StorageTek™ ACSLS

SNMP Agent

INSTALLATION AND USER'S GUIDE FOR SOLARIS

312555203
Version: 2.0.5

Automated Cartridge System Library Software

SNMP Agent

Installation and User's Guide for Solaris

Version 2.0.5

Sun Microsystems, Inc.
www.sun.com

Part No. 312555203
June, 2007

We welcome your feedback. Please contact the Sun Learning Services Feedback System at:

Sun Learning Services
Sun Microsystems, Inc.
One StorageTek Drive
Louisville, CO 80028-3256
USA

Please include the publication name, part number, and edition number in your correspondence if they are available. This will expedite our response.

Please
Recycle

Rev.C • 312555203

Revision History

EC	Date	Revision	Description
132575	June, 2007	C	This document supports the ACSLS SNMP Agent for the Solaris platform, Version 2.0.5

Contents

CHAPTER 1

Overview

SNMP (Simple Network Management Protocol) is an industry-accepted model for collecting operating status from a wide variety of information technology hardware and software nodes within a data center. Each node is equipped with agent server software that communicates to a client. The client is typically a management application that listens for incoming traps and provides comprehensive status displays on a service console. This SNMP client retrieves status information from scores of server agents across the data center.

The role of each Agent is to expose status information to the client about the set of objects that it manages. All of the managed objects are represented in a Management Information Base (MIB). The client management application is in touch with multiple Agents reporting their respective MIBs. The management application can report on the status of each object in the entire data center. It can react to problems or status changes by sending an E-mail message to an administrator or by paging an appropriate support technician.

The ACSLS SNMP Agent 2.0.5 is responsible for objects defined within the ACS-TAPE-MONITOR-MIB. It maintains status information about storage libraries under ACSLS control and it exposes the ACS-TAPE-MONITOR-MIB database to the management application, communicating any status changes of the various objects to the management console. The ACSLS Agent works behind a Solaris SNMP Master Agent whose SNMP domain reaches to the various subsystems running on the Solaris system.

The Agent is intended to run in a Solaris-10 environment (SPARC and X86) on a host that is running ACSLS 7.1 or later software. The Agent provides ACSLS-queried information about the monitored ACSs and their internal components, such as LSMs, CAPs, and drives. The Agent regularly queries the libraries through the ACSLS server and sends asynchronous messages (SNMP traps) to registered clients whenever changes are detected in the status of a library or any of its components. Standard SNMP Agents listen for requests on UDP port 161 and send traps through UDP port 162. The port assignments are adjustable for administrators who require unique and secure network configurations.

More information on SNMP can be found at *http://www.simple-times.org/index.html.*

This manual provides installation, configuration, and operation instructions for the ACSLS SNMP Agent. The document offers hints on how to use the Agent from a SNMP management application. This guide provides a troubleshooting chapter to offer guidance for restoring operation of the Agent in situations arising from common problems.

CHAPTER 2

Installation

This chapter describes the installation procedure for the ACSLS SNMP Agent on Solaris 10. ACSNMP version 2.0.5 is available for both SPARC and X86 Solaris platforms running ACSLS version 7.1 or later.

Version 2.0.5 of the SNMP Agent is available for download from the Sun StorageTek Customer Resource Center. Navigate to the ACSLS Software Web directory and look for the following packages:

- ACSNMP for SPARC Solaris 10 STKacsnmp.2.0.5.S.tar.gz
- ACSNMP for X86 Solaris 10 STKacsnmp.2.0.5.X.tar.gz

Installing the Agent

▼ Installing the agent

1. Download the package and transfer it to the **/opt** directory on your ACSLS server.
2. Login as **root**.
3. Extract the package:

   # cd /opt

   # gunzip STKacsnmp.2.0.5.S.tar.gz

   # tar -xvf STKacsnmp.2.0.5.S.tar
4. Install the package:

   # pkgadd -d .

   This operation displays a list of packages available in the current directory.
5. From the menu, select **STKacsnmp**.

   A prompt displays asking if you want to select the default installation directory, **/export/home**.
6. Select the default installation directory.

   The default installation directory is recommended for ease of support.

If you follow the default installation procedure, files associated with the Agent are in the directory, **/export/home/ACSNMP**.

▼ Verifying the Installed Package

1. Verify the installed package:

   pkginfo -r STLacsnmp.

Note – Further references to **$ACSNMP_HOME** equates to the directory path:
‘pkginfo -r STKacsnmp`/ACSNMP

In most cases, this translates to **/export/home/ACSNMP**.

The installed package contains the following:

- Agent software,
- Several utilities that assist you as you configure and test the Agent.
- The Agent MIB in located in the file:

  AcslsMib.mib.

  This file is used by your system management application.
- Numerous ACSLS SNMP Agent files are installed in system directories for purposes of registering the ACSLS Agent with the Solaris Master Agent, and for starting up the Agent automatically.

Starting and Stopping the Agent

The ACSLS SNMP Agent starts automatically by the Solaris system *init* process during a system boot.

On later updates (3 or later) of Solaris-10, the Service Management Facility (SMF) manages startup and shutdown of system SNMP services. The ACSLS Agent is intended to run as a background process as long as the master agents are up and ACSLS is running. When ACSLS stops, the Agent also stops. The Agent restarts automatically when ACSLS restarts.

