SANnet® II SNMP User's Guide

Review Draft 02

June 2006
83-00004131, Revision A

Copyright

Trademarks

Dot Hill Systems, the Dot Hill logo, SANscape, SANnet, and SANpath are registered trademarks of Dot Hill Systems Corp. All other trademarks and registered trademarks are proprietary to their respective owners.

Changes

The material in this document is for information only and is subject to change without notice. While reasonable efforts have been made in the preparation of this document to assure its accuracy, Dot Hill Systems Corp., assumes no liability resulting from errors or omissions in this document, or from the use of the information contained herein. Dot Hill Systems Corp., reserves the right to make changes in the product design without reservation and without notification to its users.

Contents

SANnet II SNMP User's Guide

Simple Network Management Protocol (SNMP) is a widely used network monitoring and control protocol. It is an application layer protocol that facilitates the exchange of management information between network devices. It is part of the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite.

SNMP enables network administrators to manage network performance, find and solve network problems, and plan for network growth. Data is passed from SNMP agents reporting activity on each network device to the workstation console used to oversee the network. The agents return information contained in a Management Information Base (MIB), which is a data structure that defines what is obtainable from the device and what can be controlled (turned on and off, etc.) A MIB is one of any number of specific characteristics of a managed device.

Managed objects are comprised of one or more object instances, which are essentially variables. Two types of managed objects exist: scalar and tabular. Scalar objects define a single object instance. Tabular objects define multiple related object instances that are grouped in MIB tables.

The SNMP MIB for your array is included with the latest controller firmware patch, version 4.14 or later. Refer to the patch README for the file name and location. See "SNMP Management Using HP OpenView" on page 14 for details on installing the MIB into OpenView. For instructions on installing the MIB into other SNMP management software, refer to the documentation for that product.

An SNMP Object Identifier (OID) is a number assigned to devices in a network for identification purposes. OID numbering is hierarchical. Using the IETF notation of digits and dots resembling very long IP addresses, various registries such as ANSI assign high-level numbers to vendors and organizations. They, in turn, append digits to the number to identify individual devices or software processes.

For details on sending SNMP traps using the RAID firmware, see "Configuring the Notification Processing Center (NPC)" on page 10. For information on using SANscape to send SNMP traps, refer to the "Email and SNMP" appendix in the *SANscape User's Guide.*

Note – SANnet II family arrays can use either SNMPv1 or community based SNMPv2 (v2c) protocols for `SET` and `GET` messages, but they only send SNMPv1 trap messages. Regardless of protocol, the security mechanism is community based and the community name is "public" for traps as well as `SET` and `GET` messages.

SNMP Traps

Polling or requesting information from large numbers of devices that have a large number of objects is not efficient or practical. A better solution is to have each agent of a managed device notify the manager without solicitation. This method is accomplished by the agent sending a message known as a trap of the event to the manager. After receiving the event, the manager displays it and may choose to take an action based on the event. For instance, the manager can poll the agent directly or poll other associated device agents to get a better understanding of the event.

In order for a management system to understand a trap sent to it by an agent, the management system must know what the object identifier (OID) defines. Therefore, it must have the MIB for that trap loaded. This provides the correct OID information so that the network management system can understand the traps sent to it.

Trap-directed notification can result in substantial savings of network and agent resources by eliminating the need for frivolous SNMP requests. However, it is not possible to totally eliminate SNMP polling. SNMP requests are required for discovery and topology changes.

SNMPv1 traps are identified by the values of the following fields:

- `enterprise` - Identifies the type of managed object generating the trap.
- `generic trap` - Indicates one of a number of generic trap types.
- `specific trap` - Indicates one of a number of specific trap codes.

The following are additional fields that may be used to identify an SNMPv1 trap.

- Agent address - Provides the address of the managed object generating the trap.
- Time stamp - Provides the amount of time that has elapsed between the last network reinitialization and generation of the trap.
- Variable bindings - The data field of trap containing PDU. Each variable binding associates a particular MIB object instance with its current value.

