.
SR_SNMP_TEST_PORT	This environment variable specifies the master agent's SNMP reception port.

Location

- Solaris: `/etc/rc.config.d/SnmpNaa`
- AIX: `/usr/CM2/ESA/opt/SnmpNaa`
- Linux: `/opt/CM2/ESA/opt/SnmpNaa`

Detailed description

SNMP_NAA_OPTIONS

This environment variable specifies the startup options for the `naaagt` process.

- `-aperror`: Specify this option if you want to collect error logs.
- `-apwarn`: Specify this option if you want to collect error and warning logs.
- `-port port-nmbr`: Changes the UDP port number for connecting to the native agent. The port number to be specified is a destination port number. The default port number is 8161.
- `-readcomm community-name`: Changes the *community-name* that the `naaagt` process uses when it sends a GET request to the native agent. The default value is `public`.
- `-writecomm community-name`: Changes the *community-name* that the `naaagt` process uses when it sends a SET request to the native agent. The default value is `public`.
- `-timeout time-out-value`: Specify *time-out-value* in seconds during which the `naaagt` process will wait for a GET response from the native agent. The default value is 4.
- `-apverbose`: Specify this option if you want to collect verbose logs.
- `-hexdump`: Displays, in hexadecimal dump to the standard output, the contents of SNMP packets sent or received by the `naaagt` process.
- `-vbdump`: Displays, to the standard output, the contents of the VarBind list for SNMP packets sent or received by the `naaagt` process.

- `-v1`: Specify this option if you want the `naaagt` process to use the SNMPv1 protocol when it sends an SNMP request to the native agent. The SNMPv1 protocol is assumed when neither `-v1` nor `-v2c` is specified.
- `-v2c`: Specify this option if you want the `naaagt` process to use the SNMPv2c protocol when it sends an SNMP request to the native agent.

SR_SNMP_TEST_PORT

This environment variable sets the master agent's SNMP reception port. Normally, you do not need to specify this environment variable since it is specified in the environment variable definition file `SnmptMaster`.

For details about how to change the SNMP reception port, see [3.4.1 Changing the SNMP reception port on SNMP Agent](#).

Definition examples

- This example specifies the `-v2c` option in the `SNMP_NAA_OPTIONS` environment variable.

Initial value

```
SNMP_NAA_OPTIONS="-aperror -apwarn -apverbose -hexdump -vbdump"
export SNMP_NAA_OPTIONS
```

`-v2c` option specified

```
SNMP_NAA_OPTIONS="-v2c -aperror -apwarn -apverbose -hexdump -vbdump"
export SNMP_NAA_OPTIONS
```

- This example specifies 161 for the SNMP request transmission port of SNMP Agent's native agent adapter:

```
SNMP_NAA_OPTIONS="-port 161 -aperror -apwarn -apverbose -hexdump -vbdump"
export SNMP_NAA_OPTIONS
```

Environment variable definition file (SnmpNative)

Format

```
SNMP_NATIVE_OPTIONS="-p port-number"  
  
SNMP_SNMPMIBD_OPTIONS="-c community-name"  
  
SNMP_HOSTMIBD_OPTIONS="-c community-name"  
  
SNMP_AIXMIBD_OPTIONS="-c community-name"
```

Overview

The environment variable definition file (`SnmpNative`) defines the following environment variables:

Environment variable name	Description
<code>SNMP_NATIVE_OPTIONS</code>	This environment variable specifies startup options for the native agent <code>snmpd</code> process in AIX.
<code>SNMP_SNMPMIBD_OPTIONS</code>	This environment variable enables the <code>snmpmibd</code> process provided by AIX to communicate with the <code>snmpd</code> process (8161/udp) provided by AIX.
<code>SNMP_HOSTMIBD_OPTIONS</code>	This environment variable enables the <code>hostmibd</code> process provided by AIX to communicate with the <code>snmpd</code> process (8161/udp) provided by AIX.
<code>SNMP_AIXMIBD_OPTIONS</code>	This environment variable enables the <code>aixmibd</code> process provided by AIX to communicate with the <code>snmpd</code> process (8161/udp) provided by AIX.

Location

AIX: `/usr/CM2/ESA/opt/SnmpNative`

Detailed description

`SNMP_NATIVE_OPTIONS`

This environment variable specifies startup options for the native agent `snmpd` process in AIX.

- `-p port-number`

This option specifies the port number of the SNMP reception port on the native agent `snmpd` in AIX. The default port number is 8161.

`SNMP_SNMPMIBD_OPTIONS`

This environment variable specifies startup options for the native agent `snmpmibd` subagent process in AIX.

- `-c community-name`

This option specifies the community name to be used for acquiring MIB values. The specified community name needs to match the native agent (`snmpd`). The default value is `public`.

`SNMP_HOSTMIBD_OPTIONS`

This environment variable specifies startup options for the native agent `hostmibd` subagent process in AIX.

- `-c community-name`

This option specifies the community name to be used for acquiring MIB values. The specified community name needs to match the native agent (`snmpd`). The default value is `public`.

SNMP_AIXMIBD_OPTIONS

This environment variable specifies startup options for the native agent `aixmibd` subagent process in AIX.

- `-c community-name`

This option specifies the community name to be used for acquiring MIB values. The specified community name needs to match the native agent (`snmpd`). The default value is `public`.

Definition examples

- This example changes `8161`, shown in the following line, to another port number:

```
SNMP_NATIVE_OPTIONS="-p 8161"  
export SNMP_NATIVE_OPTIONS
```

- This example changes `public`, shown in the following line, to a community name for which the native agent `snmpd` process permits a `get/get-next-request`:

```
SNMP_SNMPMIBD_OPTIONS="-c public"  
export SNMP_SNMPMIBD_OPTIONS  
SNMP_HOSTMIBD_OPTIONS="-c public"  
export SNMP_HOSTMIBD_OPTIONS  
SNMP_AIXMIBD_OPTIONS="-c public"  
export SNMP_AIXMIBD_OPTIONS
```

Environment variable definition file (SnmpHpunix)

Format

```
SNMP_HPUNIX_OPTIONS="[-aperror] [-apwarn] [-apconfig] [-appacket]
                    [-aptrap] [-apaccess] [-apemanate] [-apverbose]
                    [-apuser] [-retry N]"

SR_SNMP_TEST_PORT=SNMP-reception-port-number

SNMP_HTC_SOLARIS_SWAP_RESERVED={Y | value-other-than-Y} (in Solaris)

SNMP_HTC_AIX_EXCEPT_FILECACHE={Y | value-other-than-Y} (in AIX)

SNMP_HTC_LINUX_INACTIVE_MEM={Y | value-other-than-Y} (in Linux)
```

Overview

The environment variable definition file (SnmpHpunix) defines the following environment variables:

Environment variable name	Description
SNMP_HPUNIX_OPTIONS	This environment variable specifies startup options for the <code>hp_unixagt</code> process.
SR_SNMP_TEST_PORT	This environment variable sets the master agent's SNMP reception port.
SNMP_HTC_SOLARIS_SWAP_RESERVED	This environment variable specifies whether the size of the reserved space is included as part of the device swap space size in Solaris.
SNMP_HTC_AIX_EXCEPT_FILECACHE	This environment variable specifies whether the file cache is excluded from the amount of physical memory currently in use in AIX.
SNMP_HTC_LINUX_INACTIVE_MEM	This environment variable specifies how to calculate the MIB value for the amount of free memory in Linux.

Location

- Solaris: `/etc/rc.config.d/SnmpHpunix`
- AIX: `/usr/CM2/ESA/opt/SnmpHpunix`
- Linux: `/opt/CM2/ESA/opt/SnmpHpunix`

Detailed description

SNMP_HPUNIX_OPTIONS

This environment variable specifies startup options for the `hp_unixagt` process.

- `-aperror`: Specify this option if you want to collect error logs.
- `--apwarn`: Specify this option if you want to collect error and warning logs.
- `-apconfig`: Specify this option if you want to collect logs related to the configuration file.
- `-appacket`: Specify this option if you want to collect logs related to packet assembly and analysis.
- `-aptrap`: Specify this option if you want to collect logs related to trap messages.
- `-apaccess`: Specify this option if you want to collect logs related to agent processing.
- `-apemanate`: Specify this option if you want to collect logs related to the master agent and subagents.

- `-apverbose`: Specify this option if you want to collect verbose logs.
- `-apuser`: Specify this option if you want to collect user logs.
- `-retry N`: Specify this option if you want the subagent to attempt to establish a connection with the master agent at *N*-second intervals.

SR_SNMP_TEST_PORT

This environment variable sets the master agent's SNMP reception port. Normally, you do not need to specify this environment variable since it is specified in the environment variable definition file `SnmpMaster`.

For details about how to change the SNMP reception port, see [3.4.1 Changing the SNMP reception port on SNMP Agent](#).

SNMP_HTC_SOLARIS_SWAP_RESERVED={Y | *value-other-than-Y*} (in Solaris)

This environment variable specifies whether the size of the reserved space is included as part of the device swap space size.

- *Y*: The size of the reserved space is included.
- Value other than *Y*: The size of the reserved space is not included. The default is a value other than *Y*.

Details follow about how to acquire the swap space size.

By default, the Solaris device swap space size acquired by SNMP Agent does not include the reserved value. To acquire the device swap space size including the reserved value, specify *Y* in the

`SNMP_HTC_SOLARIS_SWAP_RESERVED` environment variable in the `SnmpHpunix` file.

The reserved value means the amount of swap space that is not currently allocated but that will be obtained in memory for later use.

SNMP_HTC_AIX_EXCEPT_FILECACHE={Y | *value-other-than-Y*} (in AIX)

This environment variable specifies whether the file cache is excluded from the amount of physical memory currently in use.

- *Y*: The file cache is excluded from the amount of the physical memory currently in use.
- Value other than *Y*: The file cache is not excluded from the amount of physical memory currently in use. The default is a value other than *Y*.

Details follow about how to specify the amount of free space in physical memory in AIX.

In AIX, file access speed is improved by using the physical memory as a file cache. The file cache is therefore included in the amount of physical memory in use, and the amount of free space in physical memory obtained by SNMP Agent is the actual amount of free memory that is available.

If you want to determine the amount of free memory capacity while excluding the file cache from the amount of physical memory in use, set the environment variable `SNMP_HTC_AIX_EXCEPT_FILECACHE` to *Y* in the environment variable definition file `SnmpHpunix`. Then, the amount of free space in physical memory obtained by SNMP Agent is the sum of the free memory capacity and the file cache value.

SNMP_HTC_LINUX_INACTIVE_MEM={Y | *value-other-than-Y*} (in Linux)

This environment variable specifies how to calculate the MIB value for the amount of free memory.

- *Y*: The sum of the amount of free memory, inactive buffer memory, and inactive cache memory
- Value other than *Y*: The sum of the amount of free memory, buffer memory, and cache memory. The default is a value other than *Y*.

Details follow about the amount of free space in physical memory in Linux.

Linux actively allocates memory to buffer memory and cache memory. If a memory allocation request is issued from an application, buffer memory and cache memory are freed as necessary, and memory is allocated to the application. Therefore, the amount of free memory in the physical memory acquired by SNMP Agent is the sum of the amount of free memory, buffer memory, and cache memory.

You can acquire the sum of the amount of free memory, inactive buffer memory, and inactive cache memory as the amount of free memory in the physical memory by SNMP Agent if you specify Y for the `SNMP_HTC_LINUX_INACTIVE_MEM` environment variable in the `SnmpHpunix` environment variable definition file.

Definition examples

- This example edits `SNMP_HPUNIX_OPTIONS` of the `SnmpHpunix` file used in the `hp_unixagt` process:

```
SNMP_HPUNIX_OPTIONS="-aperror -apwarn -apverbose"  
export SNMP_HPUNIX_OPTIONS
```

- This example specifies how to acquire the swap space size in Solaris:

```
SNMP_HTC_SOLARIS_SWAP_RESERVED=Y  
export SNMP_HTC_SOLARIS_SWAP_RESERVED
```

- This example specifies the amount of free space in physical memory in AIX:

```
SNMP_HTC_AIX_EXCEPT_FILECACHE=Y  
export SNMP_HTC_AIX_EXCEPT_FILECACHE
```

- This example specifies the amount of free space in physical memory in Linux:

```
SNMP_HTC_LINUX_INACTIVE_MEM=Y  
exort SNMP_HTC_LINUX_INACTIVE_MEM
```

Environment variable definition file (SnmpTrpDst)

Format

```
SNMP_TRAPDEST_OPTIONS="[-aperror] [-apwarn] [-apconfig] [-appacket]
                        [-aptrap] [-apaccess] [-apemanate] [-apverbose]
                        [-apuser] [-retry N]"

SR_SNMP_TEST_PORT=SNMP-reception-port-number
```

Overview

The environment variable definition file (SnmpTrpDst) defines the following environment variables:

Environment variable name	Description
SNMP_TRAPDEST_OPTIONS	This environment variable specifies startup options for the trapdestagt process.
SR_SNMP_TEST_PORT	This environment variable sets the master agent's SNMP reception port.

Location

- HP-UX (IPF) and Linux: /opt/CM2/ESA/opt/SnmpTrpDst
- Solaris: /etc/rc.config.d/SnmpTrpDst
- AIX: /usr/CM2/ESA/opt/SnmpTrpDst

Detailed description

SNMP_TRAPDEST_OPTIONS

This environment variable specifies startup options for the trapdestagt process.

- `-aperror`: Specify this option if you want to collect error logs.
- `--apwarn`: Specify this option if you want to collect error and warning logs.
- `-apconfig`: Specify this option if you want to collect logs related to the configuration file.
- `-appacket`: Specify this option if you want to collect logs related to packet assembly and analysis.
- `-aptrap`: Specify this option if you want to collect logs related to trap messages.
- `-apaccess`: Specify this option if you want to collect logs related to agent processing.
- `-apemanate`: Specify this option if you want to collect logs related to the master agent and subagents.
- `-apverbose`: Specify this option if you want to collect verbose logs.
- `-apuser`: Specify this option if you want to collect user logs.
- `-retry N`: Specify this option if you want the subagent to attempt to establish a connection with the master agent at *N*-second intervals.

SR_SNMP_TEST_PORT

This environment variable sets the master agent's SNMP reception port. Normally, you do not need to specify this environment variable since it is specified in the environment variable definition file `SnmpMaster`.

For details about how to change the SNMP reception port, see [3.4.1 Changing the SNMP reception port on SNMP Agent](#).

Definition examples

- This example edits `SNMP_TRAPDEST_OPTIONS` of the `SnmTrpDst` file used in the `trapdestagt` process:

```
SNMP_TRAPDEST_OPTIONS="-aperror -apwarn -apverbose"  
export SNMP_TRAPDEST_OPTIONS
```

Environment variable definition file (SnmphTcunix1)

Format

```
SNMP_HTCUNIX1_OPTIONS="[-aperror] [-apwarn] [-apconfig] [-appacket]
                        [-aptrap] [-apaccess] [-apemanate] [-apverbose]
                        [-apuser] [-retry N]"

SR_SNMP_TEST_PORT=SNMP-reception-port-number

SNMP_HTC_FILE_EXTEND={1 | value-other-than-1} (in Linux or AIX)
```

Overview

The environment variable definition file (SnmphTcunix1) defines the following environment variables:

Environment variable name	Description
SNMP_HTCUNIX1_OPTIONS	This environment variable specifies startup options for the <code>htc_unixagt1</code> process.
SR_SNMP_TEST_PORT	This environment variable sets the master agent's SNMP reception port.
SNMP_HTC_FILE_EXTEND	In AIX or Linux, specify this environment variable when the total number of blocks and inodes in the file system exceeds the standard.

Location

- HP-UX (IPF) and Linux: `/opt/CM2/ESA/opt/SnmphTcunix1`
- Solaris: `/etc/rc.config.d/SnmphTcunix1`
- AIX: `/usr/CM2/ESA/opt/SnmphTcunix1`

Detailed description

SNMP_HTCUNIX1_OPTIONS

This environment variable specifies startup options for the `htc_unixagt1` process.

