Oracle® Integrated Lights Out Manager (ILOM) 3.0

Management Protocols Reference Guide
SNMP, IPMI, WS-Man, CIM

Part No. 820-6413-13
October 2010, Revision A

Please
Recycle

Contents

Using This Documentation

This guide describes how to perform the required procedures to access Oracle Integrated Lights Out Manager (ILOM) 3.0 functions using the Simple Network Management Protocol (SNMP). This guide also provides descriptions of the procedures you can perform to access ILOM functions using the Intelligent Platform Management Interface (IPMI). Finally, this guide describes the standards supported in ILOM for managing servers using a Web Service Management (WS-Man) and Common Information Model (CIM).

This guide is written for technicians, system administrators, authorized service providers, and users who have experience managing system hardware.

To fully understand the information that is presented in this guide, use the concepts guide in conjunction with other guides in the ILOM 3.0 Documentation Collection. For a description of the guides that comprise the ILOM 3.0 Documentation Collection, see "Related Documentation" on page xiv.

This preface contains the following topics:

Related Documentation

The following table list the guides that comprise the ILOM 3.0 Documentation Collection. You can access or download these guides online at:

http://docs.sun.com/app/docs/prod/int.lights.mgr30#hic

Note – The documents comprising the collection were formerly referred to as Sun Integrated Lights Out Manager (ILOM) 3.0 guides.

Title	Content	Part Number	Format
Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide	Information that describes ILOM features and functionality	820-6410	PDF HTML
Oracle Integrated Lights Out Manager (ILOM) 3.0 Getting Started Guide	Information and procedures for network connection, logging in to ILOM for the first time, and configuring a user account or a directory service	820-5523	PDF HTML
Oracle Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide	Information and procedures for accessing ILOM functions using the ILOM web interface	820-6411	PDF HTML
Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide	Information and procedures for accessing ILOM functions using the ILOM CLI	820-6412	PDF HTML
Oracle Integrated Lights Out Manager (ILOM) 3.0 Management Protocols Reference Guide	Information and procedures for accessing ILOM functions using SNMP or IPMI management hosts	820-6413	PDF HTML
Oracle Integrated Lights Out Manager (ILOM) 3.0 CMM Administration Guide for Sun Blade 6000 and 6048 Modular Systems	Information and procedures for managing CMM functions in ILOM.	820-0052	PDF HTML
Oracle Integrated Lights Out Manager (ILOM) 3.0 Feature Updates and Release Notes	Late breaking information about new ILOM 3.0 features, as well as known problems and work arounds.	820-7329	PDF HTML

In addition to the ILOM 3.0 Documentation Collection, associated ILOM Supplement or Platform Administration guides present ILOM features and tasks that are specific to the server platform you are using. Use the ILOM 3.0 Documentation Collection in conjunction with the ILOM Supplement or Platform Administration guide for your server platform.

Documentation, Support, and Training

These websites provide additional resources:

- Documentation `http://docs.sun.com`
- Support `http://www.sun.com/support/`
- Training `http://www.sun.com/training/`

ILOM 3.0 Version Numbers

ILOM 3.0 has implemented a new version numbering scheme to help you identify which version of ILOM you are running on your system. The numbering scheme includes a five-field string, for example, `a.b.c.d.e`, where:

- `a` – Represents the major version of ILOM.
- `b` – Represents a minor version of ILOM.
- `c` – Represents the update version of ILOM.
- `d` – Represents a micro version of ILOM. Micro versions are managed per platform or group of platforms. See your platform Product Notes for details.
- `e` – Represents a nano version of ILOM. Nano versions are incremental iterations of a micro version.

For example, ILOM 3.1.2.1.a would designate:

- ILOM 3 as the major version of ILOM
- ILOM 3.1 as a minor version of ILOM 3
- ILOM 3.1.2 as the second update version of ILOM 3.1
- ILOM 3.1.2.1 as a micro version of ILOM 3.1.2
- ILOM 3.1.2.1.a as a nano version of ILOM 3.1.2.1

Documentation Comments

Submit comments about this document by clicking the Feedback[+] link at:

http://docs.sun.com.

Please include the title and part number of your document with your feedback:

Oracle Integrated Lights Out Manager (ILOM) 3.0 Management Protocols Reference Guide SNMP, IPMI, WS-Man, CIM, part number 820-6413-13.

PART I SNMP

Part 1 of this document provides an overview of the Simple Network Management Protocol (SNMP), and descriptions of the procedures you can perform to access ILOM functions.

CHAPTER 1

SNMP Overview

Topics

Description	Links
Learn about SNMP, SNMP components, and SNMP MIBs	• "About Simple Network Management Protocol" on page 4
Learn about preparing your system to use SNMP, SNMP components, and SNMP MIBs	• "Preparing Your System to Use SNMP" on page 5 • "SNMP Components" on page 5 • "ILOM SNMP MIBs" on page 6
Quick reference of Net-SNMP Command examples	• "SNMP Command Examples" on page 295

Related Topics

For ILOM	Section	Guide
• Concepts	• ILOM Overview	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide* (820-6410)
• CLI	• CLI Overview	*Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide* (820-6412)

Related Topics

For ILOM	Section	Guide
• Web interface	• Web Interface Overview	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide* (820-6411)
• IPMI	• IPMI Overview	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Management Protocols Reference Guide* (820-6413)

The ILOM 3.0 Documentation Collection is available at:
http://docs.sun.com/app/docs/prod/int.lights.mgr30#hic

About Simple Network Management Protocol

ILOM supports the Simple Network Management Protocol (SNMP), which is used to exchange data about network activity. SNMP is an open, industry-standard protocol technology that enables the management of networks and devices, or nodes, that are connected to the network. Using SNMP, data travels between a managed device (node) and a management station with network access. A managed device can be any device that runs SNMP, such as hosts, routers, web servers, or other servers on the network. SNMP messages are sent over IP using the User Datagram Protocol (UDP). Any management application that supports SNMP can manage your server.

For a more complete description of SNMP, see the SNMP five-part, introductory tutorial available at:

http://www.dpstele.com/layers/l2/snmp_l2_tut_part1.php

ILOM supports SNMP versions 1, 2c, and 3. Using SNMP v3 is strongly advised since SNMP v3 provides additional security, authentication, and privacy beyond SNMP v1 and v2c.

SNMP is a protocol, not an application, so you need an application to utilize SNMP messages. Your SNMP management software might provide this functionality, or you can use an open source tool like Net-SNMP, which is available at:

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

Note – ILOM users reading this document are assumed to have a working knowledge of SNMP. SNMP client-side commands are used in this text as examples of using SNMP. Users who do not have a working knowledge of SNMP should complete the tutorial at: `http://net-snmp.sourceforge.net/wiki/index.php/Main_Page`
This tutorial is more advanced than the introductory tutorial referred to above.

Preparing Your System to Use SNMP

To prepare your system to use SNMP, you must download and install the latest version (version 5.2.1 or higher) of Net-SNMP that works with the operating system of your management station or the SNMP tool of your choice.

For more information about preparing your system to use SNMP, see one of the following guides:

- *Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide*
- *Oracle Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide*

SNMP Components

SNMP functionality requires the following two components:

- **Network management station** – A network management station hosts management applications, which monitor and control managed nodes.
- **Managed node** – A managed node is a device such as a server, router, or hub that hosts SNMP management agents that are responsible for carrying out requests from management stations, such as a service processor (SP) running ILOM. Managed nodes can also provide unsolicited status information to a management station in the form of a trap.

SNMP is the protocol used to communicate management information between management stations and SNMP agents.

The SNMP agent is preinstalled on your Oracle Sun server platform and runs on ILOM, so all SNMP management occurs through ILOM. To utilize this feature, your operating system must have an SNMP client application.

Both management stations and agents use SNMP messages to communicate. Management stations can send and receive information. Agents can respond to requests and send unsolicited messages in the form of traps. Management stations and agents use the following functions:

- Get
- GetNext
- GetResponse
- Set
- Trap

ILOM SNMP MIBs

The base component of an SNMP implementation is the Management Information Base (MIB). A MIB is a text file that describes a managed node's available information. This tree-like, hierarchical system classifies information about resources in a network as a list of data objects, each with a unique identifier, or object ID. Thus, the MIB defines the data objects, or variables, that the SNMP agent can access. When a management station requests information from a managed node, the agent receives the request and retrieves the appropriate information from the MIBs. In ILOM, the MIB makes it possible to access the server's network configuration, status, and statistics.