The following shows examples for the `enterprise`, `generic`, and `specific` traps:

The `enterprise` has the following value (where the *product-ID* is the Dot Hill SANnet II array product number 3310, 3320, 3510, or 3511):

```
SNMPv2-SMI::enterprises.2294.product-ID
```

The `enterprises` OID is defined by SNMPv2-SMI as follows:

```
.iso.org.dod.internet.private.enterprises
```

The preceding string has the following numeric value:

```
.1.3.6.1.4.1
```

For example, a SANnet II 200 FC array has the following `enterprise` value:

```
SNMPv2-SMI::enterprises.2294.3510
```

This can be described numerically by the following value:

```
.1.3.6.1.4.1.2294.3510
```

The `generic-trap` field has the following value (where `enterpriseSpecific` has the numeric value, 6):

```
enterpriseSpecific
```

The `specific-trap` field has the numeric value, 1.

Note – While all products have a `specific-trap` value of 1, the name for this value varies based on the product-ID specified by the MIB.

Note – The `specific-trap` field always has the value 1, regardless of the event severity.

SNMP Trap Objects

The information carried by the trap is given by a list of variables or object/value pairs. The array sends one object/value pair.

The Object ID (OID) for the variable has the following form (where *product-ID* is the Dot Hill SANnet II array product number 3310, 3320, 3510 or 3511):

```
SNMPv2-SMI::enterprises.2294.product-ID.1.2.3.1.1
```

A SANnet II 200 SCSI array has the following trap OID:

```
SNMPv2-SMI::enterprises.2294.3310.1.2.3.1.1
```

A SANnet II U320 SCSI array has the following trap OID:

```
SNMPv2-SMI::enterprises.2294.3320.1.2.3.1.1
```

A SANnet II 200 SATA array has the following trap OID:

```
SNMPv2-SMI::enterprises.2294.3511.1.2.3.1.1
```

A SANnet II 200 FC array has the following trap OID:

```
SNMPv2-SMI::enterprises.2294.3510.1.2.3.1.1
```

The value for the object is a string that contains the following information:

- Source IP address
- Controller unique identifier
- Event severity
- Time stamp
- Event text string
- Primary|secondary controller

Examples:

```
RAID Event: Host=192.168.0.1, Serial/Unique=5BF, Severity=
Notification, Time=10/31/2005 15:37:19 -- LG:3 Logical Drive
NOTICE: Starting Creation(Primary)
```

```
RAID Event: Host=192.168.0.2, Serial/Unique=4504, Severity=Alert,
Time=10/31/2005 15:34:35 -- Controller BBU Absent or Failed!
!(Secondary)
```

Note – The format and contents of this string are subject to change.

All SNMPv1 messages include the SNMP version and community name. The community name for SANnet II products is “public.”

The SNMP trap listener (`snmptrapd`) is used to show the `enterprise`, `specific-trap`, and the variable OID and value.

Example:

```
# snmptrapd -f -Lo -n
2005-10-31 15:12:58 NET-SNMP version 5.2.1.2 Started.
2005-10-31 15:13:00 192.168.0.1(via UDP: [192.168.0.1]:2766) TRAP,
SNMP v1, community public
       SNMPv2-SMI::enterprises.2294.3510 Enterprise Specific Trap
(1) Uptime: 4:56:30.00
      SNMPv2-SMI::enterprises.2294.3510.1.2.3.1.1 = STRING: "RAID
Event: Host=192.168.0.1, Serial/Unique=4504, Severity=
Notification, Time=10/31/2005 15:55:25 -- Controller BBU is
Charging !(Secondary)"
2005-10-31 15:13:06 NET-SNMP version 5.2.1.2 Stopped.
```

Note – `NET-SNMP` is a popular and freely available software package that includes command line tools for performing SNMP requests such as `GET` and `SET` and a daemon that listens for traps. For more information, see `http://www.net-snmp.com`.

Note – The `agent.ini` file must be configured with trap destinations in order to receive traps. See "To Create and Save the NPC Configuration File (agent.ini)" on page 10.

System Group Objects

Dot Hill products support standard system group objects under the `iso.org.dod.internet.mgmt.mib-2` subtree, defined numerically as follows:

```
1.3.6.1.2.1
```

The system group defines information about the system and includes the objects described in TABLE 0-1.