- `-aperror`: Specify this option if you want to collect error logs.
- `--apwarn`: Specify this option if you want to collect error and warning logs.
- `-apconfig`: Specify this option if you want to collect logs related to the configuration file.
- `-appacket`: Specify this option if you want to collect logs related to packet assembly and analysis.
- `-aptrap`: Specify this option if you want to collect logs related to trap messages.
- `-apaccess`: Specify this option if you want to collect logs related to agent processing.
- `-apemanate`: Specify this option if you want to collect logs related to the master agent and subagents.
- `-apverbose`: Specify this option if you want to collect verbose logs.
- `-apuser`: Specify this option if you want to collect user logs.
- `-retry N`: Specify this option if you want the subagent to attempt to establish a connection with the master agent at *N*-second intervals.

SR_SNMP_TEST_PORT

This environment variable sets the master agent's SNMP reception port. Normally, you do not need to specify this environment variable since it is specified in the environment variable definition file `SnpMaster`.

For details about how to change the SNMP reception port, see [3.4.1 Changing the SNMP reception port on SNMP Agent](#).

SNMP_HTC_FILE_EXTEND={1 | *value-other-than-1*} (in Linux or AIX)

In AIX or Linux, specify this environment variable when the total number of blocks and inodes in the file system exceeds the standard.

In AIX or Linux, if you want to use SNMP Agent in an environment in which there are more than $2^{32}-1$ blocks and inodes (objects in the `fileSystem64` group with ID 2.1.5 to 2.1.10) in the file system, you must specify 1.

Note that the `SNMP_HTC_FILE_EXTEND` environment variable does not need to be set in a new installation, because it is specified by default.

Definition examples

- This example edits `SNMP_HTCUNIX1_OPTIONS` of the `SnpHtcunix1` file used in the `htc_unixagt1` process:

```
SNMP_HTCUNIX1_OPTIONS="-aperror -apwarn -apverbose"  
export SNMP_HTCUNIX1_OPTIONS
```

- This example uses, in AIX or Linux, SNMP Agent in an environment in which there are more than $2^{32}-1$ blocks and inodes in the file system:

```
SNMP_HTC_FILE_EXTEND=1  
export SNMP_HTC_FILE_EXTEND
```


Environment variable definition file (SnmptHtcunix2)

Format

```
SNMP_HTCUNIX2_OPTIONS="[-aperror] [-apwarn] [-apconfig] [-appacket]
                        [-aptrap] [-apaccess] [-apemanate] [-apverbose]
                        [-apuser] [-retry N]"

SR_SNMP_TEST_PORT=SNMP-reception-port-number

SNMP_HTC_HPUX_ENABLE_PROCESSOR={Y | value-other-than-Y} (in HP-UX (IPF))
```

Overview

The environment variable definition file (SnmptHtcunix2) defines the following environment variables:

Environment variable name	Description
SNMP_HTCUNIX2_OPTIONS	This environment variable specifies startup options for the htc_unixagt2 process.
SR_SNMP_TEST_PORT	This environment variable sets the master agent's SNMP reception port.
SNMP_HTC_HPUX_ENABLE_PROCESSOR	This environment variable specifies how processor information is acquired from the OS.

Location

HP-UX (IPF): /opt/CM2/ESA/opt/SnmptHtcunix2

Detailed description

SNMP_HTCUNIX2_OPTIONS

This environment variable specifies startup options for the htc_unixagt2 process.

- `-aperror`: Specify this option if you want to collect error logs.
- `--apwarn`: Specify this option if you want to collect error and warning logs.
- `-apconfig`: Specify this option if you want to collect logs related to the configuration file.
- `-appacket`: Specify this option if you want to collect logs related to packet assembly and analysis.
- `-aptrap`: Specify this option if you want to collect logs related to trap messages.
- `-apaccess`: Specify this option if you want to collect logs related to agent processing.
- `-apemanate`: Specify this option if you want to collect logs related to the master agent and subagents.
- `-apverbose`: Specify this option if you want to collect verbose logs.
- `-apuser`: Specify this option if you want to collect user logs.
- `-retry N`: Specify this option if you want the subagent to attempt to establish a connection with the master agent at *N*-second intervals.

SR_SNMP_TEST_PORT

This environment variable sets the master agent's SNMP reception port. Normally, you do not need to specify this environment variable since it is specified in the environment variable definition file `SnmptMaster`.

For details about how to change the SNMP reception port, see [3.4.1 Changing the SNMP reception port on SNMP Agent](#).

SNMP_HTC_HPUX_ENABLE_PROCESSOR={ Y | *value-other-than-Y*} (in HP-UX (IPF))

This environment variable specifies how processor information is acquired from the OS.

By default, SNMP Agent acquires information on all processors from the OS, regardless of whether a processor is enabled or disabled.

This environment variable specifies how processor information is acquired.

- Y: Acquire information only for processors that are enabled.
- Value other than Y: Acquire information from the OS on all processors, regardless of whether they are enabled or disabled.

The default is a value other than Y.

Definition examples

- This example acquires information only for processors that are enabled:

```
SNMP_HTC_HPUX_ENABLE_PROCESSOR=Y
export SNMP_HTC_HPUX_ENABLE_PROCESSOR
```

Environment variable definition file (SnmPhtcunix3)

Format

```
SNMP_HTCUNIX3_OPTIONS="[-aperror] [-apwarn] [-apconfig] [-appacket]
                        [-aptrap] [-apaccess] [-apemanate] [-apverbose]
                        [-apuser] [-retry N]"

SR_SNMP_TEST_PORT=SNMP-reception-port-number
```

Overview

The environment variable definition file (SnmPhtcunix3) defines the following environment variables:

Environment variable name	Description
SNMP_HTCUNIX3_OPTIONS	This environment variable specifies startup options for the <code>htc_unixagt3</code> process.
SR_SNMP_TEST_PORT	This environment variable sets the master agent's SNMP reception port.

Location

- Solaris: `/etc/rc.config.d/SnmPhtcunix3`
- AIX: `/usr/CM2/ESA/opt/SnmPhtcunix3`
- Linux: `/opt/CM2/ESA/opt/SnmPhtcunix3`

Detailed description

SNMP_HTCUNIX3_OPTIONS

This environment variable specifies startup options for the `htc_unixagt3` process.

- `-aperror`: Specify this option if you want to collect error logs.
- `--apwarn`: Specify this option if you want to collect error and warning logs.
- `-apconfig`: Specify this option if you want to collect logs related to the configuration file.
- `-appacket`: Specify this option if you want to collect logs related to packet assembly and analysis.
- `-aptrap`: Specify this option if you want to collect logs related to trap messages.
- `-apaccess`: Specify this option if you want to collect logs related to agent processing.
- `-apemanate`: Specify this option if you want to collect logs related to the master agent and subagents.
- `-apverbose`: Specify this option if you want to collect verbose logs.
- `-apuser`: Specify this option if you want to collect user logs.
- `-retry N`: Specify this option if you want the subagent to attempt to establish a connection with the master agent at *N*-second intervals.

SR_SNMP_TEST_PORT

This environment variable sets the master agent's SNMP reception port. Normally, you do not need to specify this environment variable since it is specified in the environment variable definition file `SnmPmaster`.

For details about how to change the SNMP reception port, see [3.4.1 Changing the SNMP reception port on SNMP Agent](#).

Definition examples

- This example edits `SNMP_HTCUNIX3_OPTIONS` of the `SnpHtcunix3` file used in the `htc_unixagt3` process:

```
SNMP_HTCUNIX3_OPTIONS="-aperror -apwarn -apverbose"  
export SNMP_HTCUNIX3_OPTIONS
```

Environment variable definition file (SnmpHtcunix4)

Format

```
SNMP_HTCUNIX4_OPTIONS="[-aperror] [-apwarn] [-apconfig] [-appacket]
                        [-aptrap] [-apaccess] [-apemanate] [-apverbose]
                        [-apuser] [-retry N]"

SR_SNMP_TEST_PORT=SNMP-reception-port-number
```

Overview

The environment variable definition file (SnmpHtcunix4) defines the following environment variables:

Environment variable name	Description
SNMP_HTCUNIX4_OPTIONS	This environment variable specifies startup options for the <code>htc_unixagt4</code> process.
SR_SNMP_TEST_PORT	This environment variable sets the master agent's SNMP reception port.

Location

- HP-UX (IPF) and Linux: `/opt/CM2/ESA/opt/SnmpHtcunix4`
- Solaris: `/etc/rc.config.d/SnmpHtcunix4`
- AIX: `/usr/CM2/ESA/opt/SnmpHtcunix4`

Detailed description

SNMP_HTCUNIX4_OPTIONS

This environment variable specifies startup options for the `htc_unixagt4` process.

- `-aperror`: Specify this option if you want to collect error logs.
- `--apwarn`: Specify this option if you want to collect error and warning logs.
- `-apconfig`: Specify this option if you want to collect logs related to the configuration file.
- `-appacket`: Specify this option if you want to collect logs related to packet assembly and analysis.
- `-aptrap`: Specify this option if you want to collect logs related to trap messages.
- `-apaccess`: Specify this option if you want to collect logs related to agent processing.
- `-apemanate`: Specify this option if you want to collect logs related to the master agent and subagents.
- `-apverbose`: Specify this option if you want to collect verbose logs.
- `-apuser`: Specify this option if you want to collect user logs.
- `-retry N`: Specify this option if you want the subagent to attempt to establish a connection with the master agent at *N*-second intervals.

SR_SNMP_TEST_PORT

This environment variable sets the master agent's SNMP reception port. Normally, you do not need to specify this environment variable since it is specified in the environment variable definition file `SnmpMaster`.

For details about how to change the SNMP reception port, see [3.4.1 Changing the SNMP reception port on SNMP Agent](#).

Definition examples

- This example edits `SNMP_HTCUNIX4_OPTIONS` of the `SmpHtcunix4` file of the `htc_unixagt4` process:

```
SNMP_HTCUNIX4_OPTIONS="-aperror -apwarn -apverbose"  
export SNMP_HTCUNIX4_OPTIONS
```

Environment variable definition file (SnmPhtcmonagt1)

Format

```
SNMP_HTCMONAGT1_OPTIONS="[-i CPU-utilization-acquisition-interval-time]
                        [-s CPU-utilization-time-information-acquisition-
interval-time] (valid for Solaris and AIX only)
                        [-d disk-busy-time-acquisition-interval-time]
(valid for Solaris and AIX only)
                        [-t trace-mask-value]"

SNMP_HTCMONAGT1_START={Y | N}

SNMP_HTC_AIX_CPU_SMT={Y | value-other-than-Y} (in AIX)
```

Overview

The environment variable definition file (SnmPhtcmonagt1) defines the following environment variables:\

Environment variable name	Description
SNMP_HTCMONAGT1_OPTIONS	This environment variable specifies startup options for the htc_monagt1 process.
SNMP_HTCMONAGT1_START	This environment variable specifies whether htc_monagt1 starts when SNMP Agent starts.
SNMP_HTC_AIX_CPU_SMT	In AIX, this environment variable specifies the method used to acquire the CPU utilization rate information.

Location

- Solaris: /etc/rc.config.d/SnmPhtcmonagt1
- AIX: /usr/CM2/ESA/opt/SnmPhtcmonagt1
- Linux: /opt/CM2/ESA/opt/SnmPhtcmonagt1

Detailed description

SNMP_HTCMONAGT1_OPTIONS

This environment variable specifies startup options for the htc_monagt1 process.

- *-i CPU-utilization-acquisition-interval-time*

This option specifies in minutes the interval for acquiring the CPU utilization rate. If you specify 0, CPU utilization rate information is not acquired.

In Solaris or AIX, during the period from the start of SNMP Agent until the first acquisition, all the MIB values of the CPU time information are returned as a noSuchName error.

The CPU information is updated at regular intervals. The interval time is set in SNMP Agent as the acquisition interval for the CPU time information (default: 5 minutes). Thus, to collect the CPU time in Solaris or AIX, use an interval time that is longer than the acquisition interval for the CPU time information.

- *-s CPU-utilization-time-information-acquisition-interval-time* (valid for Solaris and AIX only)

This option specifies in minutes the interval for acquiring CPU utilization time information. If you specify 0, CPU utilization time information is not acquired.

- *-d disk-busy-time-acquisition-interval-time* (valid for Solaris and AIX only)

This option specifies in minutes the interval for acquiring disk busy time information. If you specify 0, disk busy time information is not acquired.

- `-t trace-mask-value`

This option specifies to change the `htc_monagt1` trace mask value to the specified trace mask value. If this argument is not specified, the trace mask value is 0.

trace-mask-value

0: Trace stop

1: Trace start

Logs and traces are acquired at `/var/opt/CM2/ESA/log/htc_monagt1.log`. When this file reaches or exceeds 4 megabytes, the contents of `/var/opt/CM2/ESA/log/htc_monagt1.log` are copied to `/var/opt/CM2/ESA/log/htc_monagt1.log.old`, and `/var/opt/CM2/ESA/log/htc_monagt1.log` is overwritten.

`SNMP_HTCMONAGT1_START={Y | N}`

This environment variable specifies whether `htc_monagt1` starts when SNMP Agent starts.

The default is `Y`.

- `Y`: Starts.
- `N`: Does not start.

`SNMP_HTC_AIX_CPU_SMT={Y | value-other-than-Y}` (in AIX)

This environment variable specifies the method used to acquire the CPU utilization rate information.

- `Y`: Acquires the CPU utilization rate for the entire machine.
If the `SNMP_HTC_AIX_CPU_SMT` environment variable is set to `Y`, you cannot monitor the CPU utilization rate of each CPU.
- Value other than `Y`: Acquires the CPU utilization rate of each CPU. The default is a value other than `Y`.

Definition examples

- This example starts the `htc_monagt1` when SNMP Agent starts:

```
SNMP_HTCMONAGT1_START=Y
export SNMP_HTCMONAGT1_START
```

- This example acquires the CPU utilization rate for the entire machine:

```
SNMP_HTC_AIX_CPU_SMT=Y
export SNMP_HTC_AIX_CPU_SMT
```


Environment variable definition file (SnmpExtAgt)

Format

```
SNMP_EXTAGT_OPTIONS="[-e extended-MIB-definition-file] [-E priority]  
                    [-aperror] [-apwarn] [-apconfig] [-appacket]  
                    [-aptrap] [-apaccess] [-apemanate] [-apverbose]  
                    [-apuser] [-retry N] [-fcmdguard N] [-pipeguard N]  
                    [-invokeid]"  
  
SR_SNMP_TEST_PORT=SNMP-reception-port-number
```

Overview

Environment variable definition file (SnmpExtAgt) defines the following environment variables:

Environment variable name	Description
SNMP_EXTAGT_OPTIONS	This environment variable specifies startup options for the <code>extsubagt</code> process.
SR_SNMP_TEST_PORT	This environment variable sets the master agent's SNMP reception port.

Location

- HP-UX (IPF) and Linux: `/opt/CM2/ESA/opt/SnmpExtAgt`
- Solaris: `/etc/rc.config.d/SnmpExtAgt`
- AIX: `/usr/CM2/ESA/opt/SnmpExtAgt`

Detailed description

SNMP_EXTAGT_OPTIONS

This environment variable specifies startup options for the `extsubagt` process.

- `-e extended-MIB-definition-file`: This option specifies an extended MIB definition file. The default definition file is `/etc/SnmpAgent.d/snmpd.extend`.
- `-E priority`: This option specifies the priority of the subagent.
- `-aperror`: Specify this option if you want to collect error logs.
- `--apwarn`: Specify this option if you want to collect error and warning logs.
- `-apconfig`: Specify this option if you want to collect logs related to the configuration file.
- `-appacket`: Specify this option if you want to collect logs related to packet assembly and analysis.
- `-aptrap`: Specify this option if you want to collect logs related to trap messages.
- `-apaccess`: Specify this option if you want to collect logs related to agent processing.
- `-apemanate`: Specify this option if you want to collect logs related to the master agent and subagents.
- `-apverbose`: Specify this option if you want to collect verbose logs.
- `-apuser`: Specify this option if you want to collect user logs.
- `-retry N`: Specify this option if you want the subagent to attempt to establish a connection with the master agent at *N*-second intervals.