As of ILOM 3.0.4, SNMP MIBs are a part of the ILOM firmware. You can download MIBs directly from ILOM. For more information about MIBs, and instructions for downloading MIBs from ILOM, see the following guides:

- *Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide*
- *Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide*
- *Oracle Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide*

FIGURE 1-1 shows the standard MIB tree and the location of the ILOM MIB modules in that tree. The ILOM MIB modules are highlighted in boldface text.

FIGURE 1-1 Location of ILOM MIB Modules

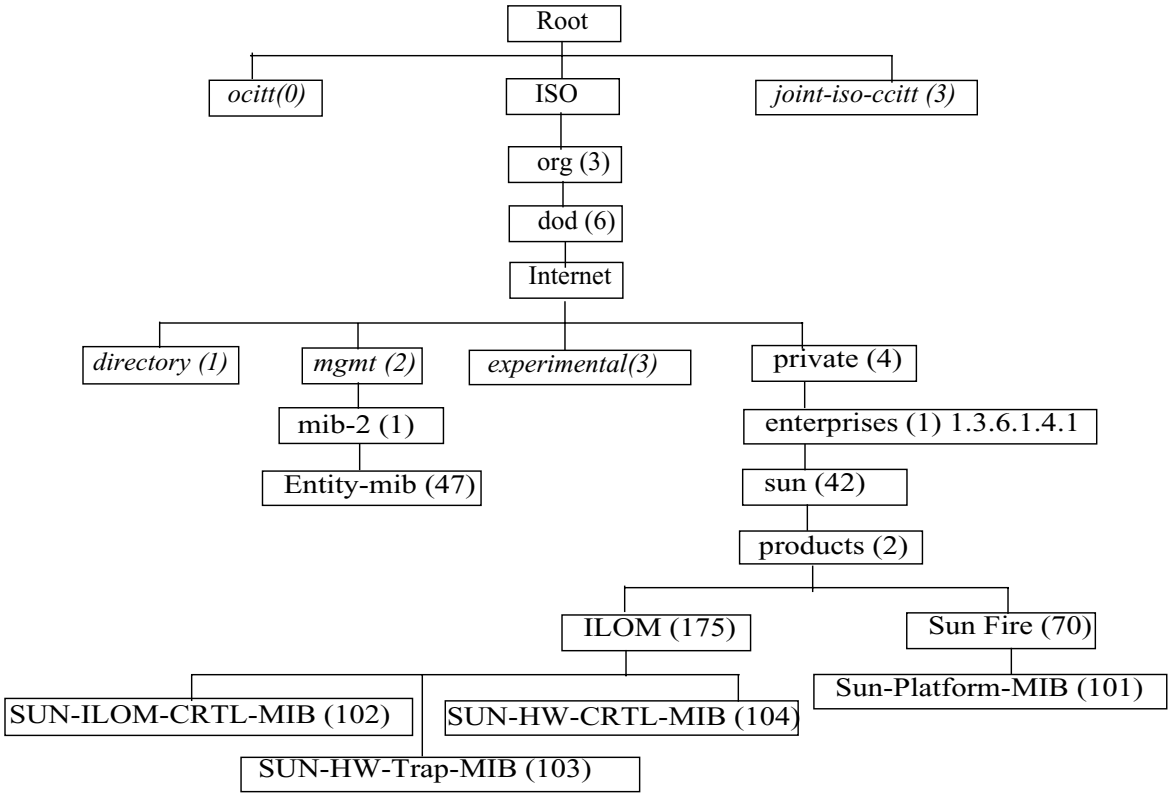


TABLE 1-1 provides a description of the ILOM MIB modules and lists the object ID for each MIB name.

TABLE 1-1 SNMP MIBs Used With ILOM

MIB Name	Description	MIB Object ID
ENTITY-MIB	The MIB module for representing multiple physical entities supported by a single SNMP agent. **Note -** The entPhysicalTable is the only part of this MIB that is implemented.	1.3.6.1.2.1.47
SUN-HW-CTRL-MIB	This MIB allows controls for all Oracle Sun server platform devices using ILOM. **Note -** Only the Power Management portions of this MIB are implemented.	1.3.6.1.4.1.42.2.175.104
SUN-HW-TRAP-MIB	This MIB describes the hardware related notifications/traps that may be generated by Oracle Sun server platforms.	1.3.6.1.4.1.42.2.175.103

TABLE 1-1 SNMP MIBs Used With ILOM *(Continued)*

MIB Name	Description	MIB Object ID
SUN-ILOM-CONTROL-MIB	This MIB provides objects for configuring and managing all ILOM functions. Configuration covered by this MIB includes functions such as authorization, authentication, logging, services, networking, and firmware management.	1.3.6.1.4.1.42.2.175.102
SUN-PLATFORM-MIB	This MIB provides extensions to the ENTITY-MIB (RFC 2737) where each entity modeled in the system is represented by means of extensions to the entPhysicalTable.	1.3.6.1.4.1.42.2.70.101

Portions of the standard MIBs listed in TABLE 1-2 are implemented by ILOM.

TABLE 1-2 Standard MIBs Implemented by ILOM

MIB Name	Description	MIB Object ID
IF-MIB	The MIB module for describing generic objects for network interface sub-layers. This MIB is an updated version of MIB-II's ifTable, and incorporates the extensions defined in RFC 1229.	1.3.6.1.2.1.31
IP-MIB	The MIB module for managing IP and ICMP implementations, but excluding their management of IP routes.	1.3.6.1.2.1.4.
SNMP-FRAMEWORK-MIB	The SNMP Management Architecture MIB.	1.3.6.1.6.3.10
SNMPv2-MIB	The MIB module for SNMP entities. **Note -** Only the system and SNMP groups from this MIB module apply to ILOM.	1.3.6.1.6.3.1
TCP-MIB	The MIB module for managing TCP implementations.	1.3.6.1.2.1.49
UDP-MIB	The MIB module for managing UDP implementations.	1.3.6.1.2.1.50

TABLE 1-3 describes MIBs that are used in support of the ILOM SNMP implementation.

TABLE 1-3 MIBs Used in Support of the ILOM SNMP Implementation

MIB Name	Description	MIB Object ID
HOST-RESOURCES-MIB	This MIB is for use in managing host systems. This MIB supports attributes common to all internet hosts including, for example, both personal computers and systems that run variants of UNIX.	1.3.6.1.2.1.25.1
IANAifType-MIB	This MIB module defines the IANAifType Textual Convention, and thus the enumerated values of the ifType object defined in MIB-II's ifTable.	1.3.6.1.2.1.30
NOTIFICATION-LOG-MIB	This MIB module is used for logging SNMP notifications (traps).	1.3.6.2.1.92.1.1.3
SNMP-MPD-MIB	This MIB module is used for Message Processing and Dispatching.	1.3.6.1.6.3.11
SNMPv2-TM	This MIB module is used for SNMP transport mappings.	1.3.6.1.6.3.19
SNMPv2-SMI	This MIB module contains definitions for the structure of management information, version 2.	1.3.6.1.6

CHAPTER 2

Configuring SNMP Settings in ILOM

Topics

Description	Links
Verify and configure the proper ILOM values for enabling SNMP on the managed device	• "Enabling SNMP on Managed Server Using the CLI" on page 12 • "Enabling SNMP On Managed Server Using the Web Interface" on page 25
Configure SNMP user accounts and communities in ILOM	• "Managing SNMP User Accounts and Communities Using the ILOM CLI" on page 15 • "Enabling SNMP on Managed Server Using the CLI" on page 12
Manage SNMP trap settings in ILOM	• "Managing SNMP Traps Using the ILOM CLI" on page 20
Reference of Net-SNMP command examples	• "SNMP Command Examples" on page 295

Related Topics

For ILOM	Section	Guide
• Concepts	• User Account Management	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide* (820-6410)
• Web	• Managing User Accounts	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide* (820-6411)

The ILOM 3.0 Documentation Collection is available at:
`http://docs.sun.com/app/docs/prod/int.lights.mgr30#hic`

Related Topics

For ILOM	Section	Guide
• CLI	• Managing User Accounts	*Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide* (820-6412)

The ILOM 3.0 Documentation Collection is available at:
`http://docs.sun.com/app/docs/prod/int.lights.mgr30#hic`

Configuring SNMP Settings in ILOM Using the CLI

You can choose to configure the SNMP settings in the ILOM CLI or web interface. This section describes the procedures for configuring the SNMP settings using the ILOM CLI.