The utility **acsnmp_ctl** is provided for all startup and shutdown operations for the ACSLS agent. This utility is located in the **$ACSNMP_HOME** directory. There are four control options provided with **acsnmp_ctl***:*

- **acsnmp_ctl start**

  This command starts the ACSLS agent if the ACSLS agent has been registered and Solstice Enterprise Agent and the System Management Agent are currently running.
- **acsnmp_ctl stop**

  This command stops the ACSLS agent.
- **acsnmp_ctl restart**

  This command acts on the Acsls Agent without affecting the master agents. The *restart* command will stop the AcslsAgt and then restart it

- **acsnmp_ctl register**

  This command stops and restarts the Solstice Enterprise Agent and the System Management Agent. It then starts the ACSLS Agent. The **register** command is used when you bring up the ACSLS Agent for the first time or whenever you have made a configuration change that must be registered with the master agents.

- **acsnmp_ctl status**

  This command displays the status of the Solstice Enterprise Agent, the System Management Agent and the ACSLS Agent. If all of these are running, the utility displays the current configuration of hosts, communities, and trap destinations. It also sends a test communication packet to the local host.

▼ To start the ACSLS Agent for the first time without a system reboot:

1. Log in as root.
2. Register the new agent with the Solstice Enterprise Agent and the System Master Agent by running:

   acsnmp_ctl register.

Uninstalling the ACSNMP software package

▼ To uninstall the ACSLS SNMP Agent:

1. Stop the ACSLS SNMP Agent.

   $ACSNMP_HOME/acsnmp_ctl stop

2. Remove the package.

   pkgrm STKacsnmp

CHAPTER 3

Configuring the ACSLS Agent

This chapter describes the use of tools provided for use in configuring ACSLS SNMP communication and access control. All of these utilities require **root** user access. These utilities are found in the ACSNMP directory which is typically installed under **/export/home/**. You can verify the actual directory path using **pkginfo -r STKacsnmp**.

SNMP configuration is largely a matter of aligning community names and host names for access to a specific MIB. When configuring SNMP for access to the ACSLS MIB there are several questions to ask:

- Who (what community) will be submitting *get* requests?
- Who (what community) will be submitting *set* requests?
- From what machines will *set* or *get* requests be submitted?
- Who (what community) will be listening for trap messages?
- To what machines must the trap messages be sent?

A community name is like a user name. This name is embedded within each SNMP request packet. Since a common user id can be used across the network, it is called a community. Different user communities may have different levels of access. Some communities may be able to read or get information. Others may be able to write or set parameters in the MIB. Some may be able to both read and write. The conventional SNMP community names are public and private. Typically, the *public* community is given read access, and the *private* community can both read and write. If security is a concern, an administrator can use unconventional names other than *public* and *private*.

General Configuration

Declaring SNMP User Communities

The ACSLS Agent cannot connect to the SNMP Master Agent unless configured communities and their privileges match from one layer to the next. Where there is a mismatch between the ACSLS Agent and the Master Agent, the ACSLS Agent will behave as if the Master Agent is not running. The SNMP ports, the community, and the permissions of that community must be consistent.

▼ To Declare the Agent's Community

1. Issue the following command to go to the directory where the Agent is installed:

   cd /*InstallationDirectory***/ACSNMP**

   The default installation directory is **/export/home/ACSNMP**.

2. Declare the various communities in the following access control files.

   /etc/snmp/conf/snmpdx.acl (Master agent access)
   /etc/snmp/conf/AcslsAgtd.acl (ACSLS agent access)

   In most cases, it is not necessary to manipulate these files directly since there are two utilities in the ACSNMP directory that can be used to declare the communities and trap destinations.

   - By default, the ACSLS agent community is declared as *public*, but you can declare a different community name using the following expression:

     AcslsAgtdSnmpConf -c <*community name*>

   - If more than one community is to be declared, then enclose the community string in quotes, separating the community names with a comma:

     AcslsAgtdSnmpConf -c <public, private>

3. If you choose to use a community name other than public or private, then be sure to declare that community in the following configuration files:

   /etc/snmp/conf/snmpd.conf
   /etc/snmp/conf/snmpdx.acl
   /etc/sma/snmp/snmpd.conf

   Having made the change, you need to register the change using the following utility in the ACSNMP home directory:

   acsnmp_ctl register

4. Verify the changes you made in the ACSLS Agent:

   AcslsAgtdSnmpConf -d

   Generally, these communities have SNMP access from any remote machine. But if you wish to limit ACSLS SNMP access to one or more specific machines, it is necessary to edit the **/etc/snmp/conf/AcslsAgtd.acl** file, listing the host names of each qualified machine in the *managers* field. By default, an asterisk (*) is inserted in the *managers* field to enable access to all hosts.

   ```
   acl = {
         {
            communities = public, private
            access = read-write
            managers = *
         }
   }
   ```

   By removing the asterisk and inserting one or more host names, you can limit access to those specific hosts.