- `-fcmdguard N`: This option specifies the *file_command* execution response monitoring time in seconds. The specified value *N* must be $1 \leq N \leq 90$.
- `-pipeguard N`: This option specifies the monitoring period in seconds from the time SNMP Agent writes data into *pipe_out_name* to the time processing results are written. The specified value *N* must be $1 \leq N \leq 90$.
- `-invokeid`: Specify this option to add an identification number as the first argument written to *pipe_out_name*. The identification number is in the form *xxxxxxx.yyyyyy*, where *xxxxxxx* indicates the number of elapsed seconds and *yyyyyy* indicates the fraction of the current second in microseconds.

SR_SNMP_TEST_PORT

This environment variable sets the master agent's SNMP reception port. Normally, you do not need to specify this environment variable since it is specified in the environment variable definition file `SnmMaster`.

For details about how to change the SNMP reception port, see [3.4.1 Changing the SNMP reception port on SNMP Agent](#).

Definition examples

- This example edits `SNMP_EXTAGT_OPTIONS` of the `SnmExtAgt` file used in the `extsubagt` process:

```
SNMP_EXTAGT_OPTIONS="-aperror -apwarn -apverbose"
export SNMP_EXTAGT_OPTIONS
```

- The following shows a specification example for configuring three startup options of the `extsubagt` process so that they are always enabled.

This example specifies that the `-fcmdguard`, `-pipeguard`, and `-invokeid` startup options of the `extsubagt` process are enabled when the system starts or when the `snmpstart` command is executed.

For Solaris, specify the following options in the `SNMP_EXTAGT_OPTION` environment variable in the environment variable definition file (`/etc/rc.config.d/SnmExtAgt`):

- Command response monitoring interval specified in `FILE_COMMAND`: 15 seconds
- Pipe response monitoring interval specified in `PIPE_IN_NAME` and `PIPE_OUT_NAME`: 25 seconds
- Whether to use the ID in the data match judgment for the data sent or received through a pipe: Yes

```
SNMP_EXTAGT_OPTIONS="-fcmdguard 15 -pipeguard 25 -invokeid"
export SNMP_EXTAGT_OPTIONS
```

These options take effect the next time the system starts or the next time `/opt/CM2/ESA/bin/snmpstart` is executed.

Operating locale definition file (esalocale.conf)

Format

```
LC_ALL=C
export LC_ALL
LANG=C
export LANG
```

Overview

Specify C in the LANG and LC_ALL environment variables as SNMP Agent's operating locale.

When you set up the system language environment, if you specify a value other than C for a locale environment variable that is higher than the LANG environment variable, specify C for the LC_ALL environment variable.

Location

HP-UX (IPF), Solaris, AIX, and Linux: /etc/SnmpAgent.d/esalocale.conf

Detailed description

Specify C in the LANG and LC_ALL environment variables.

```
LC_ALL=C
```

This line substitutes C for LC_ALL.

```
export LC_ALL
```

This line sets C for the LC_ALL environment variable. You cannot set a value other than C. Placing a hash mark (#) before export turns the line into a comment, in which case no value is set. By default, this line is not a comment.

```
LANG=C
```

This line substitutes C for LANG.

```
export LANG
```

This line sets C for the LANG environment variable. You cannot set a value other than C. Placing a hash mark (#) before export turns the line into a comment, and no value is set. By default, this line is not a comment.

Notes

For details about how to specify the operating locale definition file, see [2.6 Setting up the operating locale](#).

Definition examples

This example sets C for the LANG and LC_ALL environment variables:

```
LC_ALL=C
export LC_ALL
LANG=C
export LANG
```

File system definition file (esafilesys.conf)

Format

```
# comment
check: file-system-path-name desired-file-name-located-immediately-under-
file-system-path
exclude: file-system-path-name
```

Overview

This file defines the following settings:

- Setting for checking whether a monitored file system has been mounted
- Specification of file systems to be excluded from monitoring

Location

HP-UX (IPF), Solaris, AIX, and Linux: `/etc/SnmpAgent.d/esafilesys.conf`

Detailed description

`check:`

This argument sets up SNMP Agent to check whether the monitored file system has been mounted, and if it is not mounted, to prevent information on that file system from being acquired. This is specified mainly for cluster system shared disks that are being monitored.

You must insert a single space between `check:` and *file-system-path-name*, and between *file-system-path-name* and *desired-file-name-located-immediately-under-file-system-path*.

- *file-system-path-name*
Specify a file system path name. The maximum number of characters allowed is 1,024.
- *desired-file-name-located-immediately-under-file-system-path*
Specify the name of the desired file located immediately under the file system path. The maximum number of characters allowed is 1,024.

`exclude:`

This option specifies the path name of the file system to be excluded from monitoring. Place one space after `exclude:`.

- *file-system-path-name*
Specify a file system path name. The maximum number of characters allowed is 1,024.

`#`

The remainder of the line starting with a hash mark (#) is treated as a comment. A comment must be expressed as a string of ASCII characters.

Notes

- To specify multiple shared disks, add lines to define them.
- Make sure that the file system path directory of a shared disk located on a local node of a cluster system does not contain the same name as the desired file name located immediately under a share disk, described in the `/etc/SnmpAgent.d/esafilesys.conf` file.

- For the file system path of a shared disk, specify the following:
AIX: The first field (excluding the colon (:)) of `/etc/filesystems`
Linux: The `fs_file` field in the `/etc/fstab` file
- The `esafilesys.conf` setting is enabled for the `fileSystem64` group.
In AIX and Linux, the `/etc/SnmpAgent.d/esafilesys.conf` setting is enabled for both the `fileSystem` group and the `fileSystem64` group. For details about the `/etc/SnmpAgent.d/esafilesys.conf` setting for the `fileSystem64` group, [2.12.2 Settings for suppressing an invalid shared disk capacity response \(for AIX and Linux\)](#).
Note that, in HP-UX (IPF) and Solaris, the `/etc/SnmpAgent.d/esafilesys.conf` setting is not enabled for the `fileSystem` group.

Definition examples

The example below monitors shared disks `/shdisk1` and `/shdisk2`. The `test1` and `test2` files are located immediately under `/shdisk1` and `/shdisk2`, respectively.

```
check: /shdisk1 test1
check: /shdisk2 test2
```

The following example stops `/mnt/cdrom` and `/mnt/floppy` information from being returned:

```
exclude: /mnt/cdrom
exclude: /mnt/floppy
```

Adding file system target definition file (esafstab.conf)

Format

```
# comment  
block_special_device_name mount_point
```

Overview

If there are file systems that you want to monitor as `fileSystem64` group, specify them in the `esafstab.conf` definition file.

Location

Linux: `/etc/SnmpAgent.d/esafstab.conf`

Detailed description

Block special device:

Specify the block special device (corresponding to the first field of `/etc/fstab`) of file the file system you want to add as a target.

The maximum number of characters allowed is 1,024. You can not specify a block special device which includes spaces.

Mount point:

Specify the mount point (corresponding to the second field of `/etc/fstab`) of the file system you want to add as a target.

The maximum number of characters allowed is 1,024. You can not specify a mount point which includes spaces.

#

The remainder of the line starting with a hash mark (#) is treated as a comment. A comment must be expressed as a string of ASCII characters.

Notes

- Add multiple lines if you specify multiple file systems.
- Setting of `esafstab.conf` applies to `fileSystem64` group, and not to `fileSystem` group.

Definition examples

Monitoring mount point: `/mnt/test1` (block special device: `/dev/test`), and mount point: `/mnt/test2` (block special device: `UUID=aaaa44444-55bb-6cc6-7777-ddd888888d8d`)

```
/dev/test /mnt/test1  
UUID=aaaa44444-55bb-6cc6-7777-ddd888888d8d /mnt/test2
```

Setting procedure

Perform the following steps in order to apply the settings in `esafstab.conf`:

1. If SNMP Agent is running, execute the command `/opt/CM2/ESA/bin/snmpstop` as a superuser.
2. Edit `esafstab.conf` with an editor.

3. Execute the command `/opt/CM2/ESA/bin/snmpstart` as a superuser.

Disk definition file (esadisk.conf)

Format

```
# comment
exclude: disk-device-name
check: disk-device-name
```

Overview

If there is a disk device that you want to exclude from retrieval, specify it in the `esadisk.conf` disk definition file.

Location

Linux: `/etc/SnmpAgent.d/esadisk.conf`

Detailed description

Specify the names of the disk devices that you want to exclude from retrieval.

`exclude:`

The information on the disk device that has the specified disk device name is not retrieved. A space is required after `exclude:`. A maximum of 1,033 characters can be specified per line.

- *disk-device-name*

Specify a disk device name. The maximum number of characters allowed is 1,024.

You can specify an asterisk (*) as a wildcard at the end of a disk device name. The asterisk (*) is used to refer to any character string, including an empty string. For example, the device name `sda*` matches `sda`, `sda1`, and so on.

`check:`

Among the disk device names specified in `exclude:`, specify any exceptions that are to be retrieved. A space is required after `check:`. A maximum of 1,033 characters can be specified per line.

- *disk-device-name*

Specify a disk device name. The maximum number of characters allowed is 1,024.

You can specify an asterisk (*) as a wildcard at the end of a disk device name. The asterisk (*) is used to refer to any character string, including an empty string. For example, the device name `sda*` matches `sda`, `sda1`, and so on.

`#`

The remainder of the line starting with a hash mark (#) is treated as a comment. A comment must be expressed as a string of ASCII characters.

Initial settings

The following values are set as initial settings in `esadisk.conf`:

```
exclude: fd*
exclude: loop*
exclude: ram*
exclude: scd*
exclude: sr*
```


- `exclude: fd*` specifies that disk device information for floppy disks is not to be retrieved.
- `exclude: loop*` specifies that disk device information for loopback devices is not to be retrieved.
- `exclude: ram*` specifies that disk device information for RAM disks is not to be retrieved.
- `exclude: scd*` and `exclude: sr*` specify that disk device information for SCSI CD-ROM devices is not to be retrieved.

Definition examples

In an environment where `sda`, `sda1`, `sda2`, and `sda3` can be retrieved, the following example excludes `sda [1-3]` from retrieval:

```
# Get only sda.  
exclude: sda*  
check: sda
```

Setting procedure

Perform the following steps in order to apply the settings in `esadisk.conf`:

1. If SNMP Agent is running, execute the command `/opt/CM2/ESA/bin/snmpstop` as a superuser.
2. Edit `esadisk.conf` with an editor.
3. Execute the command `/opt/CM2/ESA/bin/snmpstart` as a superuser.

7

Troubleshooting

This chapter explains the most probable causes of problems in SNMP Agent and how to deal with such problems.

7.1 General troubleshooting procedure

The following is a general troubleshooting procedure to use when a problem occurs during operation of SNMP Agent.

1. Identify the problem.

Gain a clear understanding of the problem based on the symptoms that are exhibited. For details about how to gain a good understanding of the problem, see [7.2 Identifying the problem](#).

2. Collect logs and data.

SNMP Agent outputs logs on an ongoing basis. For details about the information output to these logs, see [7.3 Collecting logs](#).

Other data must also be collected and reported to the system administrator for troubleshooting. For details about collecting this data, see [7.4 Collecting data](#).

3. Take corrective action.

SNMP is based on UDP. However, UDP does not include error checking and does not guarantee message receipt. Because of this, a problem might occur in communications between SNMP Agent and the SNMP manager. Be aware of this possibility when you attempt to resolve an SNMP Agent problem.

Refer to [7.5 Taking corrective action](#), isolate the location and extent of the problem that has occurred, and take an appropriate action to eliminate the cause.

To resolve a problem in SNMP Agent, refer also to the explanations in the following chapters:

- [1. Introduction to SNMP Agent](#)
- [4. MIB Objects](#)

If the problem is not in SNMP Agent or in the `/etc/SnmpAgent.d/snmpd.extend` file, see the applicable OS documentation.

Note that the path names of the executable files discussed in this chapter vary depending on the OS. For details, see the path names listed in [A. SNMP Agent Files](#) for the OS that you are using.

7.2 Identifying the problem

Gain a clear understanding of the problem based on the symptoms that are exhibited. When you encounter the symptoms of a problem that has occurred, gather the following basic information:

- Location of the problem

Identify where the problem occurred. To do so, determine the following:

- Whether the problem is with the agent or the manager
- Whether the problem is with the agent or in the `/etc/SnmpAgent.d/snmpd.extend` file

Accurately determining whether the problem lies with the agent or the manager is important because a problem with the agent might appear to be a manager problem. Typically, if a problem occurs when a manager sends or receives data via SNMP, the problem is with the agent. For example, invalid information that the manager has about a node on the network was likely sent by an agent.

For details, see [7.5 Taking corrective action](#).

- The part of SNMP Agent that is affected

Determine which part of SNMP Agent is being affected by the problem. Check whether the problem affects all operations or only some operations.

- Impact of the problem

Check whether anything has changed in the network configuration (hardware, software, files, security, utilities, and so on).

- Repeatability of the problem

Check whether the problem is consistent (occurs every time) or intermittent (occurs sometimes).

- Operation when the problem occurs

Check what else was happening when the problem occurred. For instance, check the following:

- Which operation was selected.
- Which command was executed.
- What data was requested or sent.

7.3 Collecting logs

When a problem occurs, system administrators use logs to investigate its cause. If the function for defining extended MIB objects is being used, the system administrator outputs a trace log of the commands executed while acquiring the MIB values to isolate the problem location.

By default, the master agent and subagents normally output logs as follows:

- Log type: Warning and error logs
- Log output destination: `/var/adm/snmpd.logn` (*n*: value indicating the log file count (1 to 10))
- Log file size: 10 megabytes

The following subsections explain how to change the types of logs that are acquired, the output destination for the logs, and the number and size of the log files.

7.3.1 Log type

The types of logs that are acquired can be selected by means of logmasks. Logmasks specify which types of logs are acquired.

The master agent and subagents use different logmasks. Table 7-1 lists the logmasks for the master agent.

Table 7–1: Logmasks for the master agent

Logmask (value specified for <code>snmpdm -m</code>)			Explanation
Character string	Decimal number	Hexadecimal number	
--	0	0x	Log suppression
FACTORY_TRACE	8388608	0x00800000	Trace log output
FACTORY_WARN	268435456	0x10000000	Warning log output
FACTORY_ERROR	536870912	0x20000000	Error log output

Legend:

--: Not applicable

If you do not specify any logmasks for the master agent, the master agent collects error and warning logs.

To change, when the master agent is running, the types of the logs to be collected by the master agent, stop the master agent by executing the `kill` command and restart the master agent with the appropriate logmask(s) specified. For example,

```
/usr/sbin/snmpdm -m 8388608
```

The method for specifying multiple log types depends on whether the logmask values are numbers (decimal or hexadecimal) or character strings. If you use decimal or hexadecimal logmask values, add all the logmask values and specify the sum. If you use character string logmask values, enter each logmask value after `-m`. For example,

```
/usr/sbin/snmpdm -m FACTORY_TRACE FACTORY_WARN FACTORY_ERROR
```

If you specify the `-m` option with a character string in combination with other options, note that the `-m` option is the last option specified.

Table 7-2 lists the logmasks for subagents.