Topics

Description	Links
Verify and configure the proper ILOM values for enabling SNMP on the managed server	• "Enabling SNMP on Managed Server Using the CLI" on page 12
Configure SNMP user accounts and communities in ILOM	• "Managing SNMP User Accounts and Communities Using the ILOM CLI" on page 15
Set SNMP trap properties in ILOM	• "Managing SNMP Traps Using the ILOM CLI" on page 20

Enabling SNMP on Managed Server Using the CLI

Prior to using the `snmpgets` commands described in this guide, you should verify that the correct properties for SNMP are enabled in ILOM. Use the following procedure in this section to view and configure the ILOM CLI SNMP properties on the managed server.

Topics

Description	Links
Review requirements and special considerations	• "Before You Begin" on page 13
Verify and configure the proper SNMP properties in ILOM are set on the managed server	• "Configure SNMP On Managed Server Using the ILOM CLI" on page 13

Before You Begin

- To set SNMP properties in ILOM, you must have the Admin role (a) enabled.
- The SNMP `servicestate` property is `enabled` by default.
- To enable write access when using SNMP, the SNMP `sets` property in ILOM must be enabled. This property is `disabled` by default.

Note – If the `sets` property is disabled in ILOM, all SNMP objects are read-only and no snmpset commands will be processed.

- In ILOM there are three SNMP protocol version properties: `v1`, `v2c`, and `v3`. For `v1` and `v2c`, "communities" are used for authentication. For `v3`, "users" are used for authentication. By default, `v1` and `v2c` are disabled, even though there are default "communities" (public and private) provided. By default, `v3` is enabled, but there are no pre-packaged users.

▼ Configure SNMP On Managed Server Using the ILOM CLI

1. **Log in to the ILOM SP CLI.**
2. **To view the ILOM SNMP properties, type:**

   `show /SP/services/snmp`

   The following SNMP output appears.

```
/SP/services/snmp
   Targets:
       communities
       mibs
       users

   Properties:
       engineid = (none)
       port = 161
       servicestate = enabled
       sets = disabled
       v1 = disabled
       v2c = disabled
       v3 = enabled

   Commands:
       cd
       set
       show
```

3. **Use the** `set` **command to change any of the SNMP properties, for example:**
   - To enable SNMP and allow read-only only access, you must enable the `servicestate` property by typing:

     -> **`set /SP/services/snmp servicestate=enabled`**
   - To enable write privilegeswhen using SNMP, you must enable the SNMP `sets` property by typing:

     -> **`set /SP/services/snmp sets=enabled`**
   - To enable a specific SNMP protocol version (`v1`, `v2c`, or `v3`) property, type:

     -> **`set /SP/services/snmp v#=enabled`**

     where # = the SNMP protocol version you want to enable.

     If you enable the `snmp` property for `v3`, you can create an account for authorizaiton and provide read write privileges by typing:

     `create /SP/services/snmp/users/<useraccountname> authenticationpassword=password permission=rw`

     -or-

     If you enable the `snmp` property for `v3`, you can create an account for authorizaiton and provide readonly privileges by typing:

     `create /SP/services/snmp/users/<useraccountname> authenticationpassword=password`

For more information about user accounts and read and write access, see the following section about "Managing SNMP User Accounts and Communities Using the ILOM CLI" on page 15

Managing SNMP User Accounts and Communities Using the ILOM CLI

After ensuring that the proper SNMP properties are enabled in ILOM, you need to configure SNMP user accounts and communities.

Topics

Description	Links
Review requirements prior to perform the procedures in this section	• "Before You Begin" on page 15
Configure user accounts using the CLI	• "SNMP User Account Targets, Properties, and Values" on page 16 • "View and Configure SNMP Community Properties Using the ILOM CLI" on page 17 • "Add an SNMP User Account Using the ILOM CLI" on page 18 • "Edit an SNMP User Account Using the ILOM CLI" on page 19 • "Delete an SNMP User Account Using the ILOM CLI" on page 19 • "Add or Edit an SNMP Community Using the ILOM CLI" on page 19 • "Delete an SNMP Community Using the ILOM CLI" on page 20

Before You Begin

Prior to performing the procedures in this chapter, you must ensure that the following requirements are met:

- To set user account CLI properties in ILOM, you need the User Management (u) role enabled.
- Verify the proper SNMP settings are enabled in ILOM. For more details, see "Enabling SNMP on Managed Server Using the CLI" on page 12.

Note – When working in the ILOM CLI, if the `Sets` parameter is disabled, all SNMP MIB objects are read-only.

- To execute the `snmpset` command, you need to use an SNMP v1 or v2c community or an SNMP v3 user account with read/write (rw) privileges.

Note – The example SNMP commands presented in this section are based on the Net-SNMP sample applications and, therefore, will only work as presented if you have Net-SNMP and the Net-SNMP sample applications installed.

SNMP User Account Targets, Properties, and Values

The SNMP user account targets, properties, and values are accessible under the `/SP/services/snmp` target. The following table identifies the targets, properties, and values that are valid for SNMP user accounts.

TABLE 2-1 SNMP User Acccount Targets, Properties and Values

Target	Property	Value	Default
`/SP/services/snmp/communities/`*communityname*	`permissions`	ro | rw	ro
`/SP/services/snmp/users/`*username*	`authenticationprotocol`	MD5 | SHA	MD5
	`authenticationpassword*`	<*string*>	(null string)
	`permissions`	ro | rw	ro
	`privacyprotocol`	none | DES	none
	`privacypassword+`	<*string*>	(null string)
`/SP/services/snmp`	`engineid = none`	<*string*>	(null string)
	`port = 161`	<*integer*>	161
	`servicestate = enabled`	enable | disabled	enabled
	`sets = enabled`	enabled | disabled	disabled
	`v1 = disabled`	enabled | disabled	disabled
	`v2c = disabled`	enabled | disabled	disabled
	`v3 = disabled`	enabled | disabled	enabled

*> An `authenticationpassword` must be provided when creating or modifying users (SNMP v3 only).

+> If the `privacyprotocol` property has a value other than none, then a `privacypassword` must be set.

For example, to change the `privacyprotocol` for user `a1` to `DES` use the following syntax:

```
-> set /SP/services/snmp/users/a1 privacyprotocol=DES
privacypassword=password authenticationprotocol=SHA
authenticationpassword=password
```

Note that the changes would be invalid if the following syntax was specified:

```
-> set /SP/services/snmp/users/a1 privacyprotocol=DES
```

Note – You can change SNMP user permissions without resetting the `privacy` and `authentication` properties.

▼ View and Configure SNMP Community Properties Using the ILOM CLI

1. **To go to the `/SP/services/snmp` directory, type:**

   ```
   -> cd /SP/services/snmp
   ```

2. **Within that directory, type the `show` command to view SNMP settings. The default settings are as follows:**

```
-> show
   /SP/services/snmp
   Targets:
     communities
     users
   Properties:
     engineid = none
     port = 161
     servicestate = enabled
     sets = disabled
     v1 = disabled
     v2c = disabled
     v3 = enabled
  Commands:
     cd
     set
     show
```

3. **To view the communities, type:**

   -> **show /SP/services/snmp/communities**

```
-> show /SP/services/snmp/communities
/SP/services/snmp/communities
Targets:
  private
  public
Properties:
Commands:
  cd
  create
  delete
  show
```

4. **To create a community with read/write privileges, type:**

   -> **create /SP/services/snmp/communities/***communityname* **permission=rw**

```
-> create /SP/services/snmp/communities/communityname permission=
rw
Created /SP/services/snmp/communities/communityname
```

5. **View the public communities by typing:**

   -> **show /SP/services/snmp/communities/public**

```
-> show /SP/services/snmp/communities/public
/SP/services/snmp/communities/public
Targets:
Properties:
  permission = ro
Commands:
  cd
  set
  show
```

▼ Add an SNMP User Account Using the ILOM CLI

1. **Log in to the ILOM CLI.**

2. **To add an SNMP v3 read-only user account, type the following command:**

   -> **create /SP/services/snmp/users/***username*
   authenticationpassword=*password*

▼ Edit an SNMP User Account Using the ILOM CLI

1. **Log in to the ILOM CLI.**
2. **To edit an SNMP v3 user account, type the following command:**

   -> **set /SP/services/snmp/users/***username* **authenticationpassword=** *password*

Note – When changing the parameters of SNMP users, you must provide a value for `authenticationpassword`, even if you are not changing the password.