Port Configuration

Normally you will use the default ports,**161** for general (*set/get*) operations and **162** for traps. However, if an administrator has intentionally changed the default settings, the change will be seen in the startup script that launches 'snmpdx'. You will need to match that port number you specify with the ports defined for the master agent. To do this, use **AcslsAgtdSnmpConf**.

If a non-default port has been configured for the master agent, you can look for the port number in the string that launches the master agent.

1. Run **svcs snmpdx** and look for the response.
   - If the response is a logical state such as "online" then snmpdx is under SMF control.

     In this case, the file in question is 'svc-snmpdx'. To locate the specific string, use the command:

     grep SNMP_BIN /lib/svc/method/svc-snmpdx
   - If the response to **svcs snmpdx** indicates that snmpdx "does not match any instances", then the file is a legacy file that resides in the **/etc/rc3.d** directory.

     To locate the specific string, use the command

     grep "snmpdx -f" /etc/rc3.d/S76snmpdx
2. The port number in question follows the *-p* parameter in the string that launches snmpdx. If the *-p* parameter does not exist in the string, then you can be assured that the default ports 161 and 162 are used for snmp communications.
3. Once you have changed the port configuration, you should restart the ACSLS Agent.

Trap Configuration

To declare which host machines are to receive trap messages from the agent, use the trap destination utility in the ACSNMP directory. You can *add* a hostname using the following:

AcslsAgtdTrapDest -a <hostname>

Any hostname you declare should be registered in NIS or in your local /etc/hosts file. This utility allows you to add only one hostname at a time, but you can repeat the command as many times to include as many hosts that you require. To get a *list* of the trap destination hosts that have been configured:

AcslsAgtdTrapDest -l

If you wish to *remove* a specific trap destination host, use the same utility with the -r parameter:

AcslsAgtdTrapDest -r <hostname>

Configuring for *get* and *set* Permissions

SNMP files are configured by default to provide read-only access to the communities specified in step-1 above. The ACSLS MIB provides for remote access to three settable parameters (see below: Setting Properties of the Agent). To manipulate these parameters remotely, you will need to change the configuration to allow read-write access for your community.

The ACSLS SNMP agent on Solaris-10 communicates through the System Management Agent. To open read-write access to specific communities through this agent, it will be necessary to edit the System Management Agent configuration file:

/etc/sma/snmp/snmpd.conf

This file is used to configure general permissions for communities accessing the System Management Agent. Communities who will be submitting only *get* requests should be declared as a *rocommunity* for read-only access. For example:

rocommunity public

Communities that will be submitting *set* requests must be declared as a *rwcommunity* for read-write access. For example:

rwcommunity private

A community that is declared as an *rwcommunity* will be allowed to submit to both *get* and *set* SNMP requests.

If you wish to limit *set* capabilities to a specific community on a specific machine, then you must declare the machine host name on the same line:

rwcommunity <community name> <hostname>

You can also limit the ability to set parameters to a specific parameter in the MIB. To do so, simply list the MIB OID on the same line:

rwcommunity <community name> <hostname> <OID>

For example:

rwcommunity private daffy 1.3.6.1.4.1.1211.1.11.2.4.0

In this example, the system will allow the *private* community on the machine *daffy* to change only the report level of SNMP traps.

Restarting the SNMP Agent

Any time port changes or community changes are made to the configuration files, the changes must be registered with the Solstice Enterprise Agent and the System Master Agent. Use the command *acsnmp_ctl register* which resides in the **$ACSNMP_HOME** directory

acsnmp_ctl register

Testing your changes

Sun has provided a *ping-like* tool in the ACSNMP directory that submits a simple *get* request for each community that has been declared.

▼ To run the utility:

1. Go to the $ACSNMP_HOME directory.
2. Login as root and run the following command:

   pingAcslsAgt

   This utility:

   - Checks to see that the SNMP master agent and the System Management Agent are running.
   - Reveal the ACSLS permission levels for each configured community.
   - Displays the specific hosts that can communicate with the ACSLS SNMP Agent.
   - For each community, this utility verifies access by requesting the version number of the agent.
   - The utility then verifies whether the agent has established communication with ACSLS.
   - Finally, the utility lists all of the trap destination hosts that have been configured and verifies whether each is reachable.
   - If there are any problems in your configuration, this tool can lend possible assistance in identifying the source of the problem.

   Another tool in the SNMP directory reveals a complete list of all ACSLS OID's, listing each numeric ID, its human-readable translation, and the value that was returned for that object. When ACSLS software is running, this tool reveals all of the objects and their OIDs throughout the entire ACSLS MIB.

   To run this tool:

   translate

Setting Properties of the Agent

There are three settable parameters within the ACSLS MIB that determine certain operating properties of the agent. These include:

acsAgtUrl	**1.3.6.1.4.1.1211.1.11.1.4.0**
acsTrpCurPollingRate	**1.3.6.1.4.1.1211.1.11.2.2.0**
acsTrpLogReportLevel	**1.3.6.1.4.1.1211.1.11.2.4.0**

- **acsAgtUrl**

  The **acsAgtUrl** is used by the Sun StorageTek FrameWork Monitor to identify the ACSLS agent. This parameter may have little or no use by other applications and its value in most cases will be blank. You can set this value using either of two methods. You can edit the file AcslsAgt.url in the ACSNMP directory, assigning the url to AGENT_URL_ENTRY. With this value set, the URL will be established in the MIB when the ACSLS agent starts up.