Table 7–2: Logmasks for subagents

Logmask (specified as an option of the subagent command)	Explanation
<code>-aperror</code>	Error logs
<code>-apwarn</code>	Error and warning logs
<code>-aptrace</code>	Trace logs
<code>-apconfig</code>	Logs related to the configuration file
<code>-appacket</code>	Logs related to packet assembly or analysis
<code>-aptrap</code>	Logs related to trap messages
<code>-apaccess</code>	Logs related to agent processing
<code>-apemanate</code>	Logs related to the master agent and subagents
<code>-apverbose</code>	Verbose logs
<code>-apuser</code>	User logs
<code>-apall</code>	All types of logs

If you do not specify any logmasks for a subagent, the subagent will not collect any logs. To make an already running subagent collect some type(s) of logs, stop the subagent by executing the `kill` command, then restart the subagent with the appropriate logmask(s) specified. For example,

```
/usr/sbin/extsubagt -aperror
```

If you want to collect two or more types of logs, specify the logmask values in succession after the subagent command. For example,

```
/usr/sbin/extsubagt -aperror -apwarn -aptrace
```

7.3.2 Log output destination

The output destination for logs can be changed using the `SR_LOG_DIR` environment variable in the `snmpdm` process environment variable definition file (`SnmpMaster`). The file name is set to `snmpd.logn`, and cannot be changed.

For details about the `SR_LOG_DIR` environment variable, see *snmpdm* in [Chapter 5. Commands and Processes](#).

The following procedure shows how to change the output destination for logs.

Procedure

1. Use the `snmpstop` command to stop SNMP Agent.

If the OS being used is Solaris or AIX, and you do not want to shut down the native agent, execute the `snmpstop` command with the `-n` option.

2. Specify the output destination for logs in the `SR_LOG_DIR` environment variable.

Example:

```
SR_LOG_DIR=/tmp/esalog
export SR_LOG_DIR
```

3. Use the `snmpstart` command to restart SNMP Agent.

If the OS being used is Solaris or AIX, and you do not want to restart the native agent, execute the `snmpstart` command with the `-n` option.

By default, the data collection command (`jp1esalog.sh.def`) is set to collect logs from `/var/adm/snmpd.logn`. If you change the log output destination, you must also change the log output destination used during `jp1esalog.sh.def` execution. To change the log output destination when the `jp1esalog.sh.def` command is executed, use a text editor such as `vi` to change the output destination for logs. The following shows an example of editing the `jp1esalog.sh.def` command.

Before change

```
COLFILE="$COLFILE ./var/adm/snmpd.log* ./var/adm/*agt*.log ./etc/
SnmpAgent.d ./etc/srconf/agt"
```

After change

```
COLFILE="$COLFILE ./tmp/esalog/snmpd.log* ./var/adm/*agt*.log ./etc/
SnmpAgent.d ./etc/srconf/agt"
```

Important

Before you edit the `jp1esalog.sh.def` command, always make a backup of the command.

7.3.3 Number and size of the log files

You can use the following environment variables to change the number and size of the log files:

- Use the `SNMP_HTC_SNMPD_LOG_SIZE` environment variable in the `snmpdm` process environment variable definition file (`SnmpMaster`) to change the size of the `snmpd.logn` files (n : 1 to 10). For details about the `SNMP_HTC_SNMPD_LOG_SIZE` environment variable, see *snmpdm* in [Chapter 5. Commands and Processes](#).
- Use the `SNMP_HTC_SNMPD_LOG_CNT` environment variable in the `snmpdm` process environment variable definition file (`SnmpMaster`) to specify the number n of `snmpd.logn` files (n : 1 to 10). For details about the `SNMP_HTC_SNMPD_LOG_CNT` environment variable, see *snmpdm* in [Chapter 5. Commands and Processes](#).

The following procedure shows how to change the number and size of the log files.

Procedure

1. Use the `snmpstop` command to stop SNMP Agent.

If the OS being used is Solaris or AIX, and you do not want to shut down the native agent, execute the `snmpstop` command with the `-n` option.

2. Use the `SNMP_HTC_SNMPD_LOG_SIZE` environment variable to specify the size of the log files.[#]

Example:

```
SNMP_HTC_SNMPD_LOG_SIZE=10
export SNMP_HTC_SNMPD_LOG_SIZE
```

3. Use the `SNMP_HTC_SNMPD_LOG_CNT` environment variable to specify the number of log files.[#]

Example:

```
SNMP_HTC_SNMPD_LOG_CNT=10
export SNMP_HTC_SNMPD_LOG_CNT
```

4. Use the `snmpstart` command to restart SNMP Agent.

If the OS being used is Solaris or AIX, and you do not want to restart the native agent, execute the `snmpstart` command with the `-n` option.

[#]: Steps 2 and 3 can be performed in either order.

7.3.4 Notes about logs

- Logs are used by the system administrator as troubleshooting data. Logs are output to the log files in wraparound form. By default, there are 10 megabytes per file and 10 files that are created, so the log storage destination must be able to accommodate up to 100 megabytes of data.

7.4 Collecting data

By default, SNMP Agent outputs the following data on an ongoing basis:

- A master agent send/receive packet dump (a hexadecimal dump and a VarBind list)
- A native agent adapter send/receive packet dump (a hexadecimal dump and a VarBind list)

The default contents of the master agent send/receive packet dump and the native agent adaptor send/receive packet dump, and the method by which the default settings are changed is the same as for when logs are collected. For details, see [7.3 Collecting logs](#).

You can also collect the following data when a problem occurs in SNMP Agent:

- Master agent detailed trace
- Unauthorized community name logs

7.4.1 Acquiring a master agent send/receive packet dump

A master agent send/receive packet dump is used to investigate what kinds of SNMP messages were sent and received when there is no response to an SNMP request from the manager, or when you suspect that a response is to an invalid SNMP message.

The following figure shows an example of a master agent send/receive packet dump.

Figure 7–1: Example of a master agent send/receive packet dump

```
2012/02/27 13:00:00
snmpdm: Incoming Packet, addr = 10.208.45.232:1526, length = 34:
          at line xxx in file xxx

2012/02/27 13:00:00
snmpdm:  30 20 02 01 00 04 06 70 75 62 6c 69 63 a1 13 02
          at line xxx in file xxx

2012/02/27 13:00:00
snmpdm:  02 29 56 02 01 00 02 01 00 30 07 30 05 06 01 2b
          at line xxx in file xxx

2012/02/27 13:00:00
snmpdm:  05 00
          at line xxx in file xxx

2012/02/27 13:00:00
snmpdm: org = NULL TYPE/VALUE
          at line xxx in file xxx

2012/02/27 13:00:00
snmpdm: Outgoing Packet, addr = 10.208.45.232:1526, length = 84:
          at line xxx in file xxx

2012/02/27 13:00:00
snmpdm:  30 52 02 01 00 04 06 70 75 62 6c 69 63 a2 45 02
          at line xxx in file xxx

2012/02/27 13:00:00
snmpdm:  02 29 56 02 01 00 02 01 00 30 39 30 37 06 08 2b
          at line xxx in file xxx

2012/02/27 13:00:00
snmpdm:  06 01 02 01 01 01 00 04 2b 53 75 6e 4f 53 20 6e
          at line xxx in file xxx

2012/02/27 13:00:00
snmpdm:  65 74 6d 73 75 6e 31 31 20 35 2e 39 20 47 65 6e
          at line xxx in file xxx

2012/02/27 13:00:00
snmpdm:  65 72 69 63 5f 31 31 38 35 35 38 2d 32 36 20 73
          at line xxx in file xxx

2012/02/27 13:00:00
snmpdm:  75 6e 34 75
          at line xxx in file xxx

2012/02/27 13:00:00
snmpdm: sysDescr.0 = SunOS netmsun11 5.9 Generic_118558-26 sun4u
          at line xxx in file xxx
```

Indicates that snmpdm received a request (Incoming Packet) of length 34 bytes from the manager (IP address: 11.22.33.44, port number: 1526) at 13:00:00 on 2012/02/27.

Indicates that snmpdm sent a response (Outgoing Packet) of length 84 bytes to the manager (IP address: 11.22.33.44, port number: 1526) at 13:00:00 on 2012/02/27. You can see that the value of sysDescr.0 was sent.

7.4.2 Acquiring a native agent adapter send/receive packet dump

A native agent adapter send/receive packet dump is used to investigate what kinds of SNMP messages were sent and received regarding the acquisition and configuration of standard MIB objects, either when there is no response to an SNMP request from the manager or when you suspect that a response is to an invalid SNMP message.

The following figure shows an example of a native agent adapter send/receive packet dump.

Figure 7–2: Example of native agent adapter send/receive packet dump

```
naaagt: Outgoing Packet, length = 41:
                at line xxx in file xxx
2012/02/26 20:12:41
naaagt:  30 27 02 01 00 04 06 70 75 62 6c 69 63 a1 1a 02
                at line xxx in file xxx
2012/02/26 20:12:41
naaagt:  01 02 02 01 00 02 01 00 30 0f 30 0d 06 09 2b 06
                at line xxx in file xxx
2012/02/26 20:12:41
naaagt:  01 02 01 04 14 01 03 05 00
                at line xxx in file xxx
2012/02/26 20:12:41
naaagt:  iso.3.6.1.2.1.4.20.1.3 = NULL TYPE/VALUE
                at line xxx in file xxx
2012/02/26 20:12:41
naaagt: Incoming Packet, length = 50:
                at line xxx in file xxx
2012/02/26 20:12:41
naaagt:  30 30 02 01 00 04 06 70 75 62 6c 69 63 a2 23 02
                at line xxx in file xxx
2012/02/26 20:12:41
naaagt:  01 02 02 01 00 02 01 00 30 18 30 16 06 0e 2b 06
                at line xxx in file xxx
2012/02/26 20:12:41
naaagt:  01 02 01 04 14 01 03 0a 81 50 6c 4c 40 04 ff ff
                at line xxx in file xxx
2012/02/26 20:12:41
naaagt:  ff 00
                at line xxx in file xxx
2012/02/26 20:12:41
naaagt:  iso.3.6.1.2.1.4.20.1.3.11.22.33.44 = 255.255.255.0
                at line xxx in file xxx
```

Indicates that naaagt sent a request (Outgoing Packet) of length 41 bytes to the native agent at 20:12:41 on 2012/02/26.

Indicates that naaagt received a response (Incoming Packet) of length 50 bytes from the native agent at 20:12:41 on 2012/02/26. You can see that a value of iso.3.6.1.2.1.4.20.1.3.11.22.33.44 was received.

Important

On HP-UX (IPF), a native agent adapter is not provided even though SNMP Agent is a native agent. Therefore, native agent adapter send/receive packet dumps are not available on HP-UX (IPF).

7.4.3 Acquiring detailed trace information about the master agent

The detailed trace information about the master agent includes the following information:

- What SNMP requests and responses were made, and which requests and responses succeeded (or failed) between the master agent and manager
- What MIB value requests and responses were made, and which requests and responses succeeded (or failed) between the master agent and subagent

You can use the detailed trace information output for the master agent to determine the cause of a communication failure between the master agent and the manager or between the master agent and a subagent. The following procedure shows how to acquire detailed trace information about the master agent.

Procedure

1. Stop `snmpdm`.

Execute the following command to send a termination signal to the `snmpdm` process:

```
Kill -15 snmpdm-process-ID
```

Use the following method to obtain the `snmpdm` process ID.

Execute the following command:

```
/opt/CM2/ESA/bin/snmpcheck
```

Execution of this command displays a list of SNMP Agent processes similar to the following. The value listed for `pid=` in the `snmpdm` line is its process ID.

```
snmpdm    running pid=11293
mib2agt   running pid=11330
hp_unixagt    running pid=11331
ipv6agt   running pid=11334
trapdestagt    running pid=11348
extsubagt  running pid=11384
htc_unixagt1    running pid=11366
htc_unixagt2    running pid=11367
htc_unixagt4    running pid=11368
```

2. Start the master agent.

The logs can be output to any file. For details about the log output destination, see [7.3.2 Log output destination](#) in [7.3 Collecting logs](#).

```
SR_SNMP_TEST_PORT=port-number-of-SNMP-Agent (typically, 161)
export SR_SNMP_TEST_PORT
/usr/sbin/snmpdm -tcplocal -hexdump -vbdump -apall -n >> /tmp/
esa.packet.log 2>&1 &
```

3. Start `mib2agt` and `hp_unixagt` (for HP-UX (IPF)).

```
/sbin/init.d/SnmpMib2 start
/sbin/init.d/SnmpIpv6 start
/sbin/init.d/SnmpHpunix start
```

4. Verify that `snmpdm` is running.

```
/opt/CM2/ESA/bin/snmpcheck
```

Important

`/tmp/esa.packet.log` is a file that is continually added to. If you are monitoring file sizes and the size of this file grows too large, you can delete it as follows (for Solaris, run in `ksh`):

```
cat /dev/null > /tmp/esa.packet.log
```

For Solaris and AIX, if the user logs out of the terminal on which this command was executed while detailed trace information is being acquired, a HUP signal is sent to the process in which this command was generated, which might prevent detailed trace information from being acquired. This HUP event does not occur if you do not log out of the terminal. If you must log out of the terminal, use the following method:

After you execute the detailed trace information command, find its process ID, and then execute the following command:

```
/usr/bin/nohup -p -a "PID-of-process"
```

For details about the `nohup` command, see the *man* pages.

7.4.4 Collecting logs of unauthorized community names

When the community name of the SNMP request reported by the manager does not match the community name defined at SNMP Agent, you can display the community name and sending source IP address for that SNMP request.

The output example indicated below is for an instance in which the community name sent by the SNMP request at community name `bad-comm` from the `10.111.98.36` node is invalid.

Note:

When the community name consists of more than 255 characters, a message telling the user to limit it to 255 characters will be displayed.

```
2012/02/26 19:05:11
Authentication failure, bad community string
Message from 10.111.98.36:2869, community = bad-comm.
```

The following indicates the configuration method used to acquire the sending source IP address and community name for SNMP requests when the community name is invalid. You must be logged in as a superuser to execute the following tasks. The default setting does not permit SNMP Agent to acquire this log.

Procedure

1. Edit the files shown below using an editor such as `vi`.

```
/etc/rc.config.d/SnmpMaster (For Solaris)
/usr/CM2/ESA/opt/SnmpMaster (For AIX)
/opt/CM2/ESA/opt/SnmpMaster (for HP-UX (IPF), Linux)
```

2. Add the following rows to the final rows of the file.

```
SNMP_HTC_AUTH_LOG=1
export SNMP_HTC_AUTH_LOG
```

3. Save the file.

4. Restart SNMP Agent.

Execute the following command:

```
/opt/CM2/ESA/bin/snmpstart (For systems other than AIX)
/usr/CM2/ESA/bin/snmpstart (For AIX)
```

Important

The `snmpstart` command starts SNMP Agent after stopping it for a short period of time. If the OS being used is Solaris or AIX, and you do not want to restart the native agent, execute the `snmpstart` command with the `-n` option.

7.5 Taking corrective action

This section explains how to handle the following types of problems:

- Problems when SNMP Agent is starting
- Problems when SNMP Agent is running

During normal use, take note of the following to help ensure that SNMP Agent runs normally:

- Make sure that SNMP Agent setup satisfies the software and hardware requirements.
- Do not change SNMP Agent files such as `/etc/SnmpAgent.d/snmpd.conf` and `/etc/SnmpAgent.d/snmpd.extend` without first backing up the original files. Typically, a backup of the original files is made, and the backup is then used for normal operations. If a problem occurs in a file, you can restore the normal settings by using the original file. The original files are in the `/opt/OV/newconfig/EAGENT-RUN` directory.

7.5.1 Problems when SNMP Agent is starting

If you cannot start SNMP Agent, check the following:

- Software version and file access permissions
By default, on the master agent and subagents, access permissions to SNMP Agent files can be executed only by `root`.
- SNMP reception port number
If the SNMP reception port number of SNMP Agent conflicts with the SNMP reception port number of a native agent or an SNMP agent of another company, change the SNMP reception port of SNMP Agent as described in [3.4 Changing the SNMP reception port on SNMP Agent](#).
- Environment variable definition files
If an environment variable definition file is backed up with a file name that begins with `Snmp`, the backup file could be mistakenly read as an environment variable definition file. For details about what to be aware of when backing up the environment variable definition file, see [3.7 Notes about operations](#).