▼ Delete an SNMP User Account Using the ILOM CLI

1. **Log in to the ILOM CLI.**
2. **To delete an SNMP v3 user account, type the following command:**

   -> delete /SP/services/snmp/users/*username*

▼ Add or Edit an SNMP Community Using the ILOM CLI

1. **Log in to the ILOM CLI.**
2. **To add an SNMP v1/v2c community, type the following command:**

   -> **create /SP/services/snmp/communities/***communityname*

▼ Delete an SNMP Community Using the ILOM CLI

1. **Log in to the ILOM CLI.**
2. **To delete an SNMP v1/v2c community, type the following command:**

   -> **`delete /SP/services/snmp/communities/`***communityname*

Managing SNMP Traps Using the ILOM CLI

To receive notifications of SNMP traps, you must configure the proper alert rule properties in ILOM. Use the following information in this section to set the proper SNMP alert rules properties in the ILOM CLI.

Topics

Description	Links
Review requirements and special considerations	• "Before You Begin" on page 20
Configure SNMP trap properties	• "Configure SNMP Trap Properties Using the ILOM CLI" on page 21
CLI commond reference for configuring SNMP traps	• "CLI Commands for Managing Alert Rule Configurations" on page 22

Before You Begin

- To create or edit alert rules in ILOM, you need the Admin (a) role enabled.
- When defining an SNMP trap alert for SNMP v3, the SNMP user name must be defined in ILOM as an SNMP user. If the user is not defined in ILOM as an SNMP user, the receiver of the SNMP alert will be unable to decode the SNMP alert message.
- Review the CLI commands for managing alert rule configurationin ILOM. See "CLI Commands for Managing Alert Rule Configurations" on page 22
- For additional information about configuring alert management settings in ILOM, see the chapter about Managing System Alerts in the *Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide* or the *Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide*.

▼ Configure SNMP Trap Properties Using the ILOM CLI

Follow these steps to configure the destinations to which the SNMP traps are sent.

1. **Log in to the ILOM CLI.**
2. **Type the** `show` **command to display the current settings of the alert rule.**

   For example:

```
-> show /SP/alertmgmt/rules/1
/SP/alertmgmt/rules/1
   Targets:
   Properties:
      community_or_username = public
      destination = 0.0.0.0
      level = disable
      snmp_version = 1
      type = snmptrap
   Commands:
     cd
     set
     show
```

3. **To show the** `/SP/alertmgmt/rules` **directory, type these commands:**

   -> **cd /SP/alertmgmt/rules**

   -> **show**

```
-> cd /SP/alertmgmt/rules
-> show
/SP/alertmgmt/rules
  Targets:
    1
    2
    .
    .
    .
    15
  Properties:
        testalert = (Cannot show property)
  Commands:
      cd
      set
      show
```

Choose a rule (from targets 1 through 15) for which you would like to configure a destination for SNMP traps, and go to that directory.

For example:

-> **cd 4**

4. **Within that rule directory, type the** `set` **command to change the rule properties.**

   For example:

   -> **set type=snmptrap level=critical destination=** *IPaddress_of_snmp_management_station* **snmp_version=2c community_or_username=public**

CLI Commands for Managing Alert Rule Configurations

The following table describes the CLI commands that you will need to use to manage alert rule configurations in the ILOM CLI.

TABLE 2-1 CLI Commands for Managing Alert Rule Configurations

CLI Command	Description

`show`	The `show` command enables you to display any level of the alert management command tree by specifying either the full or relative path.
	Examples:
	• To display an alert rule along with its properties using a full path, you would type the following at the command prompt:
	`-> show /SP/alertmgmt/rules/1`
	`/SP/alertmgmt/rules/1`
	`Properties:`
	`community_or_username = public`
	`destination = 129.148.185.52`
	`level = minor`
	`snmp_version = 1`
	`type = snmptrap`
	`Commands:`
	`cd`
	`set`
	`show`
	• To display a single property using the full path, you would type the following at the command prompt:
	`-> show /SP/alertmgmt/rules/1 type`
	`/SP/alertmgmt/rules/1`
	`Properties:`
	`type = snmptrap`
	`Commands:`
	`set`
	`show`
	• To specify a relative path if the current tree location is `/SP/alertmgmt/rules`, you would type the following at the command prompt:
	`-> show 1/`
	`/SP/alertmgmt/rules/1`
	`Targets:`
	`Properties:`
	`community_or_username = public`
	`destination = 129.148.185.52`
	`level = minor`
	`snmp_version = 1`
	`type = snmptrap`
	`Commands:`
	`cd`
	`set`
	`show`

TABLE 2-1 CLI Commands for Managing Alert Rule Configurations *(Continued)*

CLI Command	Description
`cd`	The `cd` command enables you to set the working directory. To set alert management as a working directory on a server SP, you would type the following command at the command prompt: **`-> cd /SP/alertmgmt`**
`set`	The `set` command enables you to set values to properties from any place in the tree. You can specify either a full or relative path for the property depending on the location of the tree. For example: • For full paths, you would type the following at the command prompt: **`-> set /SP/alertmgmt/rules/1 type=snmptrap`** • For relative path (tree location is `/SP/alertmgmt`), you would type the following command path at the command prompt: **`-> set rules/1 type=snmptrap`** • For relative path (tree location is `/SP/alertmgmt/rules/1`), you would type the following command path at the command prompt: **`-> set type=snmptrap`**

Configuring SNMP Settings in ILOM Using the Web Interface

You can choose to configure the SNMP settings in the ILOM CLI or web interface. This section describes the procedures for configuring the SNMP settings using the ILOM web interface.

Topics

Description	Links
Verify and configure the proper ILOM values for enabling SNMP are set on the managed server	• "Enabling SNMP On Managed Server Using the Web Interface" on page 25
Configure SNM user accounts and communities in ILOM	• "Managing SNMP User Accounts and Communities Using the ILOM Web Interface" on page 28
Configure SNMP trap properties in ILOM	• "Managing SNMP Traps Using the ILOM CLI" on page 33

Enabling SNMP On Managed Server Using the Web Interface

Prior to using the `snmpgets` commands described in this guide, you should verify that the correct settings for SNMP are enabled in ILOM. Use the following procedure in this section to view and configure the ILOM CLI SNMP settings on the managed server.

Topics

Description	Links
Review requirements and special considerations	• "Before You Begin" on page 25
Verify and configure the proper SNMP properties in ILOM are set on the managed server	• "Enable SNMP on Managed Server Using the Web Interface" on page 25

Before You Begin

- To set SNMP properties in ILOM you must have the Admin role (a) enabled.
- The SNMP `servericestate` is enabled by default.
- To enable write access when using SNMP, the SNMP Set option must be enabled. This setting is disabled by default.

Note – If the Set option is disabled in ILOM, all SNMP objects are read-only and no snmpset commands will be processed.

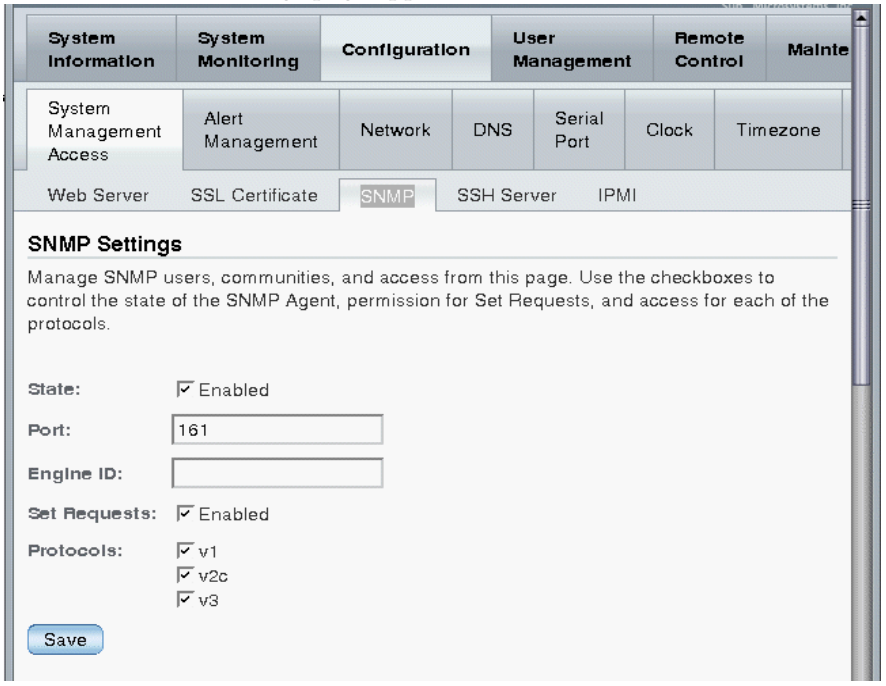
- In ILOM there are three SNMP protocol version properties: `v1`, `v2c`, and `v3`. For `v1` and `v2c`, "communities" are used for authentication. For `v3`, "users" are used for authentication. By default, `v1` and `v2c` are disabled, even though there are default "communities" (public and private) provided. By default, `v3` is enabled, but there are no pre-packaged users.