TABLE 0-1 System Group Objects

Object	Access	Description
`sysDescr`	Read Only	A description of the entity
`sysObjectID`	Read Only	Authoritative identification of the management subsystem
`sysUpTime`	Read Only	Time since the network management subsystem was initialized
`sysContact`	Read-Write	Identification and contact information for the person responsible for this node

TABLE 0-1 System Group Objects *(Continued)*

Object	Access	Description
`sysName`	Read-Write	An administrative name assigned to the node
`sysLocation`	Read-Write	Physical location of the node
`sysServices`	Read Only	A value indicating services offered

The values of these objects can be obtained with an SNMP `GET` message on the OID. A community name of “public” and a protocol version must be specified with the `GET` message. Protocol versions `1` and `2c` are supported. The objects designated with access Read-Write can be set with an SNMP `SET` message. The community name “public” must be specified along with a protocol version for SNMP `SET` messages. All other objects are Read Only.

Note – SNMP must be enabled for `GET` and `SET` messages to work. From the Main Menu, choose “view and edit Configuration parameters →Communication Parameters →Network Protocol Support →SNMP” and verify that SNMP is enabled.

---

sysDescr – The value of this object is a string that describes the product type. For example, using the `NET-SNMP` command, `snmpget`:

```
# snmpget -c public -v 2c 192.168.0.2 sysDescr.0
SNMPv2-MIB::sysDescr.0 = STRING: DotHill SANnet II SCSI
```

In the preceding example, `public` specifies the community name public, and `2c` specifies the protocol version.

sysObjectID – This object is used for administrative purposes to identify the hardware. SNMP managers, such as HP Openview Network Node Manager, can detect this object and value during the discovery process.

Example:

```
# snmpget -c public -v 2c 192.168.0.2 sysObjectID.0
SNMPv2-MIB::sysObjectID.0 =
OID: SNMPv2-SMI::enterprises.2294.3310.1
```

where `public` specifies the community name public, `2c` specifies the protocol version, and `3310` indicates a SANnet II 200 SCSI array.

sysUpTime – This value specifies the time since the last controller reset or power on.

Example:

```
# snmpget -c public -v 2c 192.168.0.2 sysUpTime.0
SNMPv2-MIB::sysUpTime.0 = Timeticks: (1302400) 3:37:04.00
```

sysContact – This object is used to specify a person responsible for the array and their contact information. The object can be set using the `NET-SNMP` `snmpset` command.

Example:

```
# snmpset -c public -v 2c 192.168.0.2 sysContact.0 s "John Doe,
212 555 1212"
SNMPv2-MIB::sysContact.0 = STRING: John Doe, 212 555 1212
```

where `s` indicates the value is of type STRING.

```
# snmpget -c public -v 2c 192.168.0.2 sysContact.0
SNMPv2-MIB::sysContact.0 = STRING: John Doe, 212 555 1212
```

sysName – This object specifies a symbolic name for the array. This value is equivalent to the `controller-name` that can be set and displayed using the CLI. This value can be set with an SNMP `SET` message.

Example:

```
# sccli 192.168.0.2 set controller-name "my array"
sccli: selected se3000://192.168.0.2:58632 [DotHill SANnet II SCSI
SN#0005BF]

# snmpget -c public -v 2c 192.168.0.2 sysName.0
SNMPv2-MIB::sysName.0 = STRING: my array

# snmpset -c public -v 2c 192.168.0.2 sysName.0 s "my other array"
SNMPv2-MIB::sysName.0 = STRING: my other array

# sccli 192.168.0.2 show controller-name
sccli: selected se3000://192.168.0.2:58632 [DotHill SANnet SCSI
SN#0005BF]
 controller-name: "my other array"
```

sysLocation – This object specifies the physical location of the array, such as a floor or room number.

Example:

```
# snmpset -c public -v 2c 192.168.0.2 sysLocation.0 s "1st floor"
SNMPv2-MIB::sysLocation.0 = STRING: 1st floor

# snmpget -c public -v 2c 192.168.0.2 sysLocation.0
SNMPv2-MIB::sysLocation.0 = STRING: 1st floor
```

sysServices – The value of this object specifies layers of the TCP/IP architecture offered. This value is hard-coded at 72.

Example:

```
# snmpget -c public -v 2c 192.168.0.2 sysServices.0
SNMPv2-MIB::sysServices.0 = INTEGER: 72
```

Enterprise Objects

Certain array information can be obtained from SANnet II products using SNMP. These objects are located under the following subtree:

```
.iso.org.dod.internet.private.enterprises.sun.prod.SANnet IIEL
```

or, defined numerically as follows:

```
.1.3.6.1.4.1.2294
```

This node is further branched by product ID as follows:

```
.1.3.6.1.4.1.2294.3310
.1.3.6.1.4.1.2294.3320
.1.3.6.1.4.1.2294.3510
.1.3.6.1.4.1.2294.3511
```

The value for these objects can be obtained using an SNMP `GET` message on the Object ID. A community name of “public” and a protocol version must be specified with the `GET` message. Protocols ‘1’ and ‘2c’ are supported. These objects are defined by the MIB for each product. All objects are read-only.