  Alternatively, if the agent is currently running, you can set the URL into the MIB using the SNMP 'set' command:

  /usr/sfw/bin/snmpset -v1 -c private localhost 1.3.6.1.4.1.1211.1.11.1.4.0 s <url>

  The `-v1` parameter specifies version-1 SNMP protocol. The ACSLS Agent uses only version-1 SNMP packets and it is necessary to include this parameter. The "s" between the OID and the url_expression is a data type (STRING) declaration and is required when setting a string value. The expression following "-c" is the community identifier. If the command fails, you should use 'pingAcslsAgt' to double-check the R/W permissions for the community name that you specify here.

  To verify the URL change, submit a 'get' request for that OID:

  /usr/sfw/bin/**snmpget** -v1 -c public localhost 1.3.6.1.4.1.1211.1.11.1.4.0

- **acsTrpCurPollingRate**

  The **acsTrpCurPollingRate** is the period of time between SNMP probes from the agent to the ACSLS server. There is a happy medium to which an administrator would aim when setting this value. By polling ACSLS frequently, the agent is assured of fresh, up-to-date information. By polling too frequently, the agent has the potential to impact library performance, interfering with activity between ACSLS and its client applications. A reasonable polling rate is typically between 15 and 60 seconds.

  You can set the current polling rate by two different methods.

  - You can edit the AcslsAgtd.cfg file in the ACSNMP directory, adjusting the "CURR_RATE" parameter.

    Once this file is changed, you will need to stop and start the ACSLS agent using

    acsnmp_ctl register.

  - Alternatively, you can dynamically change the current polling rate using an SNMP set command:

    /usr/sfw/bin/**snmpset** -v1 -c private localhost 1.3.6.1.4.1.1211.1.11.2.2.0 I <number_of_seconds>

  As described above, the "-v1" parameter defines the packet level and "-c" identifies the community. The "i" between the OID and the actual number of seconds is a type (INTEGER) declaration and is required when setting an integer value. The actual number you specify must fall in the range between fifteen seconds and sixty seconds. If you specify a number outside of this range, the set request will be ignored by the agent.

If the command fails, you can use 'pingAcslsAgt' to double-check the R/W permissions for the community name that you specify here. To verify the change, use a 'get' request:

/usr/sfw/bin/**snmpget** -v1 -c public localhost 1.3.6.1.4.1.1211.1.11.2.2.0

If there is a compelling reason to set this value lower than 15 seconds, you must first change the minimum polling rate to a value less than 15 seconds. The only way to adjust acsTrpMinPollingRate is to edit the AcslsAgtd.cfg file in the ACSNMP directory, adjusting the "MIN_RATE" parameter. Once this file is changed, you will need to stop and start the ACSLS agent using **acsnmp_ctl restart**.

- **acsTrpLogReportLevel**

  The **acsTrpLogReportLevel** defines the type and level of verbosity for trap messages. There are five possible levels to be defined:

1	silent	No trap messages will be sent.
2	error	Only error messages will be sent.
3	warning	Error messages and status changes to 'offline' will be sent.
4	info	Error messages and all status changes will be sent.
5	unclassified	Errors, status changes, and informational messages will be sent.

  The default setting is “5” (unclassified). To change the value of this parameter, use the SNMP 'set' command:

  /usr/sfw/bin/**snmpset** -v1 -c private localhost 1.3.6.1.4.1.1211.1.11.2.4.0 i <level>

  As described above, the "-v1" parameter defines the packet level and "-c" identifies the community. The "i" between the OID and the actual number of seconds is a type (INTEGER) declaration and is required when setting an integer value. The actual number you specify must fall between one and five. If you specify a number outside this range, the set request will be ignored by the agent.

  If the command fails, you can use 'pingAcslsAgt' to double-check the R/W permissions for the community name that you specify here. To verify the change, use a 'get' request:

  /usr/sfw/bin/**snmpget** -v1 -c public localhost 1.3.6.1.4.1.1211.1.11.2.4.0

Setting the Log Trace Level

The Agent logs internal events, such as entering a function or returning a null pointer, are in `AcslsAgtd.log` log files located in the Agent home installation directory.

The Agent generates the first log file called `AcslsAgtd.log`. When the log file reaches 300 KB, it is rolled over to a backup file called AcslsAgtd.log.0. The AcslsAgtd.log file is then flushed to leave room for a new 300 KB worth of information. The size of the two log files put together never exceeds 60 KB.

Four trace levels are available:

- SILENT produces no trace information
- ERROR traces errors only
- WARNING provides both error and warning information
- DEBUG traces errors, warnings and al the Agent's operations (all messages are recorded)

Changing the Log Trace Level

The trace level is set with the environment variable ACS_TRACE that is set using the following values:

DEBUG, WARNING, ERROR or SILENT

Note – The default trace level is WARNING.

Setting the log trace level to DEBUG accelerates log file roll over.

