Monitoring Servers Using SNMP

Prime Cable Provisioning supports management of servers via SNMP. Specifically, an SNMP-based management system can be used to monitor Prime Cable Provisioning server state, license utilization information, server connections, and server-specific statistics.

- SNMP Agent, on page 1
- Using snmpAgentCfgUtil.sh Tool, on page 3

SNMP Agent

Prime Cable Provisioning provides basic SNMP v2-based monitoring of the RDU and DPE servers. The Prime Cable Provisioning SNMP agents support SNMP informs and traps, collectively called notifications. You can configure the SNMP agent on the DPE using snmp-server CLI commands, and on the RDU using the SNMP configuration command-line tool.

For additional information on the SNMP configuration command-line tool, see Using snmpAgentCfgUtil.sh Tool. For additional information on the DPE CLI, see the Cisco Prime Cable Provisioning 6.1.2 DPE CLI Reference Guide.

MIB Support

Prime Cable Provisioning supports several different MIBs. The following table summarizes MIB support for each Prime Cable Provisioning component.

Table 1: Prime Cable Provisioning-Supported MIBs

Component	MIBs Supported
DPE	CISCO-BACC-SERVER-MIB
	CISCO-BACC-DPE-MIB
RDU	CISCO-BACC-SERVER-MIB
	CISCO-BACC-RDU-MIB

The SNMP agent supports the CISCO-BACC-SERVER-MIB. This MIB defines the managed objects that are common to all servers on Prime Cable Provisioning. This MIB supports the monitoring of multiple Prime

Cable Provisioning servers when they are installed on the same device. The ciscoBaccServerStateChanged notification is generated every time a server state change occurs.

The RDU SNMP agent supports the CISCO-BACC-RDU-MIB, which defines managed objects for the RDU. This MIB defines statistics related to the state of the RDU and the statistics on the communication interface between the RDU and DPE and between the RDU and Network Registrar.

The SNMP agent generates a cnaHealthNotif trap that announces that the RDU server has started, shut down, or failed, or there is a change in the exit status.

The DPE SNMP agent supports the CISCO-BACC-DPE-MIB, which defines managed objects for the components installed on a DPE. The DPE manages local caching of device configurations and configuration files used by all supported devices. This MIB provides some basic DPE configuration and statistics information, including entries for TFTP and ToD servers.

The SNMP agent also supports the CISCO-NMS-APPL-HEALTH-MIB, which defines the Cisco NMS application health status notifications and related objects. These notifications are sent to the OSS/NMS to inform them about the NMS application status, including: started, stopped, failed, busy, or any abnormal exit of applications. The default MI is MIB-II.

Note For a description of all objects, see the corresponding MIB files in the *BPR_HOME/rdu/mibs* directory.

The following table lists the Prime Cable Provisioning RDU SNMP Traps:

Table 2: Prime Cable Provisioning RDU SNMP Traps

MIB	Trap Name	Trap OID	Sub Type Varbind OID	Sub Type Varbind Value	Sub Type Varbind Value Description	Trap Description
CISCO-BAC C-RDU-MIB	ciscoBaccRdu LicenseLimit	.1.3.6.1.4. 1.9.9.353. 0.0	.1.3.6.1.4.1.9.9.353.1. 1.2.1.2.(1-n) (LicenseName)	*[Technologyname]*	Indicates the corresponding technology name of the license. For example, DOCSIS, PacketCable and so on.	The notification appears when the number of devices exceeds the limit allowed by the license for a specific technology.
			.1.3.6.1.4.1.9.9.353.1. 1.2.1.3.(1 - n) (LicenseMaxAllowed)	0..4294967295	Indicates the total number of devices or server components allowed for the technology.	
			.1.3.6.1.4.1.9.9.349.1. 1.1.1.3.(1 - n) (cbsState)	0..4294967295	Indicates the total number of licenses of specific technology type already in use.	

MIB	Trap Name	Trap OID	Sub Type Varbind OID	Sub Type Varbind Value	Sub Type Varbind Value Description	Trap Description
CISCO-BACC-SERVERMIB	ciscoBaccServerStateChanged	.1.3.6.1.4.1.9.9.349.0.0	.1.3.6.1.4.1.9.9.349.1.1.1.3.(1 - n) (cbsState)	1..8	Indicates the status of the server.	This notification appears when the status of the server is changed: • Unknown (1) • initializing (2) • disconnected (3) • shuttingDown(4) • readyOverloaded (5) • ready (6) • offline (7) • unlicensed (8)
			.1.3.6.1.4.1.9.9.349.1.1.1.6.(1 - n) (cbsServerType)	[ServerType] RDU,DPE,etc.	A unique name identifying the type of the server. For example: RDU, DPE and so on.	
			.1.3.6.1.2.1.1.5.0 (sysName)	DisplayString (SIZE (0..255))	An administratively-assigned name for the managed node. By convention, this is the fully-qualified domain name of the node. If the name is unknown, the value is a zero-length string.	