See the `NET-SNMP` documentation for the location where MIBs are installed. A typical location is `~/.snmp/mibs`.

Example:

```
# cp path-to-mibs/DOTHILL-SANNET-SCSI-MIB.txt ~/.snmp/mibs/
```

Using the `NET-SNMP` `snmpget` command with the MIB, the array information can be retrieved as shown in the following example.

```
# snmpget -c public -v 2c -m +DOTHILL-SANNET-SCSI-MIB 192.168.0.2
ctlrUniqueID.0
DOTHILL-SANNET-SCSI-MIB::ctlrUniqueID.0 = STRING: "5BF"
```

The same example, showing the Object ID in numeric format:

```
# snmpget -On -c public -v 2c -m +DOTHILL-SANNET-SCSI-MIB
192.168.0.2 ctlrUniqueID.0
.1.3.6.1.4.1.2294.3310.1.1.1.1.16.0 = STRING: "5BF"
```

The `-m` option specifies the MIB to use.

The `snmpwalk` command can be used to retrieve all index values of an Object ID in a sequence.

```
# snmpwalk -c public -v 2c -m +DOTHILL-SANNET-SCSI-MIB 192.168.0.2
ldID
DOTHILL-SANNET-SCSI-MIB::ldID.1 = STRING: "4FCF8CED"
DOTHILL-SANNET-SCSI-MIB::ldID.2 = STRING: "513A2612"
DOTHILL-SANNET-SCSI-MIB::ldID.3 = STRING: "38131F15"
DOTHILL-SANNET-SCSI-MIB::ldID.4 = STRING: "1D20B424"
DOTHILL-SANNET-SCSI-MIB::ldID.5 = STRING: "283EA66A"
```

Note – See the product MIB for details on OIDs and descriptions available for enterprise objects.

Using Multiple Product MIBs

Individual product MIBs contain many of the same names for OIDs. For example, the `ctlrUniqueID.0` object in the MIB for the SANnet II 200 FC array has the following OID:

```
.1.3.6.1.4.1.2294.3510.1.1.1.1.16.0
```

The `ctlrUniqueID.0` object in the MIB for the SANnet II 200 SCSI array has the following OID:

```
.1.3.6.1.4.1.2294.3310.1.1.1.1.16.0
```

This can lead to ambiguity if multiple products with multiple MIBs are used. In this case, the fully qualified OID or name must be used.

The fully qualified name can be specified with the MIB module name given by the `DEFINITIONS ::=BEGIN` token in the MIB.

```
# snmpget -c public -v 2c -On
-m +DOTHILL-SANNET-FC-MIB:DOTHILL-SANNET-SCSI-MIB
192.168.0.1 DOTHILL-SANNET-FC-MIB::ctlrUniqueID.0
.1.3.6.1.4.1.2294.3510.1.1.1.1.16.0 = STRING: "4504"
```

The fully qualified name can be specified numerically as follows:

```
# snmpget -c public -v 2c -On
-m +DOTHILL-SANNET-FC-MIB:DOTHILL-SANNET-SCSI-MIB
192.168.0.1 .1.3.6.1.4.1.2294.3510.1.1.1.1.16.0
.1.3.6.1.4.1.2294.3510.1.1.1.1.16.0 = STRING: "4504"
```

Configuring the Notification Processing Center (NPC)

The Notification Processing Center (NPC) is a sub-module on the controller for event notifications. It consists of three components:

- SNMP traps to be received by SNMP-based monitoring software
- Email messages
- Broadcast messages

The array can send SNMP traps about mass storage events to an enterprise management console that uses SNMP, such as HP OpenView or Sun Management Center. It can also send events as email messages or broadcast them to a number of servers.

You specify the SNMP traps, email messages, and broadcast messages you want to use and configure the recipients of these events by creating a text file called `agent.ini` and storing it in reserved space on your array.

Note – The `agent.ini` file must be configured with trap destinations in order to receive traps. See "To Create and Save the NPC Configuration File (agent.ini)" on page 10.