▼ To Change the Log Trace Level:

1. Log in as root.
2. Stop the Agent with the following command in the $ACSNMP_HOME directory:

   acsnmp_ctl stop
3. From the Bourne shell, enter the value corresponding to the desired log trace level, for example:

   ACS_TRACE=DEBUG
4. To keep this environment variable value outside this shell, enter:

   export ACS_TRACE
5. Check the results of your choice:

   echo $ACS_TRACE
6. Restart the Agent:

   acsnmp_ctl start

CHAPTER 4

Operating the ACSLS Agent

Once the Agent has been installed and configured, there is little in the way of operation or maintenance. The Agent starts and stops automatically. It responds to *get* and *set* requests from remote management applications. And it sends traps registered clients to report ACSLS and library operational events.

Agent Behavior

The ACSLS SNMP Agent operates in a continuous loop, polling the status of ACSLS devices, checking for any status changes, and sending a corresponding trap to registered trap recipients with each relevant change in status. The periodicity of this loop is between 15 and 60 seconds, determined by the MIB parameter, acsTrpCurPollingRate, which is described in Chapter-3.

If ACSLS software should go down for any reason, the SNMP Agent (AcslsAgtd) will also go down, but not completely. A component of the Agent (AcslsReStartAgent) will remain active, watching for the return of ACSLS. When ACSLS is reactivated, the ReStart component will automatically launch the Agent. Should the Agent process be killed for any reason, the ReStart component will reactivate the process. The **ReStart** component also watches over the Agent's API client to ACSLS (snmpssi).

The correct way to manually bring down the ACSLS SNMP Agent is to run *acsnmp_ctl stop* from the **$ACSNMP_HOME** directory.

acsnmp_ctl stop

This command allows you to gently shut down all of the Agent components, including **AcslsAgt, snmpssi,** and **AcslsReStartAgent**.

The ACSLS MIB

The Management Information Base (MIB) is a machine-readable document that lists all object IDs (OIDs) associated with the Agent and defines all of the information that is maintained about each object. The ACS-TAPE-MONITOR-MIB is included in the ACSNMP home directory under the file name AcslsMib.mib. It is a text file that can be copied and transferred to another machine for use by an SNMP management application. Typically such applications will compile

each MIB under its management and will use the information as a means to translate OIDs to the actual objects being monitored by the Agent. You can review a copy of the ACSLS MIB in Appendix A of this document.

The MIB defines an actual database that is maintained by the ACSLS Agent. The ACSLS Agent database contains information about library resources. It includes a current count of configured ACSs, LSMs, Drives, and CAPS in the library. It records the type and the location of each configured tape drive and it knows the size of each CAP in the library. All of this information remains stable and unchanged in the MIB database until ACSLS has been reconfigured and the Agent has restarted.

The MIB database also maintains dynamic information when object statuses change from moment to moment. The Agent keeps track of the number of free cells in each ACS and LSM. The database is updated dynamically as resources are varied offline and back online, or as drives are placed in use or as they become available. If a volume is mounted to a drive, the Agent records the drive status and the volume ID in the MIB database. If a CAP is opened or as CAP priority changes, this information is maintained by the Agent.

When the ACSLS connection is broken, the Agent purges its MIB table entries. ACS, LSM, Drive, and CAP counts will be changed to 0. When a new ACSLS connection is detected, the Agent restarts and a new database will be created from the fresh information reported by ACSLS.

To view a complete list of translated OIDs for your ACSLS system and to see the current status of each MIB object, you can use the *translate* utility in the **$ACSNMP_HOME** directory.

translate

The default behavior of *translate* will display alpha-numeric OIDs. To view the OIDs in their strictly numeric form, use the *-n* option:

translate -n

Similarly, you can quickly *walk* the ACSLS MIB on the local machine using the *walker* utility:

walker -n

To view a complete list of translated OIDs for your ACSLS system and to see the current status of each MIB object, you can use the `translate` utility in the ACSNMP directory /export/home/ACSNMP. (You can verify the installed directory using the command, **pkginfo -r STKacsnmp.)**

translate

The default behavior of translate will display alpha-numeric OIDs. To view the OIDs in their strictly numeric form, use the `-n` option:

translate -n

Similarly, you can quickly walk the ACSLS MIB on the local machine using the `walker` utility:

walker -n

SNMP Traps

A *trap* is an informational message that is sent to registered clients whenever the status of an object has changed. This information can be displayed by a management application in an event console or it can trigger an action defined by the administrator of the management application. There are as many traps defined in the MIB as there are possible statuses returned for the Agent or for the library resources that are monitored by the Agent.

Trap Samples

Agent Start Trap

Agents send a trap every time they are started. These are called *start* traps. Each start trap contains the related boot date for information purposes.

Example

```
ACSLS Agent:
Trap Number = 11
Enterprise OID = 1.3.6.1.4.1.1211.1.11
acsAgtBootDate.0 :
                    OID = 1.3.6.1.4.1.1211.1.11.1.3.0
                    Value : "2002-02-21T05:01"
```

This type of trap can be used by a management application to re-synchronize its own data model with the information available from the Agent. There are numerous reasons to restart the Agent but one typical reason is to allow the Agent to record any hardware configuration changes made to the attached library. Consequently, when receiving this trap, the management application would prudently update the information pertaining to the ACS-TAPE-MONITOR-MIB.