7.6 Problems when SNMP Agent is running

This subsection explains the action to take when one of the following problems occurs while SNMP Agent is running:

- MIB values cannot be acquired.
- SNMP traps do not reach the manager.
- SNMP Agent extended function cannot be used.

7.6.1 MIB values cannot be acquired

If SNMP Agent cannot acquire MIB values from the manager product even though it is running, check the following:

- Check whether SNMP Agent and the manager can communicate with each other. If not, the problem might be with the network configuration. Execute the `ping` command to check the network connections.
- To determine the MIB values for SNMP Agent, use the `snmpget` command provided by NNM.
- To collect all or some of the dumps from the SNMP Agent MIB groups for inspection, use the `snmpwalk` command provided by NNM.
- Check whether the object ID set in SNMP Agent matches the object ID set in the manager.
- If an attempt is being made to execute an SNMP `SetRequest`, check whether SNMP Agent is configured to respond to SNMP `SetRequests`. By default, the manager itself cannot change an SNMP Agent MIB value. To configure SNMP Agent to respond to SNMP `SetRequests`, add a `set` community name to SNMP Agent's `/etc/SnmpAgent.d/snmpd.conf` file.
- To check whether information retrieval is correct, use the **MIB Browser: SNMP** operation.
- If a MIB value provided by the native agent on Solaris, AIX, or Linux cannot be acquired, check whether the community name of the native agent adapter of SNMP Agent is the same as the community name of the native agent provided by the OS.

In the JP1/SSO resources, the MIB values provided by the native agent are the network summary, line utilization rate, interface traffic, IP traffic, ICMP traffic, TCP traffic, and UDP traffic.

- If you are not able to acquire any of the MIB values provided by SNMP Agent, perform the following procedure:

1. Check whether the community name of SNMP Agent is the same as that of the manager product.

If you change the community name of SNMP Agent, make sure that you restart SNMP Agent or the OS.

2. Check whether a firewall exists between the manager product and SNMP Agent, and, if so, check whether SNMP communications are permitted.

If necessary, reconfigure the firewall settings.

You can also use the OS-provided packet trace command to acquire a packet trace to check whether SNMP requests are reaching the OS.

For details about how to use the OS-provided packet trace command, see the *man* pages for the relevant command.

HP-UX(IPF): `nettl` command

Solaris: `snoop` command

AIX: `iptrace` command

Linux: `tcpdump` command

- If a MIB value cannot be acquired at times due to a timeout or a `noSuchName` error

This type of event occurs if the timeout period for SNMP requests from the manager product is too short.

Refer to [3.7 Notes about operations](#) to determine an appropriate timeout period.

- If a MIB value cannot be acquired due to some other cause

A command that SNMP Agent uses to acquire MIB values is not installed on Solaris, AIX, or Linux.

Execute the `/opt/CM2/ESA/bin/snmpcmdchk` command to check whether all required commands are installed. For details, see [2.2.2 Installing the commands used to acquire MIB values \(for an OS other than HP-UX \(IPF\)\)](#).

7.6.2 SNMP traps do not reach the manager

If SNMP traps do not reach the manager even though SNMP Agent is running, check the following. For details about problems between the manager system and the SNMP Agent system, see [7.6.1 MIB values cannot be acquired](#).

- Check whether the trap destination for SNMP Agent is set correctly. For details, see [2.7.6 Specifying trap destinations](#).
- Using SNMP commands provided by NNM, check the SNMP operation on the remote SNMP node.

7.6.3 The SNMP Agent extended function cannot be used

If you cannot use the SNMP Agent extended function even though SNMP Agent is running, it might be due to one of the reasons listed below. For details about problems between the manager system and the SNMP Agent system, see [7.6.1 MIB values cannot be acquired](#).

Problems in the `/etc/SnmpAgent.d/snmpd.extend` file

When troubleshooting problems in the `/etc/SnmpAgent.d/snmpd.extend` file, first perform checks of limited scope, and then perform checks on the network as a whole. Use the general methodology outlined below to resolve problems in the `/etc/SnmpAgent.d/snmpd.extend` file.

- To check whether the command responses are correct, execute from the OS command line each command in the `/etc/SnmpAgent.d/snmpd.extend` file.
- Use the following command to check whether the command executed correctly:
`echo $?`
- If the command includes arguments, check the arguments. To do so, manually specify all of the required parameters when you execute the command.
- Syntax errors are output to standard error while `extsubagt` is active.
- If SNMP Agent detects an error while reading the `/etc/SnmpAgent.d/snmpd.extend` file, it displays the line on which the error occurred, along with the correct syntax.
- Check whether the commands defined in the `/etc/SnmpAgent.d/snmpd.extend` file can be executed.
- Check whether permissions for executing the command are set.
- Using the identifier of the object used to acquire the information, check whether the correct commands are executing. To check which commands are being executed for the object, specify a logmask as follows:
`/usr/sbin/extsubagt -apall`
- Check whether the output of the executed command is the correct data type.
- Check whether the commands in the `/etc/SnmpAgent.d/snmpd.extend` file are correctly specified. For example, `/usr` might have been mistakenly typed as `/user`.

Problems originating from the manager

If a problem persists after you have resolved any problem found in the agent, check the following:

- Issue an SNMP request from the manager to each object in the `/etc/SnmpAgent.d/snmpd.extend` file to determine whether the file is functioning correctly.
- Issue an SNMP `SetRequest`, and then issue an SNMP `GetRequest` to check whether the value was set correctly.

7.7 Method for collecting log information

When a problem occurs in SNMP Agent, use the data collection tool so that you can quickly collect the data necessary for troubleshooting. For details, see *jp1esalog.sh.def* in *Chapter 5. Commands and Processes*.

Appendixes

A. SNMP Agent Files

This appendix contains lists of SNMP Agent files, arranged by operating systems.

This appendix contains the following sections:

- *A.1 List of SNMP Agent files (HP-UX (IPF))*
- *A.2 List of SNMP Agent files (Solaris)*
- *A.3 List of SNMP Agent files (AIX)*
- *A.4 List of SNMP Agent files (Linux)*

A.1 List of SNMP Agent files (HP-UX (IPF))

Category	Path name	File name
Load modules	/opt/CM2/ESA/bin	snmpdm
		extsubagt
		htc_unixagt1
		htc_unixagt2
		trapdestagt
		trapsend
		htc_unixagt4
	/opt/OV/bin	snmptrap
	systemtrap	
Starting and stopping commands	/opt/CM2/ESA/bin	snmpstart
		snmpstop
		snmpcheck
Log collection command	/opt/CM2/ESA/bin	jplesalog.sh.def
Communication between the master agent and subagents	/tmp/.AgentSockets	Files under this directory
Patch file history	/opt/CM2/ESA	esa_spackinfo
Configuration files	/etc/SnmpAgent.d	snmpd.conf
	/etc/srconf/agt	snmpd.cnf
		snmpd.cnf~
		snmpd.jnk
	/etc/srconf/mgr	snmpinfo.dat
		mgr.cnf
/etc/opt/OV/share/conf	opConfCharCode.conf	

Category	Path name	File name
		snmpmib
		snmpmib.bin
	/opt/CM2/ESA/ext	Files under this directory
Extended MIB definition samples	/opt/OV/prg_samples/eagent	snmpd.extend
		change_num_widgets
		get_processes
		list_processes
		memory.curly
		memory.larry
		memory.moe
		memory.public
		num_widgets
		root_processes
		update_inetd
user_disk_space		
MIB definition files	/opt/CM2/ESA/snmp_mibs	rfc1213-MIB-II
		hp-unix
		hitachi-cometAgt
		hitachi-cometAgt-aix
		hitachi-cometAgt-solaris
		hitachi-cometAgt-linux
		hitachi-cometAgt-tru64
File executed at startup and shutdown	/sbin/init.d	esa
File executed at shutdown (symbolic link)	/sbin/rc1.d	K440esa
File executed at startup (symbolic link)	/sbin/rc2.d	S560esa
Environment variable definition files	/opt/CM2/ESA/opt	SnmpMaster
		SnmpMib2
		SnmpHpunix
		SnmpTrpDst
		SnmpExtAgt
		SnmpHtcunix1
		SnmpHtcunix2
SnmpIpv6		

Category	Path name	File name
		SnmpHtcunix4
Installation information	/etc/.hitachi/before	before_G12B
	/etc/.hitachi/after	after_G12B
	/etc/.hitachi/remove	remove_G12B
Installation log	/tmp	esa.log
Backup installation files	/opt/CM2/ESA/newconfig	SNMPD.CNF
		esa
		init_new_SnmpHpunix
		init_new_SnmpMib2
		oracle_new_snmpd
		esafilesys.conf
		snmpd.cnf
		snmpd.conf
		snmpmib
		snmpmib.bin
		snmptrap
		systemtrap
		snmpinfo.dat
		mgr.cnf
	SNMPINFO.DAT	
	MGR.CNF	
	/opt/CM2/ESA/newconfig/rc.config.d	SnmpMaster
		SnmpMib2
		SnmpHpunix
		SnmpTrpDst
SnmpExtAgt		
SnmpHtcunix1		
SnmpHtcunix2		
SnmpIpv6		
SnmpHtcunix4		
Log files	/var/adm	snmpd.log ⁿ
File system definition file	/etc/SnmpAgent.d	esafilesys.conf
File system definition file (error)	/etc/SnmpAgent.d	esafilesys.conf.err

Category	Path name	File name
New installation check file	/opt/CM2/ESA/bin	INSTALLED
Operating locale definition file	/etc/SnmpAgent.d	esalocale.conf

#: The value of *n* is between 1 and the value specified in the `SNMP_HTC_SNMPD_LOG_CNT` environment variable. If no value is specified in the `SNMP_HTC_SNMPD_LOG_CNT` environment variable, the maximum value (10) is assumed.

A.2 List of SNMP Agent files (Solaris)

Category	Path name	File name
Load modules	/usr/sbin (symbolic link)	extsubagt
		/opt/CM2/ESA/bin
		snmpdm
		extsubagt
		hp_unixagt
		htc_unixagt1
		naaagt
		trapdestagt
		htc_unixagt3
		htc_monagt1
		trapsend
		htc_unixagt4
		/opt/OV/bin
	snmptrap	
	systemtrap	
Starting and stopping commands	/opt/CM2/ESA/bin	snmpstart
		snmpstop
		snmpcheck
Log collection command	/opt/CM2/ESA/bin	jplesalog.sh.def
MIB value acquisition commands	/opt/CM2/ESA/bin	cpuutil.exe
		freememory.exe
		physmemory.exe
		swapconfig.exe
		diskBusy.exe
		cpuInfo.exe
		processorCpuTime.exe
diskTime.exe		

Category	Path name	File name
MIB value creation directory	/opt/CM2/ESA/out	cpuutil.out
		freememory.out
		physmemory.out
		swapconfig.out
		diskBusy.out
		cpuInfo.out
		processorCpuTime.out
		diskTime.out
		temp_processorCpuTime.out
		mpstat.out
		mpstat.err.tmp
		mpstat.err
		mpstat.all
		freememory64.out
		physmemory64.out
swapconfig64.out		
Communication between the master agent and subagents	/tmp/.AgentSockets	Files under this directory
Patch file history	/opt/CM2/ESA	esa_spackinfo
Socket communication	/opt/CM2/ESA/sockets	agt3_mon1
		mon1_comm
Configuration files	/etc/SnmpAgent.d	snmpd.conf
	/etc	snmpd.conf (symbolic link)
	/etc/srconf/agt	naa.cnf
		snmpd.cnf
		snmpd.cnf~
		snmpd.jnk
	/etc/srconf/mgr	snmpinfo.dat
		mgr.cnf
	/etc/opt/OV/share/conf	opConfCharCode.conf
		snmpmib
		snmpmib.bin
	/opt/CM2/ESA/ext	Files under this directory
Extended MIB definition samples	/opt/OV/prg_samples/eagent	snmpd.extend
		change_num_widgets

Category	Path name	File name
		get_processes
		list_processes
		memory.curly
		memory.larry
		memory.moe
		memory.public
		num_widgets
		root_processes
		update_inetd
		user_disk_space
MIB definition files	/var/opt/OV/share/snmp_mibs/ eagent	rfc1213-MIB-II
		hp-unix
		hitachi-cometAgt
		hitachi-cometAgt-aix
		hitachi-cometAgt-solaris
		hitachi-cometAgt-linux
		hitachi-cometAgt-tru64
Files executed at startup and shutdown	/sbin/init.d	esa
	/opt/CM2/ESA/init.d	SnmpMaster
		SnmpHpunix
		SnmpTrpDst
		SnmpHtcunix1
		SnmpExtSubagent
		SnmpNaa
		SnmpHtcunix3
		SnmpHtcmonagt1
SnmpHtcunix4		
Files executed at shutdown (symbolic link)	/etc/rc0.d	K02esa
	/etc/rc1.d	K02esa
File executed at startup (symbolic link)	/etc/rc2.d	S97esa
Environment variable definition files	/etc/rc.config.d	SnmpMaster
		SnmpHpunix
		SnmpTrpDst
		SnmpNaa

Category	Path name	File name
		SnmpHtcmonagt1
		SnmpHtcunix1
		SnmpHtcunix3
		SnmpExtAgt
		SnmpHtcunix4
Installation information	/etc/.hitachi/before	before_5700
	/etc/.hitachi/after	after_5700
	/etc/.hitachi/remove	remove_5700
Installation log	/tmp	esa.log
Backup installation files	/opt/OV/newconfig/EAGENT-RUN	snmpdm
		hp_unixagt
		trapdestagt
		extsubagt
		naaagt
		htc_unixagt1
		snmptrap
		snmpmib
		snmpmib.bin
		systemtrap
		snmpd.conf
		esafilesys.conf
		snmpd.cnf
		naa.cnf
		SnmpExtSubagent
		SnmpHtcunix1
		config_new_SnmpMaster
		config_new_SnmpHpunix
		config_new_SnmpTrpDst
		config_new_SnmpNaa
		rc.config
		init_new_SnmpMaster
		init_new_SnmpHpunix
init_new_SnmpTrpDst		
init_new_SnmpNaa		
htc_unixagt3		

Category	Path name	File name
		htc_monagt1
		SnmpHtcunix3
		SnmpHtcmonagt1
		SVC-NET-SNMP (Solaris 11)
		svc-net-snmp.tmp (Solaris 11)
		trapsend
		snmpinfo.dat
		mgr.cnf
		htc_unixagt4
		SnmpHtcunix4
	/opt/OV/newconfig/EAGENT-RUN/rc.config.d/	SnmpExtAgt
		SnmpHtcmonagt1
		SnmpHtcunix1
		SnmpHtcunix3
		SnmpHtcunix4
Log files	/var/adm/	snmpd.log $n^{\#}$
	/var/opt/CM2/ESA/log	htc_monagt1.log
		htc_monagt1.log.old
Symbolic link files	/usr/OV/bin	snmptrap
		systemtrap
	/usr/OV/conf	snmpmib.bin
File system definition file	/etc/SnmpAgent.d	esafilesys.conf
File system definition file (error)	/etc/SnmpAgent.d	esafilesys.conf.err
OS command installation verification	/opt/CM2/ESA/bin	snmpcmdchk
Operating locale definition file	/etc/SnmpAgent.d	esalocale.conf

$\#$: The value of n is between 1 and the value specified in the `SNMP_HTC_SNMPD_LOG_CNT` environment variable. If no value is specified in the `SNMP_HTC_SNMPD_LOG_CNT` environment variable, the maximum value (10) is assumed.