Using snmpAgentCfgUtil.sh Tool

You can use the **snmpAgentCfgUtil.sh** tool to manage the SNMP agent installed on a Linux computer. Using this tool, which resides in the *BPR_HOME/snmp/bin* directory, you can add (or remove) your host to a list of other hosts that receive SNMP notifications, and start and stop the SNMP agent process. This tool should be run from the local directory.

Note The default port number of an SNMP agent running on a Linux computer is 8001.

You can use the RDU SNMP agent for:

- Adding a Host
- Deleting a Host
- Adding an SNMP Agent Community

- Deleting an SNMP Agent Community
- Starting the SNMP Agent
- Stopping the SNMP Agent
- Configuring an SNMP Agent Listening Port
- Changing the SNMP Agent Location
- Setting Up SNMP Contacts
- Displaying SNMP Agent Settings
- Specifying SNMP Notification Types

Adding a Host

You use this command to add the host address to the list of hosts that receive SNMP notifications from the SNMP agent.

Syntax Description

snmpAgentCfgUtil.sh add host *ip-addr* **community** *community* [**udp-port** *port*]

- *ip-addr*—Specifies the IP address of the host to which notifications are sent.
- *community*—Specifies the community (read or write) to be used while sending SNMP notifications.
- *port*—Identifies the UDP port used for sending the SNMP notifications.

Example

```
# ./snmpAgentCfgUtil.sh add host 10.10.10.5 community trapCommunity udp-port 162
OK
Please restart [stop and start] SNMP agent.
```

Note The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the **/etc/init.d/bprAgent restart snmpAgent** command. For detailed information, see Prime Cable Provisioning Process Watchdog.

Deleting a Host

You use this command to remove a host from the list of those receiving SNMP notifications from the SNMP agent.

Syntax Description

snmpAgentCfgUtil.sh delete host *ip-addr*

ip-addr—Specifies the IP address of the host that you want to delete from the list of hosts.

Examples

```
# ./snmpAgentCfgUtil.sh delete host 10.10.10.5
OK
Please restart [stop and start] SNMP agent.
```

Note The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the **/etc/init.d/bprAgent restart snmpAgent** command. For detailed information, see Prime Cable Provisioning Process Watchdog.

Adding an SNMP Agent Community

You use this command to add an SNMP community string to allow access to the SNMP agent.

Syntax Description

snmpAgentCfgUtil.sh add community *string* [**ro** | **rw**]

- *string*—Identifies the SNMP community.
- **ro**—Assigns a read-only (**ro**) community string. Only *get* requests (queries) can be performed. The ro community string allows *get* requests, but no *set* operations. The NMS and the managed device must reference the same community string.
- **rw**—Assigns a read-write (**rw**) community string. SNMP applications require read-write access for *set* operations. The **rw** community string enables write access to OID values.

Note The default **ro** and **rw** community strings are baccread and baccwrite, respectively. We recommend that you change these values before deploying Prime Cable Provisioning.

Examples

```
# ./snmpAgentCfgUtil.sh add community fsda54 ro
OK
Please restart [stop and start] SNMP agent.
```

Note The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the **/etc/init.d/bprAgent restart snmpAgent** command. For detailed information, see Prime Cable Provisioning Process Watchdog.

Deleting an SNMP Agent Community

You use this command to delete an SNMP community string to prevent access to the SNMP agent.

Syntax Description

snmpAgentCfgUtil.sh delete community *string* [**ro** | **rw**]

- *string*—Identifies the SNMP community.
- **ro**—Identifies the specified community as a read-only one.
- **rw**—Identifies the specified community as a read-write one.

Note See Adding an SNMP Agent Community, for additional information on the **ro** and **rw** community strings.

Examples

```
# ./snmpAgentCfgUtil.sh delete community fsda54 ro
OK
Please restart [stop and start] SNMP agent.
```

Note The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the **/etc/init.d/bprAgent restart snmpAgent** command. For detailed information, see Prime Cable Provisioning Process Watchdog.

Starting the SNMP Agent

You use this command to start the SNMP agent process on a Linux computer on which Prime Cable Provisioning is installed.

Note You can also start the SNMP agent by invoking the Prime Cable Provisioning process watchdog using the **/etc/init.d/bprAgent start snmpAgent** command. For more information, see Using Prime Cable Provisioning Process Watchdog from CLI.