Status Traps

Status traps alert the client that a status change has occurred on a component within the MIB. To facilitate component identification among the collection of components detected by the Agent, object identification along with status information is provided in the trap. For ACS, this information includes the ACS state, index, and ID. For LSMs, this information includes the LSM state and status, the ACS index the LSM index and the LSM ID. For drives, this information includes the drive state and status, the ACS index the LSM index the drive index and the drive ID. For CAPs, this information includes the CAP state and status, the CAP priority, the ACS index the LSM index the CAP index and the CAP ID.

The following example shows the structure of a typical status trap.

Example: Drive State Offline

```
Trap Number = 51
Enterprise OID = 1.3.6.1.4.1.1211.1.11
acsDriveId.1.2.15
                OID = 1.3.6.1.4.1.1211.1.11.3.3.2.1.4.1.2.15
                Value: “0, 1, 10, 2”
acsDriveState.1.2.15
                OID = 1.3.6.1.4.1.1211.1.11.3.3.2.1.6.1.2.15
                Value: 3
acsDriveStatus.1.2.15
                OID = 1.3.6.1.4.1.1211.1.11.3.3.2.1.5.1.2.15
                Value: 1
acsDriveAcsIndex.1.2.15
                OID = 1.3.6.1.4.1.1211.1.11.3.3.2.1.1.1.2.15
                Value: 1
acsDriveLsmIndex.1.2.15
                OID = 1.3.6.1.4.1.1211.1.11.3.3.2.1.2.1.2.15
                Value: 2
acsDriveIndex.1.2.15
                OID = 1.3.6.1.4.1.1211.1.11.3.3.2.1.3.1.2.15
                Value: 15
```

SNMP Client Utilities

The information provided by the ACSLS SNMP Agent is viewable only by means of an SNMP client application. Such applications are designed to submit SNMP queries (e.g. *get, getnext, walk*), to change (*set*) variable parameters in the Agent, and ultimately, to listen for trap messages from the Agent. There is a wide range of commercial applications for network management including HP OpenView, IBM Tivoli NetView, and CA Unicenter. What follows in this section are a few pointers to commonly available tools on standard systems.

NetSNMP

The most widely-accessible client application is *NetSNMP.* This is an open-source, public domain application that is freely available on all popular platforms including Solaris, AIX, HP-UX, Linux, MacOS, and Windows. *NetSNMP* is bundled as a standard package with Solaris-10.

Common *NetSNMP* commands on Solaris reside in /usr/sfw/bin. These include *snmpget, snmpgetnext, snmpbulkget, snmpwalk,* and *snmpset. NetSNMP* is a command-line application and each command contains a structure of parameters. For example, to view a specific OID from the ACSLS MIB, you can use the snmpget command as follows:

snmpget -v1 -m <MIB pathname> -c public <hostname> acsAgtRelease.0

In this example, we are requesting the release level of the ACSLS Agent. The SNMP packet version (v1) is specified since the ACSLS Agent supports only v1 packets. The path to the ACSLS MIB is specified. You would copy the MIB file from the ACSLS machine to any remote machine if you intend to submit MIB-referenced SNMP queries from that machine. The

community name and hostname are also passed as arguments. Finally the object id is specified. In this example, we are asking for the first instance acsAgtRelease. The *snmpget* utility will consult the MIB to translate the object id to its actual numeric value.

If you know the translated OID, you can pass its numeric value directly in a command string that does not require a MIB lookup:

snmpget -v1 -c public <hostname> 1.3.6.1.4.1.1211.1.11.1.1.0

(HINT: You can get a complete list of ACSLS Agent OIDs using *translate -n* in the ACSNMP directory of the ACSLS machine).

A *NetSNMP* trap listener utility resides in /usr/sfw/sbin.

snmptrapd -P -m <*MIB pathname(s)*>

In this example, traps that are sent to the machine from which you launched the trap daemon will be displayed to standard error of your shell environment.

For more information about *NetSNMP*, see the SourceForge web site:

http://net-snmp.sourceforge.net/

Windows NT Resource Kit

The Windows NT resource kit provides a utility called *snmputil*.

c:\ntreskit>**snmputil get** <*hostname*> public .1.3.6.1.4.1.12.11.1.11.1.2.0

The command in this example returns the status of the Agent. Of course, you need to know the OID of the object of your interest and notice that the OID expression in this application always begins with a dot. (You can get a complete list of ACSLS Agent OIDs using *translate -n* in the ACSNMP directory of the ACSLS machine). The Windows *snmptutil* also provides a *walk* function.

AIX snmpinfo

AIX provides the /usr/sbin/snmpinfo command (equivalent to the Windows NT *snmputil* command) which performs SNMPGET and SET requests.

To request the status of the Agent using snmpinfo, the command would be as follows:

/usr/sbin/**snmpinfo -m get -h** <*hostname*> 1.3.6.1.4.1.1211.1.11.1.2.0

The snmpinfo utility supports SNMP *get, getnext, set,* and *dump* operations.

CHAPTER 5

Troubleshooting

This chapter summarizes various diagnostic tools that have been provided and itemizes a number of issues that might arise during the operation of the ACSLS SNMP Agent. With each issue, an explanatory context is provided along with a suggested course of action.