A.3 List of SNMP Agent files (AIX)

Category	Path name	File name
Load modules	/usr/sbin	snmpdm
		extsubagt

Category	Path name	File name
		hp_unixagt
		htc_unixagt1
		naaagt
		trapdestagt
		htc_unixagt3
		htc_monagt1
		htc_unixagt4
	/usr/OV/bin	snmptrap
		systemtrap
	/usr/CM2/ESA/bin	trapsend
Starting and stopping commands	/usr/CM2/ESA/bin	snmpstart
		snmpstop
		snmpcheck
Log collection command	/usr/CM2/ESA/bin	jplesalog.sh.def
MIB value acquisition command	/usr/CM2/ESA/bin	disk.exe
		freememory.exe
		loadave.exe
		page.exe
		physmemory.exe
		process.exe
		reconfigure.exe
		swapconfig.exe
		systemactive.exe
		usingmemory.exe
		vmactive.exe
		vmforks.exe
		vmtotal.exe
		cpuInfo.exe
		processorCpuTime.exe
diskTime.exe		
MIB value creation directory	/usr/CM2/ESA/out	freememory.out
		loadave.out
		pageSize.out
		physmemory.out
		process.out

Category	Path name	File name
		swapconfig.out
		systemactive.out
		vmtotal.out
		cpuInfo.out
		vmactive64.out
		vmtotal64.out
		vmforks64.out
		processorCpuTime.out
		diskTime.out
		page.out
		disk.out
		vmactive.out
		vmforks.out
		h_process.out
		htc_swapconfig.out
		temp_processorCpuTime.out
		temp_vmtotal.out
		mpstat.out
		mpstat.err.tmp
		mpstat.err
		mpstat.all
		freememory64.out
		physmemory64.out
		swapconfig64.out
		disk64.out
Communication between the master agent and subagents	/tmp/.AgentSockets	Files under this directory
Patch file history	/usr/CM2/ESA	esa_spackinfo
Socket communication	/usr/CM2/ESA/sockets	agt3_mon1
		mon1_comm
Configuration files	/etc/SnmpAgent.d	snmpd.conf
	/etc/srconf/agt	snmpd.cnf
		naa.cnf
		snmpd.cnf~
		snmpd.jnk

Category	Path name	File name
	/etc/srconf/mgr	snmpinfo.dat
		mgr.cnf
	/usr/OV/conf	opConfCharCode.conf
		snmpmib
		snmpmib.bin
	/usr/CM2/ESA/ext	Files under this directory
	Extended MIB definition samples	/usr/OV/prg_samples/eagent
change_num_widgets		
get_processes		
list_processes		
memory.curly		
memory.larry		
memory.moe		
memory.public		
num_widgets		
root_processes		
update_inetd		
user_disk_space		
MIB definition files	/usr/OV/snmp_mibs/eagent	rfc1213-MIB-II
		hp-unix
		hitachi-cometAgt
		hitachi-cometAgt-aix
		hitachi-cometAgt-solaris
		hitachi-cometAgt-linux
		hitachi-cometAgt-tru64
Files executed at startup	Saved under /etc/inittab	--
	/usr/CM2/ESA/bin	esa
Files executed at shutdown	/etc/rc.shutdown	--
Environment variable definition files	/usr/CM2/ESA/opt	SnmpMaster
		SnmpHpunix
		SnmpTrpDst
		SnmpNaa
		SnmpNative
		SnmpHtcmonagt1
		SnmpHtcunix1

Category	Path name	File name
		SnmpHtcunix3
		SnmpExtAgt
		SnmpHtcunix4
Installation information	/etc/.hitachi/before	before_112B
	/etc/.hitachi/after	after_112B
	/etc/.hitachi/remove	remove_112B
Installation log	/tmp	esa.log
Backup installation files	/usr/OV/newconfig/EAGENT-RUN	snmpdm
		hp_unixagt
		trapdestagt
		extsubagt
		naaagt
		htc_unixagt1
		htc_unixagt3
		htc_monagt1
		snmptrap
		snmpmib
		snmpmib.bin
		systemtrap
		snmpd.conf
		esafilesys.conf
		snmpd.cnf
		naa.cnf
		trapsend
		snmpinfo.dat
		mgr.cnf
		htc_unixagt4
	/usr/OV/newconfig/EAGENT-RUN/rc.config.d	SnmpExtAgt
		SnmpMaster
		SnmpMib2
		SnmpHpunix
		SnmpTrpDst
		SnmpNaa
		SnmpHtcunix1
SnmpHtcunix3		

Category	Path name	File name
		SnmpHtcumonagt1
		SnmpNative
		SnmpHtcunix4
Log files	/usr/adm/	snmpd.log $n^{\#}$
	/usr/CM2/ESA/log	htc_monagt1.log
		htc_monagt1.log.old
File system definition file	/etc/SnmpAgent.d	esafilesys.conf
File system definition file (error)	/etc/SnmpAgent.d	esafilesys.conf.err
OS command installation verification	/usr/CM2/ESA/bin	snmpcmdchk
Operating locale definition file	/etc/SnmpAgent.d	esalocale.conf

Legend:

--: Not applicable

#: The value of n is between 1 and the value specified in the `SNMP_HTC_SNMPD_LOG_CNT` environment variable. If no value is specified in the `SNMP_HTC_SNMPD_LOG_CNT` environment variable, the maximum value (10) is assumed.

A.4 List of SNMP Agent files (Linux)

Category	Path name	File name
Load modules	/usr/sbin	snmpdm
		extsubagt
		hp_unixagt
		htc_unixagt1
		naaagt
		trapdestagt
		htc_unixagt3
		htc_monagt1
		htc_unixagt4
	/opt/OV/bin	snmptrap
		systemtrap
/opt/CM2/ESA/bin	trapsend	
Starting and stopping commands	/opt/CM2/ESA/bin	snmpstart
		snmpstop
		snmpcheck

Category	Path name	File name
Log collection command	/opt/CM2/ESA/bin	jplesalog.sh.def
MIB value acquisition commands	/opt/CM2/ESA/bin	linuxSystem.exe
		loadave.exe
		process.exe
		cpuInfo.exe
		linuxPhysMem.exe
		linuxSwap.exe
MIB value creation directory	/opt/CM2/ESA/out	linuxSystem.out
		loadave.out
		process.out
		cpuInfo.out
		linuxPhysMem.out
		linuxSwap.out
		mpstat.out
		mpstat.err.tmp
		mpstat.err
		mpstat.all
		linuxPhysMem64.out
linuxSwap64.out		
Communication between the master agent and subagents	/tmp/.AgentSockets	Files under this directory
Patch file history	/opt/CM2/ESA	esa_spackinfo
Configuration files	/etc/SnmpAgent.d	snmpd.conf
	/etc	snmpd.conf (symbolic link)
	/etc/srconf/agt	naa.cnf
		snmpd.cnf
		snmpd.cnf~
		snmpd.jnk
	/etc/srconf/mgr	snmpinfo.dat
		mgr.cnf
	/etc/opt/OV/share/conf	opConfCharCode.conf
		snmpmib
		snmpmib.bin
/opt/CM2/ESA/ext	Files under this directory	

Category	Path name	File name
Extended MIB definition samples	/opt/OV/prg_samples/eagent	snmpd.extend
		change_num_widgets
		get_processes
		list_processes
		memory.curly
		memory.larry
		memory.moe
		memory.public
		num_widgets
		root_processes
		update_inetd
		user_disk_space
MIB definition files	/var/opt/OV/share/snmp_mibs/eagent	rfc1213-MIB-II
		hp-unix
		hitachi-cometAgt
		hitachi-cometAgt-aix
		hitachi-cometAgt-solaris
		hitachi-cometAgt-linux
		hitachi-cometAgt-tru64
Socket communication	/opt/CM2/ESA/sockets	agt3_mon1
		mon1_comm
Files executed at startup and shutdown (common)	/opt/CM2/ESA/bin	sub_snmpstart
		sub_snmpstop
Files executed at startup and shutdown (in RHEL 6, CentOS 6, and Oracle Linux 6)	/etc/rc.d/init.d	esa
Files executed at shutdown (symbolic link) (in RHEL 6, CentOS 6, and Oracle Linux 6)	/etc/rc.d/rc0.d	K65esa
	/etc/rc.d/rc2.d	K65esa
	/etc/rc.d/rc3.d	K65esa
	/etc/rc.d/rc5.d	K65esa
	/etc/rc.d/rc6.d	K65esa
File executed at startup (symbolic link) (in RHEL 6, CentOS 6, and Oracle Linux 6)	/etc/rc.d/rc2.d	S55esa
	/etc/rc.d/rc3.d	S55esa
	/etc/rc.d/rc5.d	S55esa
Files executed at startup and shutdown	/opt/CM2/ESA/bin	jpl_esa

Category	Path name	File name
(in RHEL 7, CentOS 7, Oracle Linux 7, and SUSE Linux 12)		
Service configuration file (in RHEL 7, CentOS 7, Oracle Linux 7, and SUSE Linux 12)	/usr/lib/systemd/system	jpl_esa.service
Environment variable definition files	/opt/CM2/ESA/opt	SnmpMaster
		SnmpHpunix
		SnmpTrpDst
		SnmpNaa
		SnmpHtcmonagt1
		SnmpHtcunix1
		SnmpHtcunix3
		SnmpExtAgt
Installation information	/etc/.hitachi/before	before_112B
	/etc/.hitachi/after	after_112B
	/etc/.hitachi/remove	remove_112B
Installation log	/tmp	esa.log
Backup installation files	/opt/OV/newconfig/EAGENT-RUN	snmpdm
		hp_unixagt
		trapdestagt
		extsubagt
		naaagt
		htc_unixagt1
		snmptrap
		snmpmib
		snmpmib.bin
		systemtrap
		snmpd.conf
		esafilesys.conf
		esadisk.conf
		snmpd.cnf
		naa.cnf
esa		
htc_monagt1		

Category	Path name	File name
		htc_unixagt3
		trapsend
		snmpinfo.dat
		mgr.cnf
		htc_unixagt4
	/opt/OV/newconfig/EAGENT-RUN/rc.config.d	SnmpExtAgt
		SnmpMaster
		SnmpMib2
		SnmpHpunix
		SnmpTrpDst
		SnmpNaa
		SnmpHtcunix1
		SnmpHtcunix3
		SnmpHtcmonagt1
SnmpHtcunix4		
Log files	/var/adm/	snmpd.log $n^{\#}$
	/var/opt/CM2/ESA/log	htc_monagt1.log
		htc_monagt1.log.old
		esastart.log
File system definition file	/etc/SnmpAgent.d	esafsys.conf
Disk definition file	/etc/SnmpAgent.d	esadisk.conf
File system definition file (error)	/etc/SnmpAgent.d	esafsys.conf.err
Adding file system target definition file	/etc/SnmpAgent.d	esafstab.conf
Adding file system target definition file (error)	/etc/SnmpAgent.d	esafstab.conf.err
Disk definition file (error)	/etc/SnmpAgent.d	esadisk.conf.err
File system information storage file	/opt/CM2/ESA/conf	fileSystemID.db fileSystem64ID.db
OS command installation verification	/opt/CM2/ESA/bin	snmpcmdchk
Operating locale definition file	/etc/SnmpAgent.d	esalocale.conf

$\#$: The value of n is between 1 and the value specified in the `SNMP_HTC_SNMPD_LOG_CNT` environment variable. If no value is specified in the `SNMP_HTC_SNMPD_LOG_CNT` environment variable, the maximum value (10) is assumed.

B. Port Numbers

This appendix lists the port numbers used by SNMP Agent and shows the direction in which data passes through a firewall.

B.1 Port numbers used by SNMP Agent

The following table lists the port numbers used by SNMP Agent.

Table B–1: Port numbers used by SNMP Agent (Solaris, AIX, and HP-UX(IPF))

Service name	Port	Description
--	161/udp	SNMP request reception
--	8161/udp	Communication between a native agent adapter and a native agent (not used under HP-UX (IPF))
--	7161/tcp	Communication with a subagent

Legend:

--: Not applicable

Table B–2: Port numbers used by SNMP agent (Linux)

Service name	Port	Description
--	22161/udp	SNMP request reception
--	161/udp	Communication between a native agent adapter and a native agent
--	22161/tcp	Communication with a subagent

Legend:

--: Not applicable

B.2 Direction in which data passes through a firewall

The following table shows the direction in which data passes through a firewall.

Table B–3: Direction in which data passes through a firewall (Solaris, AIX, and HP-UX(IPF))

Port number on manager host	Pass-through direction	Port number on the SNMP Agent host
ANY	→	161/udp
ANY	←	161/udp
162/udp	←	ANY

Table B–4: Direction in which data passes through a firewall(Linux)

Port number on manager host	Pass-through direction	Port number on the SNMP Agent host
ANY	→	22161/udp
ANY	←	22161/udp
162/udp	←	ANY

C. List of Kernel Parameters

Adjust the OS kernel parameters to optimally allocate resources needed for executing SNMP Agent. This appendix describes the kernel parameters that need to be adjusted for each OS.

The following table describes the meanings of the symbols used in the estimation expressions in this appendix.

Symbol	Meaning
$\alpha 1$	When <code>/etc/SnmpAgent.d/snmpd.extend</code> is set, this symbol indicates a 5. When not set, this symbol indicates a 0.
$\alpha 2$	Number of extended MIB definition files set under <code>/opt/CM2/ESA/ext/ x 5</code>
$\alpha 3$	When <code>/etc/SnmpAgent.d/snmpd.extend</code> is set, this symbol indicates a 3. When not set, this symbol indicates a 0.
$\alpha 4$	Number of extended MIB definition files set under <code>/opt/CM2/ESA/ext/ x 3</code>
$\alpha 5$	Number of extended MIB definition files set under <code>/opt/CM2/ESA/ext/</code>

C.1 HP-UX (IPF)

System resource	Parameter	Estimate
File system	<code>nfile</code>	$45 + \alpha 1 + \alpha 2$
	<code>maxfiles</code>	Current value + $\alpha 5$
Process	<code>nproc</code>	$8 + \alpha 3 + \alpha 4$

C.2 Solaris

System resource	Parameter	Estimate
File system	<code>rlim_fd_cur</code>	Current value + $\alpha 5$
Process	<code>max_nprocs</code>	$8 + \alpha 3 + \alpha 4$

C.3 AIX

In AIX, you do not need to adjust the kernel parameters.

C.4 Linux

In Linux, you do not need to adjust the kernel parameters.

D. List of Prerequisite Patches and Processes (Services) for SNMP Agent

The patches and processes (services) that are needed for SNMP Agent are listed below. If these patches and processes (services) are not installed, such problems as responses with invalid MIB values or inability to acquire MIB values might occur. To avoid such problems, make sure that you install all patches and processes (services). If a particular operating system is not listed for a patch or process (service), the patch or process (service) is not available for that operating system.

Note:

The prerequisite patches listed below do not necessarily update the corresponding prerequisite processes (services)

OS	Prerequisite patch	Prerequisite process (service)
HP-UX 11i V3 (IPF)	No prerequisite patches are available.	SNMP Agent connects to the <code>mib2agt</code> , <code>ipv6agt</code> , and <code>hp-unixagt</code> processes provided by HP-UX (IPF).
Solaris 11	No prerequisite patches are available.	<code>snmpd</code>
AIX V 7.1,7.2	No prerequisite patches are available.	If the native agent uses the <code>snmpdv1</code> daemon <code>snmpd</code> , <code>dpid2</code> , <code>hostmibd</code> , <code>aixmibd</code> If the native agent uses the <code>snmpdv3</code> daemon <code>snmpd</code> , <code>snmpmibd</code> , <code>hostmibd</code> , <code>aixmibd</code>
Linux	No prerequisite patches are available.	<code>snmpd</code> (For more information about RPM package name, see Release Notes)

E. Version Changes

E.1 Revisions in version 12-00

- SNMP Agent no longer supports the following operating systems:
 - Solaris 10
 - AIX V 6.1
- Supported adding file system target definition file `file/etc/SnmpAgent.d/esafstab.conf`.
- Added files `jpl_esa_log.sh.def` gets.