▼ Enable SNMP on Managed Server Using the Web Interface

1. **Log in to the ILOM web interface.**

2. **Select Configuration --> System Management Access --> SNMP.**

   The SNMP Settings page appears.



3. **Select the State check box to enable the SNMP port.**

   When State is disabled, the SNMP port is blocked, prohibiting all SNMP communication between ILOM and the network.

4. **Type the port number in the Port text field.**

5. **Leave the Engine ID field blank. This allows the default setting to be used.**

   The Engine ID is automatically set by the SNMP agent. While you can use this field to set the Engine ID, we recommend that you leave this field blank. The Engine ID uniquely identifies the SNMP engine and enables users to query the SNMP agent. You should only use this field to set the Engine ID if you are familiar with SNMP v3 security and how this setting is used.

6. **Select or clear the Set Requests check box to enable or disable the Set Requests option.**

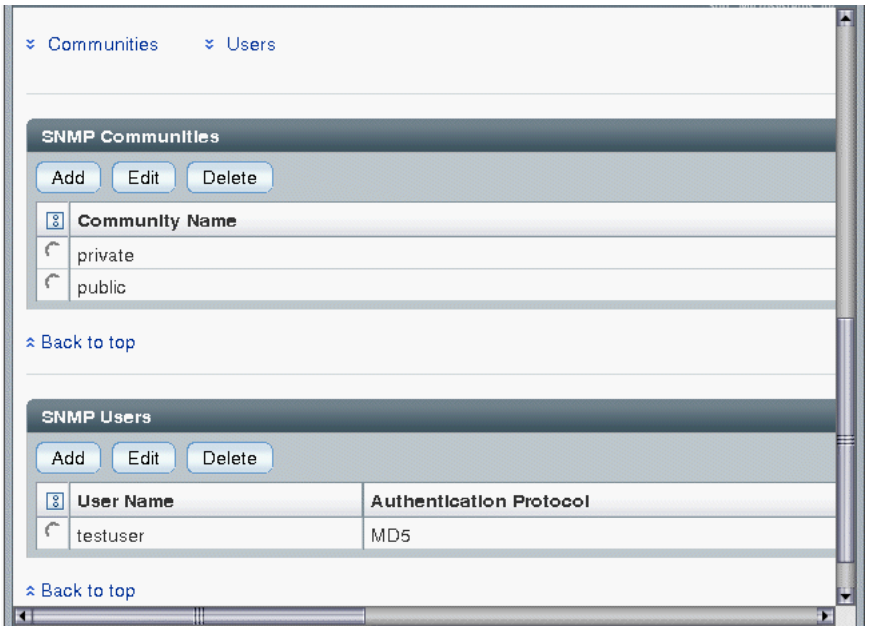
   If the Set Requests option is disabled, all SNMP objects are read-only and no snmpset commands will be processed.

7. **Select a Protocols check box to enable SNMP v1, v2c, or v3.**

   SNMP v3 is enabled by default. You can enable or disable v1, v2c, and v3 protocol versions.

8. **Click Save.**

   At the bottom of the SNMP Settings page, you can also add, edit, or delete SNMP communities or users, as shown in the following screen.



Managing SNMP User Accounts and Communities Using the ILOM Web Interface

After ensuring that the proper SNMP settings are enabled in ILOM, you need to configure SNMP user accounts and communities.

Topics

Description	Links
Configure user accounts using the web interface	• "Add or Edit an SNMP Community Using the Web Interface" on page 29 • "Delete an SNMP Community Using the Web Interface" on page 30 • "Add or Edit an SNMP User Account Using the Web Interface" on page 30 • "Delete an SNMP User Account Using the Web Interface" on page 32 • "Configure SNMP Traps Using the Web Interface" on page 33

Before You Begin

Prior to performing the procedures in this chapter, you must ensure that the following requirements are met:

- To set user account CLI properties in ILOM, you need the User Management (`u`) role enabled.
- Verify that the proper SNMP settings are enabled in ILOM. For more details, see "Enabling SNMP on Managed Server Using the CLI" on page 12.

Note – When working in the ILOM CLI, if the `Sets` parameter is disabled, all SNMP MIB objects are read-only.

- To execute the `snmpset` command, you need to use an SNMP v1 or v2c community or an SNMP v3 user account with read/write (rw) privileges.

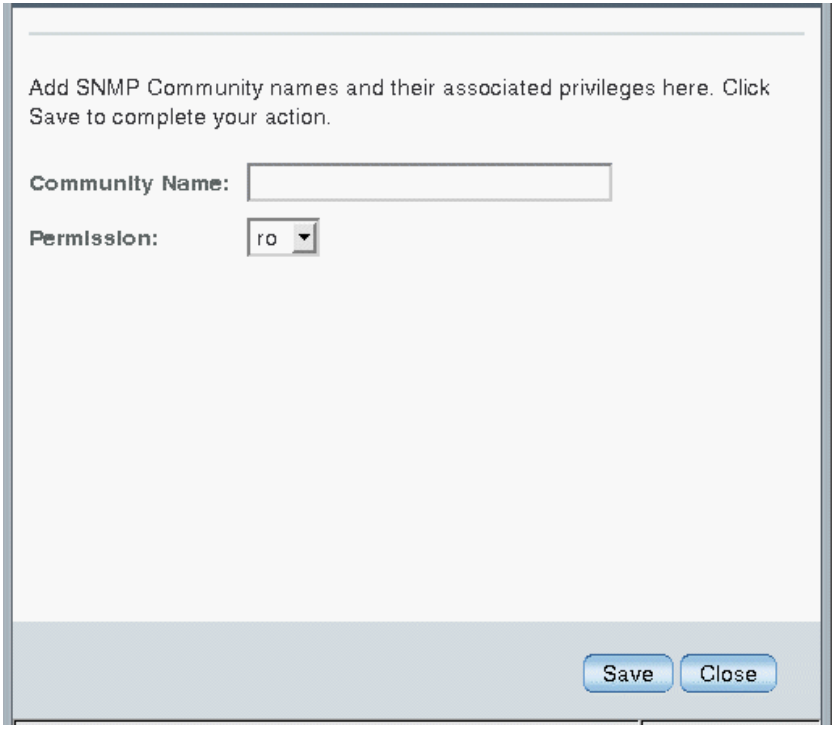
▼ Add or Edit an SNMP Community Using the Web Interface

Follow these steps to add or edit an SNMP v1 or v2c community:

1. **Log in to the ILOM web interface.**
2. **Select Configuration --> System Management Access --> SNMP.**

   Scroll to the bottom half of the SNMP Settings page to find the SNMP Communities dialog.
3. **To add a community, click Add.**

   The SNMP Community dialog box appears.



4. **To edit a community, do the following:**

   a. **Select the appropriate community radio button**

   b. **Click Edit.**

      The SNMP Community dialog box appears.

5. **If you are adding a new community, type the name of the community in the Community Name field; otherwise proceed to the next step.**

   The community name can contain up to 35 characters. It must start with an alphabetic character and cannot contain a space.

6. **Select read-only (ro) or read-write (rw) in the Permissions drop-down list.**

7. **Click Save.**

▼ Delete an SNMP Community Using the Web Interface

Follow these steps to delete an SNMP v1 or v2c community:

1. **Log in to the ILOM web interface.**

2. **Select Configuration --> System Management Access --> SNMP.**

   The SNMP settings page appears.