Tools

A general purpose diagnostic tool is provided that can verify Agent configuration and check for multiple dependencies. To run the tool, go to the ACSNMP home directory and run:

pingAcslsAgt

This utility will check the following dependencies:

- Verifies ACSLS is running.
- Verifies the Solstice Enterprise Master Agent, *snmpdx* is running.
- Verifies the System Management Agent, *snmpd* is running.
- Verifies SNMP requests are allowed from the local host.
- Displays which communities are configured to submit requests.
- Verifies that each community is known to the System Management Agent.
- Verifies *read-only* or *read-write* MIB access to each community.
- Exercises a *read-only* operation for each community.
- Looks to see what hosts have been defined as trap recipients.
- Confirms that each host is known to the Enterprise Master Agent.
- Pings each trap destination host to verify the host is reachable.

If any of these dependencies is lacking, the tool will display an appropriate error message. Otherwise, it will display the information that it has confirmed.

Other tools

To review the details of the Agent package installed on your machine.

pkginfo -STKacsnmp

To locate the directory where the Agent has been installed.

pkginfo -r STKacsnmp

To stop the ACSLS Agent

$ACSNMP_HOME/acsnmp_ctl stop

To start the ACSLS Agent

$ACSNMP_HOME/acsnmp_ctl start

Execution Issues

Solstice Enterprise Master Agent Does Not Start

The Master Agent may fail to start if it cannot parse information contained in its configuration files (see chapter-3). When the Master Agent cannot interpret configuration data in these files, a parse error message will be recorded in the system log, /var/adm/messages. Check to be sure that any changes you have made to the configuration files are correct and accurate.

If the hostname of a trap destination cannot be resolved, the Master Agent can fail to start. Make sure that the host is reachable for any trap recipient that has been configured.

The Library Configuration Has Changed

Whenever changes are made to the library hardware, you should reconfigure the ACSLS server (refer to the *ACSLS Installation, Configuration and Administration Guide*) and restart the ACSLS server. The Agent will restart when ACSLS restarts and the hardware changes will be applied to the ACS-TAPE-MONITOR-MIB.

Stopping the ACSLS Server is not always required when adding, removing or changing tape drives. (Refer to the *ACSLS Installation, Configuration and Administration Guide* for details on the *config drives* command.) If the tape drive configuration is changed without restarting ACSLS, you should stop and restart the Agent in order to apply the changes to the ACS-TAPE-MONITOR-MIB.

$ACSNMP_HOME/acsnmp_ctl register

ACSLS Library Server Not Running

If ACSLS is not running, the ACSLS Agent will not run. The agent will remain inactive until ACSLS restarts. The Agent will start automatically after ACSLS starts.

Trouble Removing the Agent

The Agent must be stopped before it can be removed. Any attempt to run **pkgrm STKacsls** while the Agent is running will fail with the message:

ERROR, the Agent is running

SNMP trap errors

If a configured SNMP management application fails to receive traps from the ACSLS Agent, you can verify the trap destination using **pingAcslsAgt**.

If *pingAcslsAgt* reports that the trap destination is OK, you should check whether the destination host machine may be configured with firewall software. You should also check to confirm that the application is listening on the appropriate port. The default trap listener port is 162. If the client is configured to listen on a different port, you can reconfigure the defined trap port of the Agent using *AcslsAgtSnmpConf*:

AcslsAgtSnmpConf -t <*port*>

If *pingAcslsAgt* reports that the trap destination is not configured, then check the file /etc/snmp/conf/snmpdx.acl. Make sure that the desired trap hostname is listed among the hosts in the trap parameters, and make sure that the trap communities are properly defined.

If *pingAcslsAgt* reports that the trap destination is unreachable, then check to see that the hostname is defined in your NIS database or in your local /etc/hosts file. You should use a standard ping to verify that the remote host is reachable from the Agent machine. If ping fails, you should resolve any routing issues on your local network.

If *pingAcslsAgt* reports *none* for the trap destination, use the *AcslsAgtdTrapDest* routine to add the appropriate host name.

AcslsAgtdTrapDest -a <*hostname*>

You will need to restart the Agent if you have made any configuration changes to the trap destination list or to the port.

Note – When restarting the Agent, the SNMP start trap is sent with the boot date.

SNMP Requests Generate SNMP Timeouts

If the Agent is running but there is no sign of activity in response to a *get* or *set* request, you should use *pingAcslsAgt* to verify the community configuration. Go to the $ACSNMP directory.

For *get* requests, make sure that the specified community has *read-only* or *read-write* access. For *set* requests, make sure that the specified community has *read-write* access.

SNMP Requests Generate “Connection refused” Error

The *connection refused* error will be returned whenever the SNMP Manager is trying to retrieve information on an unauthorized port. This is usually due to a difference between the SNMP port used by the Master Agent and the port used by the ACSLS Agent.

The default port is 161. If you change this port number from the default, make sure the port you specify agrees with any changes that may have been specified in the master agent startup script. To quickly verify the string that launches the master agent, use the following grep expression.

You can easily determine whether the SEA agent is controlled by SMF by running the following command:

svcs snmpdx

If the system responds with *online* or other logical state, this is a sign that SEA is under SMF control.