For information about the alternative approach of configuring SANscape to use SNMP, refer to the "Email and SNMP" appendix of the *SANscape User's Guide*.

▼ To Create and Save the NPC Configuration File (`agent.ini`)

1. **Enable FTP on your array by choosing view and edit Configuration parameters → Communication Parameters →Network Protocol Support →FTP - and verifying that it is enabled.**

2. **Create a new text file using a plain text editor.**
3. **Type in the necessary information, specifying which sections you want to enable and the host and email addresses necessary.**

   For a sample file enabling the sending of events as SMTP traps, see "A Simple Sample agent.ini File" on page 11.

   For a sample file enabling the sending of events as SMTP traps, email, and broadcast notifications, see "A Complete Sample agent.ini File" on page 12.

   For a full description of `agent.ini` file parameters, see "agent.ini File Parameters" on page 12.
4. **Save the file as** `agent.ini`**.**
5. **FTP to your array from your workstation.**
6. **Login as root and press Return.**
7. **When prompted for a password, press Return.**

   If you have previously specified a password for the controller, you need to type that password when prompted.
8. **Set the file type to binary (BIN).**
9. `cd` **to the** `/cfg` **directory.**
10. `put` **the file from your workstation into the** `/cfg` **directory.**
11. **Quit the FTP session.**
12. **(Optional) For security reasons, disable FTP on your array by choosing view and edit Configuration parameters →Communication Parameters →Network Protocol Support →FTP - and verifying that it is disabled.**
13. **Reset the controllers by choosing system Functions →Reset controller.**

A Simple Sample `agent.ini` File

A basic sample configuration is shown below
where *nnn.nnn.nnn.nnn* is the IP address of the machine you are referring to,
xxxx@address.com is the sender or recipient's email address,
and ENABLED=0 for OFF, or 1 for ON.

This configuration enables the sending of controller events as SNMP traps to a single host IP address (RECEIVER1) and the sending of controller events in email messages from a specified email address (SENDER_MAIL_BOX) through its SMTP mail server (SMTP_SERVER) to a single email address (also called RECEIVER1, but different from the host computer receiving the SMTP traps). Sending controller events as broadcast messages is not enabled and not specified in the file.

```
[SNMP_TRAP]
ENABLED=1
COMMUNITY=public
RECEIVER1=nnn.nnn.nnn.nnn
[EMAIL]
ENABLED=1
SUBJECT=RAID Event
SENDER_MAIL_BOX=xxxx@address.com
SMTP_SERVER=123.123.123.123
RECEIVER1=xxxx@address.com
```

A Complete Sample `agent.ini` File

This configuration enables the sending of controller events of any severity level as SNMP traps to four host IP addresses and the sending of controller events in email messages to four email addresses. It also enables sending controller events as broadcast messages to four host IP addresses at the same time.

```
[SNMP_TRAP]
ENABLED=1
SEVERITY=1
RECEIVER1=192.168.0.1
RECEIVER2=192.168.0.2
RECEIVER3=192.168.0.3
RECEIVER4=192.168.0.4
[EMAIL]
ENABLED=1
SEVERITY=1
SUBJECT=Event from SANnet RAID Controller
SENDER_MAIL_BOX=user@example.com
SMTP_SERVER=192.168.0.1
RECEIVER1=user1@example.com
RECEIVER2=user2@example.com
RECEIVER3=user3@example.com
RECEIVER4=user4@example.com
[BROADCAST]
ENABLED=1
SEVERITY=1
RECEIVER1=192.168.0.1
RECEIVER2=192.168.0.2
RECEIVER3=192.168.0.3
RECEIVER4=192.168.0.4
```

`agent.ini` File Parameters

The parameters you can specify in the agent.ini file are defined below:

The configuration file is comprised of three major sections: SNMP, Email and Broadcast. Each notifying method can be separately enabled or disabled.

The SNMP_TRAP Section

`[SNMP_TRAP]` – section header

`[ENABLED]` – 1=enabled, 0=disabled (applies to this section only)

`[SEVERITY]` - level of severity of the messages to be received. (1 covers events of all levels. 3 sends only the most serious events.)

- 1. notification
- 2. warning
- 3. alert

`[COMMUNITY]` – SNMP community name of the destination/ receiver

`[RECEIVER#]` – The IP address of the receiver computer. Add additional lines to specify multiple receivers. Up to 4 receivers can be configured.