Example:

```
# ./snmpAgentCfgUtil.sh start
Process snmpAgent has been started
```

Stopping the SNMP Agent

You use this command to stop the SNMP agent process on a Linux computer on which Prime Cable Provisioning is installed.

Note You can also stop the SNMP agent by invoking the Prime Cable Provisioning process watchdog using the **/etc/init.d/bprAgent stop snmpAgent** command. For more information, see Using Prime Cable Provisioning Process Watchdog from CLI.

Examples

```
#./snmpAgentCfgUtil.sh stop
Process snmpAgent has stopped
```

Configuring an SNMP Agent Listening Port

You use this command to specify the port number that the SNMP agent will listen to. The default port number used by RDU SNMP agent is 8001.

Syntax Description

snmpAgentCfgUtil.sh udp-port *port*

port identifies the port number that the SNMP agent will listen to.

Examples

```
# ./snmpAgentCfgUtil.sh udp-port 8001
OK
Please restart [stop and start] SNMP agent.
```

Note The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the **/etc/init.d/bprAgent restart snmpAgent** command. For detailed information, see Prime Cable Provisioning Process Watchdog.

Changing the SNMP Agent Location

You use this command to enter a string of text that indicates the location of the device running the SNMP agent. This could, for example, be used to identify the physical location of the device. You can enter any character string that is fewer than 255 characters.

Syntax Description

snmpAgentCfgUtil.sh location *location*

location is the character string identifying the agent's location.

Examples

In this example, the physical location of the SNMP agent is in an equipment rack identified as rack 5D:

```
# ./snmpAgentCfgUtil.sh location "equipmentrack5D"
OK
Please restart [stop and start] SNMP agent.
```

Note The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the **/etc/init.d/bprAgent restart snmpAgent** command. For detailed information, see Prime Cable Provisioning Process Watchdog.

Setting Up SNMP Contacts

You can use this command to enter a string of text that identifies the contact person for the SNMP agent, together with information on how to contact this person. This could, for example, be used to identify a specific person including that person's telephone number. You can enter any character string that is fewer than 255 characters.

Syntax Description

snmpAgentCfgUtil.sh contact *contact-info*

contact-info is the character string identifying the individual to contact concerning the SNMP agent.

Examples

In this example, the contact name is Terry and the telephone extension is 1234:

```
# ./snmpAgentCfgUtil.sh contact "Terry-ext1234"
OK
Please restart [stop and start] SNMP agent.
```

Note The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the **/etc/init.d/bprAgent restart snmpAgent** command. For detailed information, see Prime Cable Provisioning Process Watchdog.

Displaying SNMP Agent Settings

You use this command to display all current SNMP settings.

Examples

```
# ./snmpAgentCfgUtil.sh show
Location                      : equipmentrack5D
Contact                       : Terry-ext1234
Port Number                   : 8001
Notification Type             : trap
Notification Recipient Table  :
        [ Host IP address, Community, UDP Port ]
        [ 10.10.10.5 , trapCommunity , 162 ]
Access Control Table          :
        Read Only Communities
                baccread
        Read Write Communities
                baccwrite
```

Specifying SNMP Notification Types

You use this command to specify the types of notifications (traps or informs) that will be sent from the SNMP agent. By default, traps are sent, though you can set the agent to send SNMP informs instead.

Note For the SNMP trap feature to work, you must enable the notification flag. In other words, the value for the MIB variable 0cbsNotifEnableFlags (OID = .1.3.6.1.4.1.9.9.349.1.1.1.1.5.1) must be set to 1.

Syntax Description

snmpAgentCfgUtil.sh inform [**retries timeout**] | **trap**

Where the parameter is the backoff timeout between retries.

Examples

```
# ./snmpAgentCfgUtil.sh inform retries 3 timeout 1000
OK
Please restart [stop and start] SNMP agent.
```

Note The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the **/etc/init.d/bprAgent restart snmpAgent** command. For detailed information, see Prime Cable Provisioning Process Watchdog.

Use the **snmpAgentCfgUtil.sh show** command to verify your configuration settings.

```
# ./snmpAgentCfgUtil.sh show
Location                      : equipmentrack5D
Contact                       : Terry-ext1234
Port Number                   : 8001
Notification Type             : inform
Notification Retries          : 3
Notification Timeout          : 1000
Notification Recipient Table  :
        [ Host IP address, Community, UDP Port ]
        [ 10.10.10.5 , trapCommunity , 162 ]
Access Control Table          :
        Read Only Communities
                baccread
        Read Write Communities
                baccwrite
```