3. **Click the Communities link or scroll down to the communities list.**

4. **Select the radio button of the SNMP community to delete.**

5. **Click Delete.**

   A confirmation dialog box appears.

6. **Click OK to delete the SNMP community.**

▼ Add or Edit an SNMP User Account Using the Web Interface

Follow these steps to add or edit a SNMP v3 user accounts.

Note – User accounts are not applicable to SNMP v1 and v2c because communities are used to control access.

---

1. **Log in to the ILOM web interface.**

2. **Select Configuration --> System Management Access --> SNMP.**

   The SNMP Settings page appears.

3. **Click the Users link to expand the SNMP Settings page and display SNMP Users.**

4. **To add an SNMP user, click Add.**

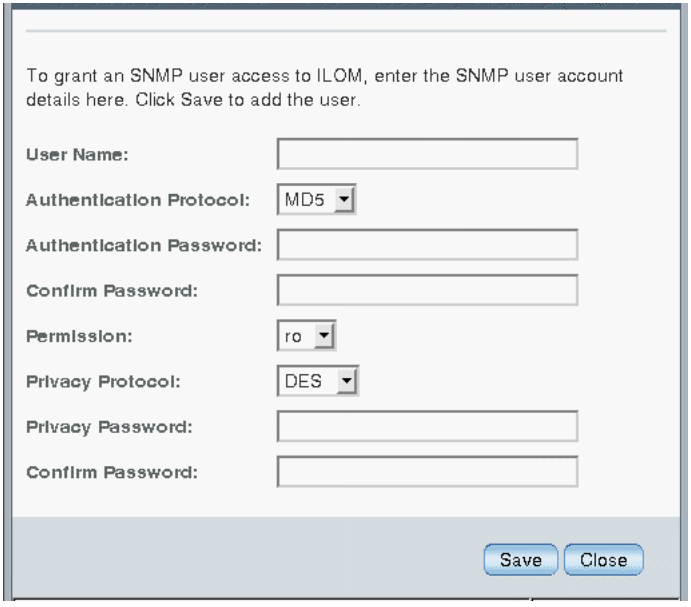
   The Add or Edit SNMP User dialog box appears.

5. **To edit an SNMP user, do the following:**

   a. **Select the appropriate user radio button**

   b. **Click Edit.**

      The Add or Edit SNMP User dialog box appears.



6. **If you are adding a user, type a user name in the User Name text field; otherwise proceed to the next step.**

   The user name can include up to 35 characters. It must start with an alphabetic character and cannot contain spaces.

7. **Select either Message Digest 5 (MD5) or Secure Hash Algorithm (SHA) in the Authentication Protocol drop-down list.**

8. **Type a password in the Authentication Password text field.**

   The authentication password is case-sensitive and must contain 8 to 16 characters, with no colons or space characters.

9. **Retype the authentication password in the Confirm Password text field.**

10. **Select read-only (ro) or read-write (rw) in the Permissions drop-down list.**
11. **Select DES or None in the Privacy Protocol drop-down list.**
12. **Type a password in the Privacy Password text field.**

    The privacy password is case-sensitive and must contain 8 to 16 characters, with no colons or space characters.
13. **Retype the password in the Confirm Password text field.**
14. **Click Save.**

▼ Delete an SNMP User Account Using the Web Interface

Follow these steps to delete an SNMP v3 user account:

1. **Log in to the ILOM web interface.**
2. **Select Configuration --> System Management Access --> SNMP.**

   The SNMP Settings page appears.
3. **Click the Users link or scroll down to the SNMP Users list.**
4. **Select the radio button of the SNMP user account to delete.**
5. **Click Delete under the SNMP User's List.**

   A confirmation dialog box opens.
6. **Click OK to delete the user account.**

Managing SNMP Traps Using the ILOM CLI

To receive notifications of SNMP traps, you must configure the proper alert rule properties in ILOM. Use the following information in this section to set the proper SNMP alert rules properties in the ILOM CLI.

Topics

Description	Links
Review requirements and special considerations	• "Before You Begin" on page 33
Configure SNMP trap settings in ILOM	• "Configure SNMP Traps Using the Web Interface" on page 33

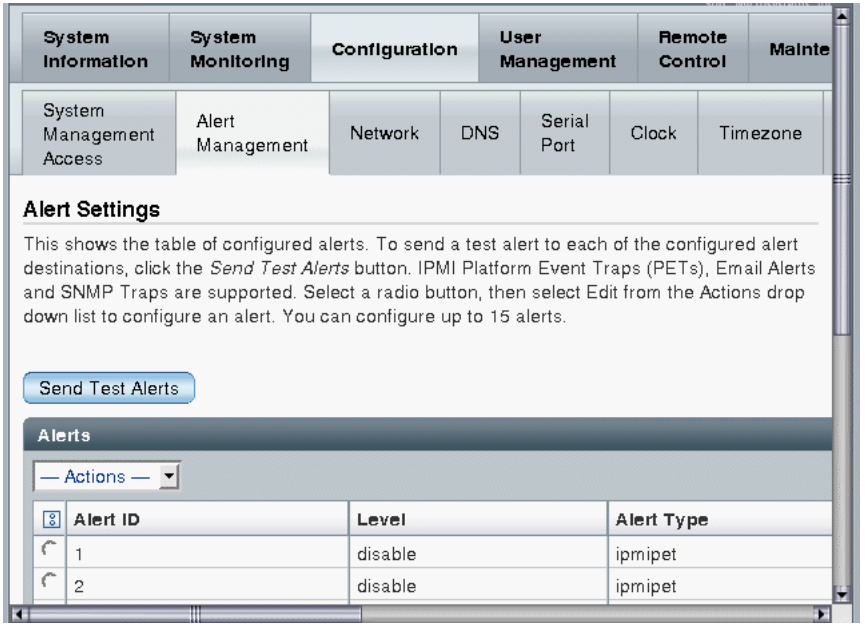
Before You Begin

- To create or edit alert rules in ILOM, you need the Admin (a) role enabled.
- When defining an SNMP trap alert for SNMP v3, the SNMP user name must be defined in ILOM as an SNMP user. If the user is not defined in ILOM as an SNMP user, the receiver of the SNMP alert will be unable to decode the SNMP alert message.
- For additional information about configuring alert management settings in ILOM, see the chapter about Managing System Alerts in the Integrated *Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide* or the *Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide.*

▼ Configure SNMP Traps Using the Web Interface

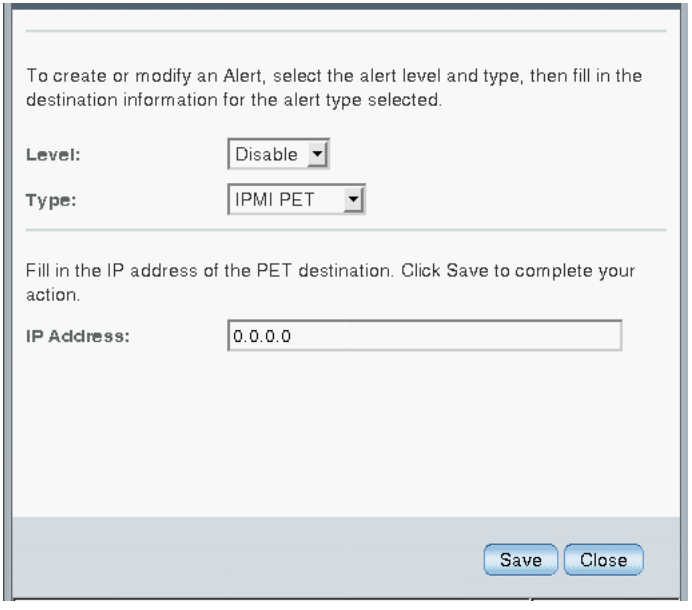
1. **Log in to the ILOM web interface.**
2. **Select Configuration --> Alert Management.**

   The Alert Settings page appears. This page shows a table of the alerts that you can configure. You can configure up to 15 alerts.



3. **To create or modify an alert, select an alert radio button.**
4. **From the Actions drop-down list, select Edit.**

   The Create or Modify Alert dialog appears.