E.2 Revisions in version 11-00

- SNMP Agent supported the following operating systems:
 - CentOS 6
 - CentOS 7
 - Oracle Linux 7
 - SUSE Linux Enterprise Server 12
- SNMP Agent no longer supports the following operating systems:
 - Red Hat Enterprise Linux 5 (x86, AMD/Intel 64)
 - Red Hat Enterprise Linux 5 Advanced Platform (x86, AMD/Intel 64)
 - Red Hat Enterprise Linux Server 6 (32-bit x86)
 - Oracle Enterprise Linux 5
 - Oracle Linux 6 (32-bit x86)
 - SUSE Linux Enterprise Server 11
- The explanations about the case in RHEL 7, CentOS 7, Oracle Linux 7 and SUSE Linux 12 were added for the notes about installation.
- The default of environment variable `LC_ALL` is changed, which is defined in Operating locale definition file `/etc/SnmpAgent.d/esalocale.conf`.
- The explanation about the case in RHEL 7, CentOS 7, Oracle Linux 7 and SUSE Linux 12 were added for the files executed during system start/shutdown.
- The following files were added to the list of SNMP Agent files:
 - `sub_snmpstart`
 - `sub_snmpstop`
 - `jpl_esa`
 - `jpl_esa.service`
 - `esastart.log`

E.3 Changes from version 10-10 to version 10-50

- The `htc_unixagt4` process was added.
- Explanations were added for the following notes:
 - Setting up the operating locale
 - Notes about CPU information
 - Setting for preventing information responses in the case of file systems that do not require a response (for Linux)
- Notes were added for cases in which the host name is changed after the installation of SNMP Agent.
- Object group `computerSystem64` was added to the Hitachi enterprise-specific MIB objects.
- Implementation of Hitachi enterprise-specific MIB objects (in the `disk64Ex` group) was changed for AIX.
- *6. Definition Files* was added. The following types of definition files are used:
 - Configuration file
 - Environment variable definition file
 - Operating locale definition file
 - File system definition file
 - Disk definition file
- `ifMIB(1.3.6.1.2.1.31)` was added to the configuration file `naa.cnf`.
- The environment variable definition file `SnmpHtcunix4` was added.
- The following files were added to the list of SNMP Agent files:
 - `htc_unixagt4`
 - `SnmpHtcunix4`
 - `freememory64.out`
 - `physmemory64.out`
 - `swapconfig64.out`
 - `disk64.out`
 - `linuxPhysMem64.out`
 - `linuxSwap64.out`

E.4 Changes from version 10-00 to version 10-10

- The `disk64Ex` group was added to the Hitachi enterprise-specific MIB objects.
- The native agent adapter now supports SNMPv2c.
- The `SNMP_HTC_FILE_EXTEND` environment variable can now be used in AIX.
- The following `cpuUtil` group objects are now supported in Linux:
 - `cpuUtilWio`
 - `cpuUtilTotalWio`
- The data collection command can now collect additional types of data.

- Linux support now includes Red Hat Enterprise Linux Server 5 and Red Hat Enterprise Linux Advanced Platform 5.

E.5 Revisions in version 10-00

- SNMP Agent now supports Solaris 11 (SPARC).
- SNMP Agent no longer supports the following operating systems:
 - AIX 5L V5.3 (POWER5, POWER6)
 - HP-UX 11i V2 (IPF)
 - Red Hat Enterprise Linux AS4/ES4 (x86, AMD64&EM64T)
 - Red Hat Enterprise Linux 5 (x86, AMD64&EM64T)
 - Red Hat Enterprise Linux AS4 (IPF)
 - Red Hat Enterprise Linux 5 (IPF)
 - Red Hat Enterprise Linux Advanced Platform 5 (IPF)
 - Solaris 9 (SPARC)
- SNMP Agent now supports the use of IPv6.
- Added the `trapsend` command.
- The configuration file `/etc/srconf/agt/snmpd.cnf` is not overwritten during an upgrade installation.
- Added the `-n` option to the `snmpstart` and `snmpstop` commands for AIX and Solaris.
- For Linux, added the environment variable `SNMP_HTC_LINUX_INACTIVE_MEM` to the `hp_unixagt` process.
- For HP-UX (IPF), added the environment variable `SNMP_HTC_HPUX_ENABLE_PROCESSOR` to the `htc_unixagt2` process.
- For HP-UX (IPF), made the native agent work with the `ipv6agt` process.

E.6 Revisions in version 09-00

- HP-UX is no longer supported.
- The `mib2agt` process is no longer provided.
- The Solaris 10 edition that supports SMF (Service Management Facility) is now supported.
- SNMP Agent is now supported on additional systems.
- A startup shell script (`/usr/CM2/ESA/bin/esa`) has been added for the AIX edition.
- The default location where logs, hexadecimal packet dumps, and VarBind traces are output has been changed. Along with this change, the `SR_LOG_DIR`, `SNMP_HTC_SNMPD_LOG_SIZE`, and `SNMP_HTC_SNMPD_LOG_CNT` environment variables have been added.
- Along with a change in the method of acquiring the size of swap space for the Solaris edition, the `SNMP_HTC_SOLARIS_SWAP_RESERVED` environment variable has been added.
- Along with a change in the method of acquiring CPU utilization rate information in an SMT environment for the AIX edition, the `SNMP_HTC_AIX_CPU_SMT` environment variable has been added.

- Along with a change in the method of acquiring the size of physical memory for the AIX edition, the `SNMP_HTC_AIX_EXCEPT_FILECACHE` environment variable has been added.
- Along with a change in the timing according to which `coldStart` traps are sent, the `SNMP_HTC_INIT_WAIT_TIME` environment variable has been added.
- A method of enabling the `/etc/SnmpAgent.d/esafilesys.conf` setting has been added for the `fileSystem` group (AIX only).
- A method has been added by which the `/etc/srconf/agt/naa.conf` configuration file can be used for setting the community name applied when the native agent adapter sends a GET request or a SET request to the native agent.
- The following groups have been added to the existing set of Hitachi enterprise-specific MIB objects.
 - `fileSystem64` group
 - `diskBusyAvail` group
- The following Hitachi enterprise-specific MIB objects have been added to the `cpuUtil` group:
 - `cpuUtilTotalUser`
 - `cpuUtilTotalSystem`
 - `cpuUtilTotalWio`
 - `cpuUtilTotalIdle`
- The `snmpcmdchk` command has been added.

F. Reference Material for This Manual

This appendix provides reference information, including various conventions, for this manual.

F.1 Related publications

This manual is part of a related set of manuals. The manuals in the set are listed below (with the manual numbers):

- *JP1 Version 12 Network Management: Getting Started* (3021-3-E01(E))
- *JP1 Version 12 JP1/Network Node Manager i Setup Guide* (3021-3-E02(E))
- *JP1 Version 12 JP1/SNMP System Observer* (3021-3-E04(E))
- *Job Management Partner 1/Software Distribution Manager Description and Administrator's Guide* (3000-3-841(E))
- *Job Management Partner 1/Software Distribution Client Description and User's Guide* (3020-3-S85(E)), for UNIX systems

F.2 Conventions: Abbreviations for product names

This manual uses the abbreviations listed below for Hitachi product names and for the names of products from other companies. The following table lists the naming convention used in this manual along with the full name of each product.

Abbreviation		Full name or meaning
AIX		AIX 7.1 (POWER6 and later)
		AIX 7.2 (POWER6 and later)
HP-UX (IPF)		HP-UX 11i V3 (IPF)
JP1/SSO		JP1/SNMP System Observer
		JP1/Cm2/SNMP System Observer
		JP1/Performance Management/SNMP System Observer
Linux	RHEL 6	Red Hat Enterprise Linux(R) Server 6 (64-bit x86_64) ^{#1}
	RHEL 7	Red Hat Enterprise Linux(R) Server 7 ^{#2}
	CentOS 6	CentOS 6 (64-bit x86_64) ^{#1}
	CentOS 7	CentOS 7 ^{#2}
	Oracle Linux 6	Oracle Linux 6 (64-bit x86_64) ^{#1}
	Oracle Linux 7	Oracle Linux 7 ^{#2}
	SUSE Linux 12	SUSE Linux Enterprise Server 12
NNM	HP NNM	HP Network Node Manager Software, version 6 and earlier
		HP Network Node Manager Starter Edition Software, version 7.5 and earlier
	JP1/NNM	JP1/Cm2/Network Node Manager, version 7 and earlier
		JP1/Cm2/Network Node Manager Starter Edition 250, version 8 and earlier

Abbreviation		Full name or meaning
		JP1/Cm2/Network Node Manager Starter Edition Enterprise, version 8 and earlier
NNMi	HP NNMi	HP Network Node Manager i Software
	JP1/NNMi	JP1/Cm2/Network Node Manager i
		JP1/Network Node Manager i
SNMP Agent		JP1/Extensible SNMP Agent
Solaris		Solaris 11 (SPARC)
SubManager		JP1/Cm2/SubManager

#1: The versions of Red Hat Enterprise Linux Server 6, CentOS 6, and Oracle Linux 6 that are supported are 6.1 and later.

#2: The versions of Red Hat Enterprise Linux Server 7, CentOS 7, and Oracle Linux 7 that are supported are 7.1 and later.

HP-UX (IPF), Solaris, AIX, and Linux are often referred to collectively as *UNIX*.

F.3 Conventions: Acronyms

This manual also uses the following acronyms:

Acronym	Full name or meaning
DLPAR	Dynamic Logical Partition
IPF	Itanium(R) Processor Family
MIB	Management Information Base
RFC	Request for Comments
SMF	Service Management Facility
SMT	Simultaneous Multi-Threading
SNMP	Simple Network Management Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol

F.4 Conventions: File naming

The directory and file naming conventions used in this manual are generally those of the product that runs on Solaris and Linux. These differ from the directory and file names of the products that run on HP-UX (IPF) and AIX. In the descriptions, replace the names used in this manual with the name from the following table for the operating system you are using.

HP-UX (IPF)	Solaris and Linux	AIX
/opt/OV/bin/snmptrap	/opt/OV/bin/snmptrap	/usr/OV/bin/snmptrap
/opt/OV/bin/systemtrap	/opt/OV/bin/systemtrap	/usr/OV/bin/systemtrap
/opt/OV/prg_samples/eagent	/opt/OV/prg_samples/eagent	/usr/OV/prg_samples/eagent
/opt/CM2/ESA/newconfig	/opt/OV/newconfig/EAGENT-RUN	/usr/OV/newconfig/EAGENT-RUN

HP-UX (IPF)	Solaris and Linux	AIX
/opt/CM2/ESA/bin/snmpstart	/opt/CM2/ESA/bin/snmpstart	/usr/CM2/ESA/bin/snmpstart
/opt/CM2/ESA/bin/snmpstop	/opt/CM2/ESA/bin/snmpstop	/usr/CM2/ESA/bin/snmpstop
/opt/CM2/ESA/bin/snmpcheck	/opt/CM2/ESA/bin/snmpcheck	/usr/CM2/ESA/bin/snmpcheck
--	/opt/CM2/ESA/bin/snmpcmdchk	/usr/CM2/ESA/bin/snmpcmdchk
/opt/CM2/ESA/ext	/opt/CM2/ESA/ext	/usr/CM2/ESA/ext
/opt/CM2/ESA/bin/trapsend	/opt/CM2/ESA/bin/trapsend	/usr/CM2/ESA/bin/trapsend
/opt/CM2/ESA/newconfig/mgr.cnf	/opt/OV/newconfig/EAGENT-RUN/trapsend	/usr/OV/newconfig/EAGENT-RUN/trapsend
/opt/CM2/ESA/newconfig/SNMPINFO.DAT	/opt/OV/newconfig/EAGENT-RUN/snmpinfo.dat	/usr/OV/newconfig/EAGENT-RUN/snmpinfo.dat
/opt/CM2/ESA/newconfig/MGR.CNF	/opt/OV/newconfig/EAGENT-RUN/mgr.cnf	/usr/OV/newconfig/EAGENT-RUN/mgr.cnf
/opt/CM2/ESA/snmp_mibs	/var/opt/OV/share/snmp_mibs	/usr/OV/conf/snmp_mibs
--	/var/opt/CM2/ESA/log/htc_monagt1.log	/usr/CM2/ESA/log/htc_monagt1.log
--	/var/opt/CM2/ESA/log/htc_monagt1.log.old	/usr/CM2/ESA/log/htc_monagt1.log.old
/opt/CM2/ESA/bin/snmpdm	/usr/sbin/snmpdm (/opt/CM2/ESA/bin/snmpdm in Solaris)	/usr/sbin/snmpdm
/usr/sbin/ipv6agt	--	--
/usr/sbin/mib2agt	--	--
/usr/sbin/hp_unixagt	/usr/sbin/hp_unixagt (/opt/CM2/ESA/bin/hp_unixagt in Solaris)	/usr/sbin/hp_unixagt
/opt/CM2/ESA/bin/trapdestagt	/usr/sbin/trapdestagt (/opt/CM2/ESA/bin/trapdestagt in Solaris)	/usr/sbin/trapdestagt
/opt/CM2/ESA/bin/extsubagt	/usr/sbin/extsubagt	/usr/sbin/extsubagt
/opt/CM2/ESA/bin/htc_unixagt1	/usr/sbin/htc_unixagt1 (/opt/CM2/ESA/bin/htc_unixagt1 in Solaris)	/usr/sbin/htc_unixagt1
/opt/CM2/ESA/bin/htc_unixagt2	--	--
--	/usr/sbin/htc_unixagt3 (/opt/CM2/ESA/bin/htc_unixagt3 in Solaris)	/usr/sbin/htc_unixagt3
--	/usr/sbin/htc_monagt1 (/opt/CM2/ESA/bin/htc_monagt1 in Solaris)	/usr/sbin/htc_monagt1
/opt/CM2/ESA/bin/htc_unixagt4	/usr/sbin/ htc_unixagt4(/opt/CM2/ESA/bin/ htc_unixagt4 in Solaris)	/usr/sbin/htc_unixagt4

Legend:

--: Not applicable

F.5 Conventions: KB, MB, GB, and TB

This manual uses the following conventions:

- 1 KB (kilobyte) is 1,024 bytes.
- 1 MB (megabyte) is 1,024² bytes.
- 1 GB (gigabyte) is 1,024³ bytes.
- 1 TB (terabyte) is 1,024⁴ bytes.

G. Glossary

agent

A network management process that, when running on a managed node, manages network resources as managed objects.

agent system

A system that runs an agent.

authentication failure (authentication error)

A standard SNMP error that occurs on an agent when operations are being performed on a MIB. This error occurs when an unauthorized communication is performed.

community name

A password that is needed to access MIB values on an agent under the SNMP protocol.

enterprise ID

A code that identifies an enterprise (enterprise name).

enterprise-specific trap number

A number that identifies a trap specific to a user enterprise. Enterprise-specific trap numbers must be unique within the group of trap numbers sharing the same enterprise ID.

extended MIB object

A user-specific MIB object. Extended MIB objects are defined by using the extended MIB object definition function provided by SNMP Agent.

Extensible SNMP Agent

A Hitachi-provided agent that runs on a UNIX system.

generic trap number

An SNMP trap number defined in RFC 1157.

local registration file

A file that contains information about the background processes of NNM. Startup configuration information is created from the local registration file.

manager

A network management facility that manages its subordinate nodes.

manager system

A system that runs a manager.

MIB module

A group of multiple MIB objects organized in a tree structure.

MIB object

Management information of a specific type or class.

MIB operation

An operation performed on a MIB object. MIB operations include Get operations, GetNext operations, and Set operations.

MIB value

The value of a MIB object.

netmon (network monitoring process)

See *network monitoring process*.

network monitoring process (netmon)

A background process of NNM that performs polling using SNMP requests and ICMP echo requests to locate nodes on the network.