The EMAIL Section

`[EMAIL]` – section header

`[ENABLED]` – 1=enabled, 0=disabled (applies to this section only)

`[SEVERITY]` - level of severity of the messages to be received: notification, 2. warning, 3. alert. “1” covers events of all levels. “3” sends only the most serious events.)

`[SUBJECT]` – add a topic to email. This can be used to specify the location of the RAID system, if there are many.

`[SENDER_MAIL_BOX]` – a valid email address to be used as the “from” part of the email message.

`[SMTP_SERVER]` – SMTP server used to send email. IP address only, do not enter a host name here.

`[RECEIVER#]` – receiver’s email address. The receiver’s number followed by an “=” mark, an email address, comma, and the number to specify the message severity level.

The BROADCAST Section

`[BROADCAST]` – section header

`[ENABLED]` – 1=enabled, 0=disabled (applies to this section only)

`[SEVERITY]` – level of severity of the messages to be received: 1. notification, 2. warning, 3. alert. “1” covers events of all levels. “3” only the most serious events will be broadcast.)

`[RECEIVER#]` – The IP address of the receiver computer. Add additional lines to specify multiple receivers. Up to four receivers can be configured.

SNMP Management Using HP OpenView

HP OpenView is a complete enterprise network management, maintenance, and monitoring solution. Included with the HP OpenView software are tools for sending SNMP `GET` and `SET` messages and tools for listening for SNMP traps.

This section discusses using HP OpenView for SNMP management with SANnet II family arrays. Topics covered include:

- "Loading MIBs" on page 14
- "Configuring Events" on page 15
- "Viewing and Setting System Group Objects" on page 16
- "Viewing Enterprise Objects" on page 18

Loading MIBs

By itself, OpenView can listen for and dispatch SNMP traps. However, MIBs are supplied to make the best use of the management feature.

Note – The `agent.ini` file must be configured with trap destinations in order to receive traps. See "To Create and Save the NPC Configuration File (agent.ini)" on page 10.

It is assumed that HP OpenView has discovered the SANnet II family nodes. Refer to your HP OpenView documentation for details on node discovery.

On Solaris, type the following command to launch OpenView.

```
# /opt/OV/bin/ovw &
```

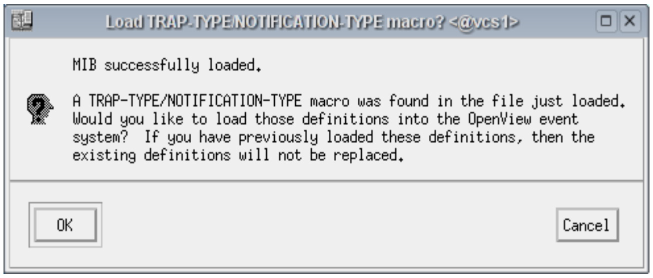
From the Root dialog, navigate to the network segment on which the SANnet II nodes reside.

To load MIBs, perform the following steps.

1. **Choose Options →Load/Unload MIBs: SNMP.**
2. **Click Load...**

3. **Select the MIBs to be loaded and click OK.**

   Since the MIB contains TRAP/NOTIFICATION information, OpenView will detect this. A dialog is displayed requesting confirmation to load the definitions into the OpenView event system.

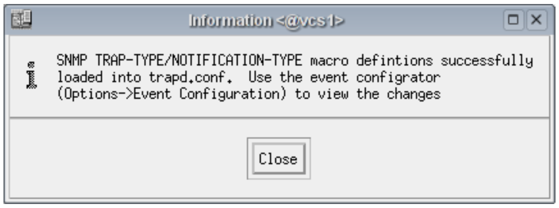


MIB successfully loaded.

A TRAP-TYPE/NOTIFICATION-TYPE macro was found in the file just loaded. Would you like to load those definitions into the OpenView event system? If you have previously loaded these definitions, then the existing definitions will not be replaced.

4. **Click OK to continue.**

   If the definitions load successfully, an information dialog is displayed.



SNMP TRAP-TYPE/NOTIFICATION-TYPE macro defintions successfully loaded into trapd.conf. Use the event configrator (Options->Event Configuration) to view the changes

5. **Click Close.**

   If more than one array product type is being monitored (for example, a SANnet II 200 SCSI array and a SANnet II 200 FC array), you can load multiple MIBs.