If the SEA is controlled by SMF, the **e**xpression is

grep SNMP_BIN /lib/svc/method/svc-snmpdx

If the SEA is controlled by the legacy rc script, then the expression is

grep "snmpdx -f" /etc/rc3.d/S76snmpdx

If the string that launches the master agent includes "-p <port number>", this implies that the default request port, 161, is not being used. You can redefine the port as follows. From the ACSNMP home directory, issue the following command from the $ACSNMP_HOME directory:

AcslsAgtdSnmpConf -p < port number>

The port number you configure for the ACSLS agent should agree with the port number that was defined for the master agent. Once you have changed the port configuration, you should restart the ACSLS Agent.

ACSLS SNMP Agent Starts but then Stops after One Minute

If the ACSNMP/AcslsAgtd.log file includes messages, “failed to connect to Master Agent”, this is a sign that the ACSLS Agent could not determine whether the Master Agent was active. Either of the following could account for this:

- The Master Agent is not running.
- The Master Agent and the Agent do not use the same SNMP request port.

You can verify the common port using the procedure described just above.

To restart the master agent, use the following procedure:

$**ACSNMP_HOME/acsnmp_ctl register**

Log Files

When troubleshooting problems with the ACSLS Agent, be sure to look at clues that are left behind in the Agent log. This log file resides in the $ACSNMP directory under the name **AcslsAgt.log**.

Another useful file to inspect when SNMP fails to start is the system messages file, **/var/adm/messages.**

Master Agent Tracing

In the event that the logs provide no clues to the problem you are experiencing, you can extract trace information from request traffic between the System Management Agent and the Solstice Enterprise Master Agent.

1. First stop the Solstice Enterprise Agent (SEA).

   There are two ways to stop the SEA. The method you choose will depend on whether the SEA agent is managed by the System Management Facility (SMF) in Solaris-10 or if it is using the legacy startup scripts in /etc/rc3.d.

   You can easily determine whether the SEA agent is controlled by SMF by running the following command:

   svcs snmpdx

   - If the system responds with online or another logical state, this is a sign that SEA is under SMF control. To stop the Solstice Enterprise Agent using SMF, use the following command:

     svcadm disable snmpdx

   - If the response to *svcs* states that snmpdx does not match any instances, this is a sign that the SEA is managed by the legacy startup scrpts in /etc. In this case, the command to stop the SEA is as follows:

     /etc/rc3.d/S76snmpdx stop

2. Restart the master agent as follows:

   /usr/lib/snmp/snmpdx -f 0 -y -c /etc/snmp/conf -d 4

   The packets bound to the Master Agent will be displayed to the shell where you invoked the snmpdx command. Keep this shell open for display.

3. Open a new window and generate request traffic by restarting the System Management Agent.

   - If the Systme Management Agent is controlled by SMF, the command is

     svcadm restart sma

   - If the SMA is controlled by legacy startup scripts, the command is

     /etc/init.d/init.sma restart

   When the System Management Agent restarts, you should see traffic activity in the shell window where you started the Master Agent.

   If you do not see trace activity, this points to a problem between the System Management Agent and the Solstice Enterprise Master Agent. Review any changes you may have made in the files **/etc/sma/snmp/snmpd.conf** and **/etc/snmp/conf/snmpdx.conf**.

   For further information, consult the *Solaris System Management Agent Administration Guide* http://192.18.109.11/817-3000/817-3000.pdf.

   If the trace window shows transaction activity, try sending a command from the ACSLS Agent. One easy way to do this is to run the *walker* utility from the $ACSNMP_HOME directory. This simple test utility will submit a series of *snmpget* requests through the interface to the master agent. If the trace window reveals activity, it confirms good communication via the request port (161) between the ACSLS Agent and the System Management Agent. The information shown in the trace may offer further hints by posting error messages that are relevant to the root of the problem.

   If the trace window remains static in response to the *walker* command, it points to communication problems between the ACSLS Agent and the System Management Agent. Review any changes you may have made in the files /etc/snmp/conf/AcslsAgt.acl, /etc/snmp/conf/snmpdx.acl and /etc/snmp/conf/snmpd.conf.

Essential Commands

The following table provides a list the most commonly used commands for the SNMP Agent.

~ACSNMP/acsnmp_ctl start	Starts the ACSLS agent
~ACSNMP/acsnmp_ctl register	Registers changes to the ACSLS agent and restarts
~ACSNMP/acsnmp_ctl register	Restarts both master agents and the ACSLS agent
~ACSNMP/acsnmp_ctl stop	Stops the ACSLS agent
~ACSNMP/AcslsAgtdTrapDest	Adds or removes trap target addresses
~ACSNMP/AcslsAgtdSnmpConf	Configures Agent community and ports
~ACSNMP/pingAcslsAgt	General purpose ACSLS Agent diagnostic utility
~ACSNMP/walker	Simple test probe of the entire ACSLS MIB.
~ACSNMP/translate	Shows all ACSLS OIDs and their values.

Note – ACSNMP is the installation directory for the ACSLS Agent, $ACSNMP_HOME. The default directory is /export/home/ACSNMP