5. **In the Level drop-down list, select the level of the alert.**
6. **In the Type drop-down list, select the alert type.**
7. **In the IP Address field, specify the alert destination IP address.**
8. **Click Save for your changes to take effect.**

CHAPTER 3

Configure ILOM Communication Settings Using SNMP

Topics

Description	Links
Review the prerequisites	• "Before You Begin" on page 38
Configure network settings	• "Assign Host Name and System Identifier" on page 39 • "View and Configure Network Settings" on page 41 • "View and Configure Serial Port Settings" on page 45 • "View and Configure HTTP and HTTPS Settings" on page 48 • "Configure IP Addresses" on page 49
Configure Secure Shell settings	• "View the Current Key and Key Length" on page 53 • "Enable and Disable SSH" on page 54 • "Generate a New SSH Key" on page 55 • "Restart the SSH Server" on page 56

Related Topics

For ILOM	Section	Guide
• Concepts	• ILOM Network Configurations and Log In Requirements	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide* (820-6410)
• CLI	• Configuring ILOM Communication Settings	*Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide* (820-6412)
• Web Interface	• Configuring ILOM Communication Settings	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide* (820-6411)

The ILOM 3.0 Documentation Collection is available at:
`http://docs.sun.com/app/docs/prod/int.lights.mgr30#hic`

Before You Begin

Prior to performing the procedures in this chapter, you must ensure that the following requirements are met.

- To execute the `snmpset` command, you need to use an SNMP v1 or v2c community or SNMP v3 user with read/write (rw) privileges.
- Before you can use the `snmpgets` commands described in this guide, you should verify that the correct properties for SNMP are enabled in ILOM. For more details, see "Enabling SNMP on Managed Server Using the CLI" on page 12.
- Before you can use SNMP to view and configure ILOM settings, you must configure SNMP. For more information, see "Configuring Network Settings" on page 39.
- For examples of SNMP commands, see "SNMP Command Examples" on page 295.

Note – The example SNMP commands presented in this chapter are based on the Net-SNMP sample applications and, therefore, will only work as presented if you have Net-SNMP and the Net-SNMP sample applications installed.

Configuring Network Settings

Topics

Description	Links
Configure network settings	• "Assign Host Name and System Identifier" on page 39 • "View and Configure Network Settings" on page 41 • "View and Configure Serial Port Settings" on page 45 • "View and Configure HTTP and HTTPS Settings" on page 48 • "Configure IP Addresses" on page 49

Before You Begin

- You can use the `get` and `set` commands to view and configure host name and system identifier MIB object settings. For a description of the MIB objects used in this procedure, see "Host Name and System Identifier MIB Objects" on page 40.

This section describes how to configure the network parameters for ILOM using the SNMP interface. If you are using the Net-SNMP sample applications, you can use the `snmpget` and `snmpset` commands to view and configure network settings.

▼ Assign Host Name and System Identifier

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ipaddress*

   `Password:` *password*

2. **To get the host name, type:**

```
% snmpget -v2c -cprivate SNMP_agent_ipaddress ilomCtrlHostName.0
SUN-ILOM-CONTROL-MIB::ilomCtrlHostName.0 = STRING: wgs97-218
```

3. To set the host name, type:

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlHostName.0 s wgs97-200
SUN-ILOM-CONTROL-MIB::ilomCtrlHostName.0 = STRING: wgs97-200
```

4. To get the system identifier, type:

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlSystemIdentifier.0
SUN-ILOM-CONTROL-MIB::ilomCtrlSystemIdentifier.0 = STRING: none
```

5. To set the system identifier, type:

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlSystemIdentifier.0 s wgs97-200
SUN-ILOM-CONTROL-MIB::ilomCtrlSystemIdentifier.0 = STRING: wgs97-
200
```

Host Name and System Identifier MIB Objects

The following MIB objects, values, and types are valid for host name and system identifier.

TABLE 3-1 Valid MIB Objects, Values, and Types for Host Name and System Identifier Settings

MIB Object	Description	Allowed Values	Type	Default
ilomCtrlHostName	The host name for ILOM.	hostname (Size: 0 to 255)	String	None
ilomCtrlSystemIdentifier	The identifier that is sent out on the varbind for all traps that ILOM generated. This string is often the host name of the server that is associated with ILOM.	systemidentifier (Size: 0 to 255)	String	None

▼ View and Configure Network Settings

Note – For a description of the MIB objects used in this procedure, see "Network Settings MIB Objects" on page 44 and the SUN-ILOM-CONTROL-MIB.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   Password**:** *password*

2. **To determine the name of the network target and the current network settings, type:**

```
% snmpwalk -v2c -cprivate -mALL SNMP_agent_ipaddress ilomCtrlNetwork
```

   This command displays the following information:

```
SUN-ILOM-CONTROL-MIB::ilomCtrlNetworkMacAddress."SP/network" = STRING:
00:14:4F:0E:23:B8
SUN-ILOM-CONTROL-MIB::ilomCtrlNetworkIpDiscovery."SP/network" = INTEGER:
static(1)
SUN-ILOM-CONTROL-MIB::ilomCtrlNetworkIpAddress."SP/network" = IpAddress:
ipaddress
SUN-ILOM-CONTROL-MIB::ilomCtrlNetworkIpGateway."SP/network" = IpAddress:
ipaddress
SUN-ILOM-CONTROL-MIB::ilomCtrlNetworkIpNetmask."SP/network" = IpAddress:
ipaddress
SUN-ILOM-CONTROL-MIB::ilomCtrlNetworkPendingIpDiscovery."SP/network" = INTEGER:
static(1)
SUN-ILOM-CONTROL-MIB::ilomCtrlNetworkPendingIpAddress."SP/network" = IpAddress:
ipaddress
SUN-ILOM-CONTROL-MIB::ilomCtrlNetworkPendingIpGateway."SP/network" = IpAddress:
ipaddress
SUN-ILOM-CONTROL-MIB::ilomCtrlNetworkPendingIpNetmask."SP/network" = IpAddress:
ipaddress
SUN-ILOM-CONTROL-MIB::ilomCtrlNetworkCommitPending."SP/network" = INTEGER:
false(2)
```

   The network target name as shown above is "SP/network."

3. **To view the current network IP address for network target named "/SP/network", type:**

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlNetworkIpAddress."/SP/network"
```

4. **To specify a new network IP address, type:**

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlNetworkPendingIpAddress."/SP/network" s 10.300.10.15
```

5. **To put the new network IP address into effect, type:**

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlNetworkCommitPending."/SP/network" i 1
```

6. **Refer to the following SNMP commands for other examples:**
   - To view the MAC address of the out-of-band management interface (where applicable), type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlNetworkOutOfBandMacAddress.0
```

   - To view the MAC address of the sideband management interface (where applicable), type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlNetworkSidebandMacAddress.0
```

   - To view the pending management port for the given target, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlNetworkPendingManagementPort.TARGET_INTERFACE
```

   - To set the pending management port for the given target, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlNetworkPendingManagementPort.TARGET_INTERFACE s
'pendingmanagementport'
```

Note – This property setting does not take effect until the `ilomCtrlNetworkCommitPending` property is set to `true` for the given row.

   - To view the current management port for the given target, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlNetworkgManagementPort.0
```

- To set the current management port for the given target, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlNetworkManagementPort.0 s 'managementport'
```

- To view the address of the DHCP server for this row, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlNetworkDHCPServerAddr.0
```

- To view whether the network state row is enabled, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlNetworkState.0
```