6. **Repeat the preceding steps for each MIB to be loaded.**

Note – You can also unload MIBs by selecting them from the list and clicking "Unload."

Note – MIBs are distributed with firmware patches. You might have to reload MIBs if firmware changes impact the SNMP agent on the array controller.

Configuring Events

Since the MIBs contain information on traps, it is possible to configure these events. Events can be displayed in the Alarm browser, in a pop-up window, forwarded to other hosts, and logged to files. Refer to your HP OpenView documentation for details.

If the MIBs have been loaded successfully, the Enterprise Name and Enterprise ID will be displayed in the Event Configuration dialog.

To view and modify events, perform the following steps.

1. **Choose Options →Event Configuration.**

   The Event Configuration dialog is displayed.

2. **To modify an event, select the Enterprise Name in the upper pane and double-click on the Event Name in the lower pane.**

   The Modify Event dialog is displayed.

3. **To display the event in the Alarm browser, select an appropriate category.**

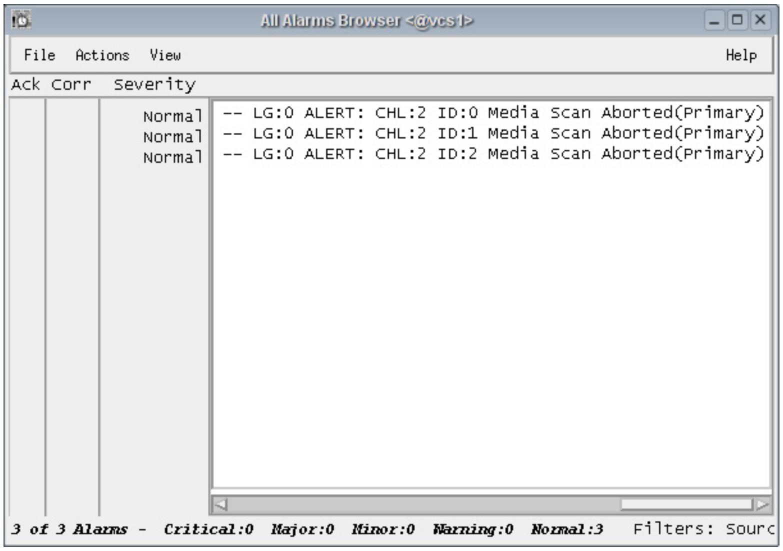
   Since only one trap type is supported, all events will go into this category. In this example, a category of “Status Alarms” is used.

4. **Enter a message for the event in the Event Log Message text box.**

   In this example, “$1” represents the variable string of the SNMP trap. See the HP OpenView documentation for details on other “$” variables available as part of the event message.

5. **To view alarms, from the main menu bar, choose Fault →Alarms.**

   The following dialog is displayed.



```
Normal  -- LG:0 ALERT: CHL:2 ID:0 Media Scan Aborted(Primary)
Normal  -- LG:0 ALERT: CHL:2 ID:1 Media Scan Aborted(Primary)
Normal  -- LG:0 ALERT: CHL:2 ID:2 Media Scan Aborted(Primary)

3 of 3 Alarms -  Critical:0  Major:0  Minor:0  Warning:0  Normal:3   Filters: Sourc
```

Viewing and Setting System Group Objects

SNMP must be enabled on the array to view and set system group objects. From the RAID firmware Main Menu, choose “view and edit Configuration parameters → Communication Parameters →Network Protocol Support →SNMP” and verify that SNMP is enabled.

1. **From HP OpenView, browse the system group objects for a node by selecting the node on the segment map.**

2. **Choose Tools →SNMP MIB Browser.**

   Confirm that the correct Name or IP Address is displayed.

3. **Navigate to the following MIB Object ID:**

   `iso.org.dod.internet.mgmt.mib-2`

4. **Select `system` from the list and click Start Query.**

   Read/Write values can be set from this dialog.

5. **To set the system name, select `sysName.0` from the list and enter a new value in "SNMP Set Value".**

6. **Click Set.**

   A new query on the system group shows the new value.