NNM or NNMi

A program used to manage network configurations, performance, and errors.

ovspmd (process manager)

See *process manager*.

pmd (post master)

See *post master*.

post master (pmd)

A background process of NNM that allocates SNMP traps and SNMP requests.

process manager (ovspmd)

A background process of NNM that monitors the activation, termination, and status of its child processes.

SNMP

Stands for Simple Network Management Protocol. SNMP is a protocol that is used for network management in Internet environments.

standard MIB object

An Internet standard MIB object defined in RFC 1213.

startup configuration information

Configuration information that contains the settings for individual processes provided by NNM. The process manager references this information during the execution of the startup command `ovstart`. Startup configuration information is in the `/usr/OV/conf/ovsuf` file.

Index

Symbols

/etc/SnmpAgent.d/snmpd.conf 44

A

abbreviations for products 348
acronyms 349
Adding file system target definition file (esafstab.conf) 302
AddressTranslation group 131
agent 352
agent system 352
argument 71, 72
authentication failure (authentication error) 352
authentication failure trap, sending 49

B

backing up 121
 configuration file 121

C

cluster environment, setting for operation in 96
cluster group 145
command 214, 215
 details of 216
commands, list of 215
community name 45, 352
 collecting log of unauthorized 317
 of native agent (for Solaris), specifying 47
 registering 45
 setting 45
 specifying 47
 storing in manager 47
community names, types of 45
computerSystem64 group 184, 211
computerSystem group 141, 148
configuration file
 backing up 121
 customizing 44
 format of 50
 naa.cnf 270
 notes about specification of (for Solaris) 60
 restoring 121
 snmpd.cnf 267

 snmpd.conf 264
conventions
 abbreviations for products 348
 acronyms 349
 fonts and symbols 5
 KB, MB, GB, and TB 351
 version numbers 6
CPU information, notes about 103
cpuUtil group 168
customized definition file, backing up 32

D

data
 collecting 313
 to be passed to pipe_in_name 68
 to be passed to pipe_out_name 69
data type 65
defining
 enterprise-specific trap 94
 extended MIB object 61
 MIB object 62, 79
definition file 259
 description format for 263
detailed trace information about master agent, acquiring 315
diskBusyAvail group 183
diskBusy group 166
diskBusyInfo group 181
disk definition file (esadisk.conf) 304
disk group (in AIX) 186
diskInfo64 group 179
diskInfo group 164

E

enterprise ID 19, 352
 set in SNMP trap 19
enterprise-specific trap 18, 20
 defining 94
 using 94
enterprise-specific trap number 352
environment variable
 SNMP_HTC_AIX_CPU_SMT 244
 SNMP_HTC_AIX_EXCEPT_FILECACHE 242
 SNMP_HTC_AUTH_LOG 238

SNMP_HTC_HPUX_ENABLE_PROCESSOR 250
 SNMP_HTC_INIT_WAIT_TIME 238
 SNMP_HTC_LINUX_INACTIVE_MEM 242
 SNMP_HTC_SNMPD_LOG_CNT 238
 SNMP_HTC_SNMPD_LOG_SIZE 238
 SNMP_HTC_SOLARIS_SWAP_RESERVED 242
 SNMP_HTCMONAGT1_START 244
 SR_LOG_DIR 238
 SR_SNMP_TEST_PORT (extsubagt) 240
 SR_SNMP_TEST_PORT (hp_unixagt) 242
 SR_SNMP_TEST_PORT (htc_unixagt1) 247
 SR_SNMP_TEST_PORT (htc_unixagt2) 250
 SR_SNMP_TEST_PORT (htc_unixagt3) 252
 SR_SNMP_TEST_PORT (htc_unixagt4) 254
 SR_SNMP_TEST_PORT (naaagt) 256
 SR_SNMP_TEST_PORT (snmpdm) 238
 SR_SNMP_TEST_PORT (trapdestagt) 258
 SR_TRAP_TEST_PORT 238

environment variable definition file
 provided by SNMP Agent 111

SnmExtAgt 297
 SnmHpunix 282
 SnmHtcmonagt1 295
 SnmHtcunix1 287
 SnmHtcunix2 289
 SnmHtcunix3 291
 SnmHtcunix4 293
 SnmMaster 273
 SnmNaa 278
 SnmNative 280
 SnmTrpDst 285

startup options that can be specified in 111

extended function 19

extended MIB definition file
 adding 77
 adding by stopping SNMP Agent 77
 adding without stopping SNMP Agent 77
 configuring 63
 configuring more than one 76
 storing 77

extended MIB object 20, 352
 copying, to manager 75
 defining 61
 example of defining 79
 non-table format 64
 procedure for defining 62
 reading 77

setting startup options definition file for 78
 table format 66

extensible SNMP Agent 352

extsubagt process 239

F

file
 executed during system shutdown 115
 executed during system startup 113
 to be processed during SNMP request, creating 74

fileSystem64 group 182

file system definition file (esafsys.conf) 300

file system for which response is not required, setting to prevent responses with information about (for Linux) 105

fileSystem group 142

firewall, pass-through direction 341

font conventions 5

free space in physical memory, notes about amount of 100

full-backup, notes about 121

full-restoration, notes about 121

G

GB meaning 351

generic trap 18

generic trap number 352

get community name
 registering 46
 specifying 47

H

Hewlett-Packard enterprise-specific MIB object 140
 cluster group 145
 computerSystem group 141
 description of 140
 fileSystem group 142
 icmp group 147
 ieee8023Mac group 145
 implementation of 148
 organization of 140
 processes group 144
 snmpdConf group 147
 trap group 147

Hewlett-Packard enterprise-specific MIB object, implementation of
 cluster group 151

- computerSystem group 148
- fileSystem group 149
- icmp group 152
- ieee8023Mac group 151
- processes group 149
- snmpdConf group 153
- trap group 152
- Hitachi enterprise-specific MIB object 154
 - cpuUtil group 168
 - description of 156
 - disk64Ex group 184
 - diskBusyAvail group 183
 - diskBusy group 166
 - diskBusyInfo group 181
 - disk group (in AIX) 186
 - diskInfo64 group 179
 - diskInfo group 164
 - fileSystem64 group 182
 - implementation of 188
 - linuxSystem group (for Linux) 187
 - opConf group 156
 - organization of 154
 - page group (in AIX) 186
 - process64 group 175
 - process group 160
 - processor64 group 177
 - processor group 162
 - swapInfo group 164
 - swapSpace group 165
 - swapSystem64 group 180
 - swapSystem group 166
 - system group (in AIX) 186
 - system group (in Solaris) 187
 - systemInfo64 group 171
 - systemInfo group 156
 - virtualMemory64Ex group 173
 - virtualMemory64 group 169
 - virtualMemory group 157
- Hitachi enterprise-specific MIB object, implementation of
 - cpuUtil group 199
 - disk64Ex group 211
 - diskBusyAvail group 210
 - diskBusy group 197
 - diskBusyInfo group 209
 - diskInfo64 group 207
 - diskInfo group 196
 - fileSystem64 group 210
 - group specific to AIX 212
 - group specific to Linux 213
 - group specific to Solaris 213
 - opConf group 189
 - process64 group 204
 - process group 192
 - processor64 group 206
 - processor group 194
 - swapInfo group 196
 - swapSpace group 197
 - swapSystem64 group 208
 - swapSystem group 198
 - systemInfo64 group 201
 - systemInfo group 189
 - virtualMemory64Ex group 202
 - virtualMemory64 group 199
 - virtualMemory group 190
- Hitachi Program Product Installer
 - starting 30
 - using 30
- host, notes about renaming 126
- hp_unixagt process 241
- htc_monagt1 process 243
- htc_unixagt1 process 246
- htc_unixagt2 process 249
- htc_unixagt3 process 251
- htc_unixagt4 process 253

I

- icmpEchoReq 147
- icmp group 147
- ICMP group 134
- identification number 68
- ieee8023Mac group 145
- information collection daemon 234
 - OS supporting 24
 - process 24
- installation 28
 - notes about 37
 - notes about (for AIX) 39
 - notes about (for HP-UX (IPF)) 37
 - notes about (for Linux) 39
 - notes about (for Solaris) 38
 - preparation for 28
 - procedure 27
- installing

- command used to acquire MIB value for OS other than HP-UX (IPF) 28
- SNMP Agent 29
- Interfaces group 130
- invalid shared disk capacity response, setting for suppressing (for AIX and Linux) 97
- IP 48
- IP group 132
- IPv6 54

J

- jp1esalog.sh.def command 217

K

- KB meaning 351
- kernel parameters, list of 342

L

- linuxSystem group (in Linux) 187
- local registration file 352
- log
 - collecting 309
 - notes about 312
 - output destination for 310
- log files, number and size of 311
- logging in to system 71
- log information, method for collecting 322
- logmask
 - for master agent 309
 - for subagent 310
- log output option, setting when performing overwrite installation on version 07-10 or earlier 33
- logs, types of 309

M

- manager 352
- manager commands, verifying objects using 75
- manager system 352
- master agent, process performed at 22
- maximum number of connected subagents 120
- MB meaning 351
- MIB 14
 - integrating in manager 75
- MIB module 352
- MIB object 353
 - defining 62, 79

- managed by SNMP Agent 16
- provided by native agent 17
- MIB operation 353
- MIB tree, structure of 80
- MIB value 353

N

- naa.cnf configuration file, notes about specification of (for Solaris) 60
- naaagt process 255
- native agent 17
 - changing communication protocol with 59
 - configuring (for Solaris and AIX) 59
 - starting 117
 - terminating 117
- native agent adapter 57
 - configuring 59
 - function of 57
 - notes about using 60
 - operation at startup 58
 - operation when SNMP requests are issued from NNM or NNMi 58
 - setting up (for Solaris, AIX, and Linux) 57
 - target native agent of 58
- native agent snmpd, changing SNMP reception port on (for AIX) 119
- netmon (network monitoring process) 353
- network environment setting, notes about 106
- network monitoring process (netmon) 353
- NNM 353
- NNMi 353
- non-table format, extended MIB object in 64
- Notes about operations (for Linux) 126

O

- opConf group 156
- operating local, setting up 43
- operating local definition file (esalocale.conf) 299
- operation
 - notes about 123
 - notes about (for AIX) 125
 - notes about (for Solaris) 125
- option 48
- ovspmd (process manager) 353

P

- page group (in AIX) 186

- physical memory, notes about amount of free space in 100
- pipe_in_name, data to be passed to 68
- pipe_out_name, data to be passed to 69
- pmd (post master) 353
- port number 341
 - used by SNMP Agent 341
- post master (pmd) 353
- PowerHA (HACMP), setting for using 98
- problem
 - corrective action for 318
 - identifying 308
- procedure, from installation to setup 27
- process 214, 234
 - constituting SNMP Agent 21
 - customizing startup option for 110
 - defining environment variable for 110
 - detailed descriptions of 235
 - environment variable that can be specified for 112
 - information collection daemon 24
 - notes about terminating individually 115
 - performed at master agent 22
 - performed at subagent 22
 - used in master agent operation 234
 - used in subagent operation 234
- process64 group 175
- processes, list of 234
- processes group 144
- process group 160
- process manager (ovspmd) 353
- processor64 group 177
- processor group 162
- program, writing 71

R

- restoring 121
 - configuration file 121

S

- script
 - sample 94
 - writing 71
- send/receive packet dump
 - acquiring master agent 313
 - acquiring native agent adapter 314
- set community name

- registering 46
- specifying 48
- setting up
 - all SNMP Agent 75
 - environment for SNMP Agent 26
- setup
 - notes about 106
 - notes about (for AIX) 107
 - notes about (for Linux) 108
 - procedure from installation to 27
- shared disk, required setting for monitoring (for Linux) 96
- shell command to be executed during SNMP request, creating 70
- SNMP 353
- SNMP_HTC_AIX_CPU_SMT environment variable 244
- SNMP_HTC_AIX_EXCEPT_FILECACHE environment variable 242
- SNMP_HTC_AUTH_LOG environment variable 238
- SNMP_HTC_HPUX_ENABLE_PROCESSOR environment variable 250
- SNMP_HTC_INIT_WAIT_TIME environment variable 238
- SNMP_HTC_LINUX_INACTIVE_MEM environment variable 242
- SNMP_HTC_SNMPD_LOG_CNT environment variable 238
- SNMP_HTC_SNMPD_LOG_SIZE environment variable 238
- SNMP_HTC_SOLARIS_SWAP_RESERVED environment variable 242
- SNMP_HTCMONAGT1_START environment variable 244
- SNMP Agent 13, 14
 - changing SNMP reception port on 118
 - command 25
 - function of 16
 - installing 29
 - introduction 13
 - list of patches and processes for (services) 343
 - operating 109
 - operating environment for 15
 - operation using 24
 - problem during operation 319
 - problem during startup 318
 - process 21
 - process configuration (for HP-UX (IPF)) 21
 - process configuration (for Solaris, AIX and Linux) 22

- setting up all 75
- setting up environment for 26
- starting 110
- startup processing 24
- system configuration of 14
- terminating 115
- SNMP Agent files 324
 - list of (AIX) 331
 - list of (HP-UX (IPF)) 324
 - list of (Linux) 336
 - list of (Solaris) 327
- snmpcheck command 222
- snmpcmdchk command 223
- snmpd.cnf 54, 55
- snmpdConf group 147
- snmpdm process 236
- SNMP group 136
- SNMP reception port
 - changing 118
 - changing, on native agent snmpd (for AIX) 119
 - changing, on SNMP Agent 118
- SNMP request processing 16
- snmpstart command 224
- snmpstop command 225
- SNMP trap 16
 - agent address at time of issuing 18
 - issuing 17
 - overview of issuing 17
 - standard trap number of 18
- snmptrap command 226
- SNMP trap transmission port number, setting when performing overwrite installation on version 07-10 or earlier 33
- SR_LOG_DIR environment variable 238
- SR_SNMP_TEST_PORT environment variable
 - extsubagt 240
 - hp_unixagt 242
 - htc_unixagt1 247
 - htc_unixagt2 250
 - htc_unixagt3 252
 - htc_unixagt4 254
 - naaagt 256
 - snmpdm 238
 - trapdestagt 258
- SR_TRAP_TEST_PORT environment variable 238
- standard MIB object 129, 353
 - AddressTranslation group 131
 - description of 129
 - ICMP group 134
 - implementation of 138
 - Interfaces group 130
 - IP group 132
 - organization of 129
 - SNMP group 136
 - System group 130
 - TCP group 135
 - UDP group 136
- starting SNMP Agent 110
- startup configuration information 353
- subagent
 - process performed at 22
 - process provided by SNMP Agent and OS 23
 - reconfiguring 75
- subagents, maximum number of connected 120
- swapInfo group 164
- swapSpace group 165
- swap space size, notes about 102
- swapSystem64 group 180
- symbol conventions 5
- system contact, setting 44
- system group
 - AIX 186
 - Solaris 187
 - System group 130
 - standard MIB object, implementation status of 138
- systemInfo64 group 171
- systemInfo group 156
- system location, setting 44
- systemtrap command 229

T

- table format, extended MIB object in 66
- TB meaning 351
- TCP group 135
- terminating SNMP Agent 115
- trapdestagt process 257
- trap destination 49
 - setting 49
 - setting for NNM 49
 - setting for NNMi and any manager 50
- trap group 147
- trapsend command 230
- troubleshooting 306
 - general procedure for 307

U

- UDP group [136](#)
- uninstallation [41](#)
 - notes about [41](#)
- upgrade installation, performing [32](#)

V

- verifying object using manager command [75](#)
- version number conventions [6](#)
- VIEW [49](#)
- virtualMemory64Ex group [173](#)
- virtualMemory64 group [169](#)
- virtualMemory group [157](#)

 **Hitachi, Ltd.**

6-6, Marunouchi 1-chome, Chiyoda-ku, Tokyo, 100-8280 Japan