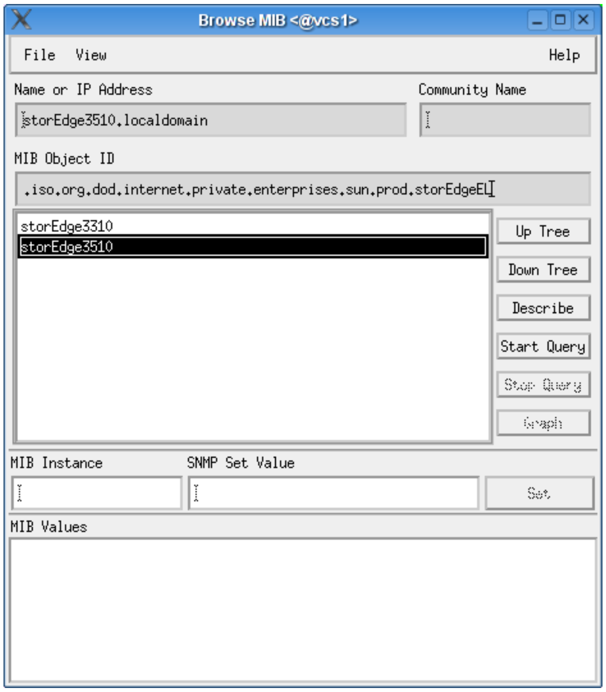
Viewing Enterprise Objects

SNMP must be enabled on the array to view enterprise objects. From the RAID firmware Main Menu, choose "view and edit Configuration parameters → Communication Parameters →Network Protocol Support →SNMP" and verify that SNMP is enabled.

Certain array information can be obtained by browsing the MIB. These objects are located under the following subtree:

```
.iso.org.dod.internet.private.enterprises.sun.product.storEdgeEL
```

1. Navigate to this object ID with the MIB browser and select the appropriate branch for the product.

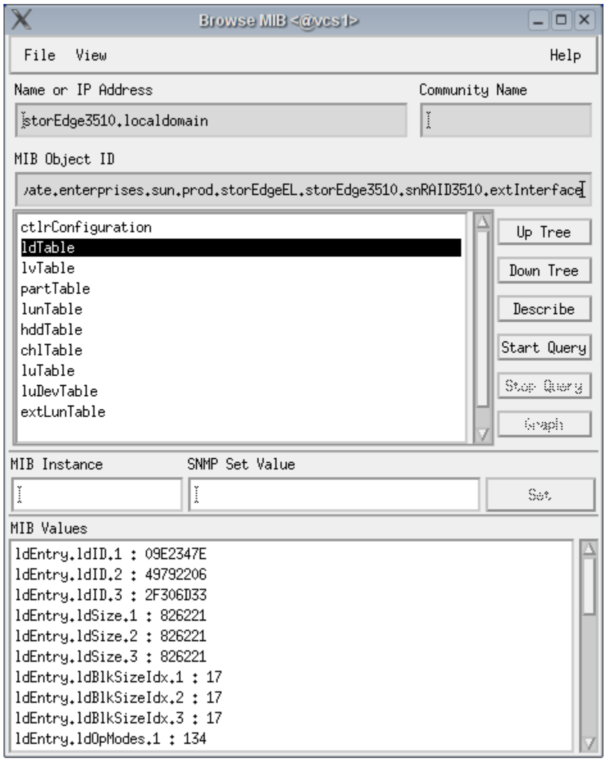


Name or IP Address: storEdge3510.localdomain
Community Name:
MIB Object ID: .iso.org.dod.internet.private.enterprises.sun.prod.storEdgeEL

storEdge3310
storEdge3510

2. **Continue to browse to the** `extInterface` **node and select the desired item from the list.**

3. Click Start Query.



Name or IP Address: storEdge3510.localdomain

Community Name:

MIB Object ID: /ate.enterprises.sun.prod.storEdgeEL.storEdge3510.snRAID3510.extInterface

```
ctlrConfiguration
ldTable
lvTable
partTable
lunTable
hddTable
chlTable
luTable
luDevTable
extLunTable
```

MIB Instance:

SNMP Set Value:

MIB Values

```
ldEntry.ldID.1 : 09E2347E
ldEntry.ldID.2 : 49792206
ldEntry.ldID.3 : 2F306D33
ldEntry.ldSize.1 : 826221
ldEntry.ldSize.2 : 826221
ldEntry.ldSize.3 : 826221
ldEntry.ldBlkSizeIdx.1 : 17
ldEntry.ldBlkSizeIdx.2 : 17
ldEntry.ldBlkSizeIdx.3 : 17
ldEntry.ldOpModes.1 : 134
```

For details on OIDs and descriptions available for enterprise objects, see the MIB for your array.