- To set the network state row to enabled, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlNetworkState.0 i 1
```

Network Settings MIB Objects

The following MIB objects, values, and types are valid for network settings.

TABLE 3-2 Valid MIB Objects, Values, and Types for Network Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlNetwork Target`	This is the nomenclature name for a target that has a configurable network. On some systems, there are multiple targets that have networks. On a rackmount stand-alone server, this table will contain only one row for the network configuration of the service processor, which has a nomenclature name of '/SP'. On blade systems, this table will contain multiple rows. There will be a row for each blade's service processor. For example, a blade's service processor nomenclature takes the form of '/CH/BL0/SP', '/CH/BL1/SP' and so on. **Note -** This object is not accessible.	*network_target_name*	String	None
`ilomCtrlNetwork MacAddress`	Indicates the MAC address of the service processor. **Note -** This object is read-only.	*MAC_address*	String	None
`ilomCtrlNetwork IPDiscovery`	Indicates whether the current target is configured to have static IP settings or whether these settings are retrieved dynamically from DHCP. **Note -** This object is read-only.	`Static(1), Dynamic(2)`	Integer	None
`ilomCtrlNetwork IpAddress`	Indicates the current IP address for the given target. **Note -** This object is read-only.	*ipaddress*	String	None
`ilomCtrlNetwork IpGateway`	Indicates the current IP gateway for the given target. **Note -** This object is read-only.	*ip_gateway*	String	None
`ilomCtrlNetwork IpNetmask`	Indicates the current IP netmask for the given target. **Note -** This object is read-only.	*ip_netmask*	String	None

TABLE 3-2 Valid MIB Objects, Values, and Types for Network Settings *(Continued)*

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlNetwork PendingIp Discovery`	This object is used to set the pending value for the mode of IP discovery for the given target. The possible values are `static(1)` or `dynamic(2)`. Static values can be specified by setting the other pending properties in this table: `ilomCtrlNetworkPendingIp-Address`, `ilomCtrlNetworkPendingIp-Gateway`, and `ilomCtrlNetworkPendingIpNetmask`. If dynamic is specified, the other pending properties should not be set. This setting does not take effect until the `ilomCtrlNetworkCommitPending` property is set to `true` for the given row.	`static(1)`, `dynamic(2)`	Integer	None
`ilomCtrlNetwork PendingIp Address`	This object is used to set the pending IP address for the given target. This setting does not take effect until the `ilomCtrlNetworkCommitPending` property is set to `true` for the given row.	*pending_ip_address*	String	None
`ilomCtrlNetwork PendingIp Gateway`	This object is used to set the pending IP gateway for the given target. This setting does not take effect until the `ilomCtrlNetworkCommitPending` object is set to `true` for the given row.	*pending_ip_gateway*	String	None
`ilomCtrlNetwork PendingIp Netmask`	This object is used to set the pending IP netmask for the given target. This setting does not take effect until the `ilomCtrlNetworkCommitPending` object is set to `true` for the given row.	*pending_ip_netmask*	String	None
`ilomCtrlNetwork CommitPending`	This object is used to commit pending settings for the given row. Settings this object to `true(1)` will cause the network to be reconfigured according to the values specified in the other pending settings.	`true(1)`, `false(2)`	Integer	None

▼ View and Configure Serial Port Settings

Note – You can use the `get` and `set` commands to view and configure serial port settings. For a description of the MIB objects used in this procedure, see "Serial Port Settings MIB Objects" on page 46.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   Password**:** *password*

2. **To determine whether the service processor has an internal serial port that is configurable, type:**

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlSerialInternalPortPresent.0
```

3. **To set the baud rate of the internal port to 9600, type:**

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlSerialInternalPortBaudRate.0 i 1
```

Serial Port Settings MIB Objects

The following MIB objects, values, and types are valid for serial port settings.

TABLE 3-3 Valid MIB Objects, Values, and Types for Serial Port Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlSerial Internal PortPresent`	Indicates whether the given device has an internal serial port that is configurable. **Note -** This object is read-only.	`true(1), false(2)`	Integer	None
`ilomCtrlSerial InternalPort BaudRate`	Specifies the current baud rate setting for the internal serial port. This object is only readable or settable if `ilomCtrlSerialInternal-PortPresent` is `true`.	`baud9600(1), baud19200(2), baud38400(3), baud57600(4), baud115200(5)`	Integer	None

TABLE 3-3 Valid MIB Objects, Values, and Types for Serial Port Settings *(Continued)*

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlSerial ExternalPort Present`	Indicates whether the given device has an external serial port that is configurable. **Note -** This object is read-only.	`true(1), false(2)`	Integer	None
`ilomCtrlSerial ExternalPort BaudRate`	Specifies the current baud rate setting for the external serial port. This object is only readable or settable if `ilomCtrlSerialExternalPort-Present` is `true`.	`baud9600(1), baud19200(2), baud38400(3), baud57600(4), baud115200(5)`	Integer	None
`ilomCtrlSerial ExternalPort FlowControl`	Specifies the current flow control setting for the external serial port. This object is only readable or settable if `ilomCtrlSerialExternalPort-Present` is `true`.	`unknown(1), hardware(2), software(3), none(4)`	Integer	None

▼ View and Configure HTTP and HTTPS Settings

Note – You can use the `get` and `set` commands to view and configure HTTP or HTTPS web access. For a description of the MIB objects used in this procedure, see "HTTP and HTTPS Settings MIB Objects" on page 49.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username*`@`*snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP commands for examples:**

   - To get the HTTP state, type:

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlHttpEnabled.0
```

   - To enable HTTP, type:

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlHttpEnabled.0 i 1
```

   - To set the HTTP port number, type:

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlHttpPortNumber.0 i 80
```

   - To configure HTTP to redirect HTTP connections to HTTPS, type:

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlHttpSecureRedirect.0 i 1
```

HTTP and HTTPS Settings MIB Objects

The following MIB objects, values, and types are valid for HTTP and HTTPS settings.

TABLE 3-4 Valid MIB Objects, Values, and Types for HTTP and HTTPS Settings

MIB Object	Description	Allowed Values	Type	Default
HTTP				
`ilomCtrlHttp Enabled`	Specifies whether the embedded web server should be running and listening on the HTTP port.	`true(1), false(2)`	Integer	None
`ilomCtrlHttp PortNumber`	Specifies the port number that the embedded web server should listen on for HTTP requests.	Range: 0..65535	Integer	None
`ilomCtrlHttp SecureRedirect`	Specifies whether the embedded web server should redirect HTTP connections to HTTPS.	`true(1), false(2)`	Integer	Enabled
HTTPS				
`ilomCtrlHttps Enabled`	Specifies whether the embedded web server should be running and listening on the HTTPS port.	`true(1), false(2)`	Integer	True
`ilomCtrlHttps PortNumber`	Specifies the port number that the embedded web server should listen on for HTTPS requests.	Range: 0..65535	Integer	None

▼ Configure IP Addresses

Note – You can use `get` and `set` commands to edit existing IP addresses in ILOM. For a description of the MIB objects used in this procedure, see "Valid MIB Objects for IP Addresses" on page 51.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ipaddress*

   `Password:` *password*

2. **To get a network IP address, type:**

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlNetworkIpAddress.0
```

3. **To set a network IP address, type:**

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlNetworkPendingIpAddress.0 s ipaddress
ilomCtrlNetworkCommitPending.0 i 1
```

Valid MIB Objects for IP Addresses

The following MIB objects, properties, values, and types are valid for IP addresses.

TABLE 3-5 Valid MIB Objects, Properties, Values, and Types for IP Addresses

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlNetworkTarget`	This is the nomenclature name for a target that has a configurable network. On some systems, there are multiple targets that have networks. On a rackmount stand-alone server, this table will contain only one row for the network configuration of the service processor, which has a nomenclature name of '/SP'. On blade systems, this table contains multiple rows. There will be a row for '/SC' which allows for configuration of the network settings. In addition, there are rows for each blade's service processor. For example, a blade's service processor nomenclature takes the form of '/CH/BL0/SP', '/CH/BL1/SP' and so on. This allows for the configuration of the service processors from the CMM. **Note -** This MIB object is not accessible.	*target*	String	none
`ilomCtrlNetworkMacAddress`	The MAC address of the service processor or system controller. **Note -** This object is read-only.	*MAC_address*	String	none
`ilomCtrlNetworkIpDiscovery`	Indicates whether the current target is configured to have static IP settings or whether these settings are retrieved dynamically from DHCP. **Note -** This object is read-only.	`static(1), dynamic(2)`	Integer	none
`ilomCtrlNetworkIpAddress`	Indicates the current IP address for the given target. **Note -** This object is read-only.	*ip_address*	String	none
`ilomCtrlNetworkIpGateway`	Indicates the current IP gateway for the given target. **Note -** This object is read-only.	*ip_gateway*	String	none
`ilomCtrlNetworkIpNetmask`	Indicates the current IP netmask for the given target. **Note -** This object is read-only.	*ip_netmask*	String	none

TABLE 3-5 Valid MIB Objects, Properties, Values, and Types for IP Addresses *(Continued)*

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlNetworkPending IpAddress`	This object is used to set the pending IP address for the given target. This property does not take effect until the `ilomCtrlNetworkCommitPending` property is set to `true` for the given row.	*pending_ipaddress*	String	None
`ilomCtrlNetworkPending IpGateway`	This object is used to set the pending IP gateway for the given target. This setting does not take effect until the `ilomCtrlNetworkCommitPending` property is set to `true` for the given row.	*pending_ip_gateway*	String	None
`ilomCtrlNetworkPending IpDiscovery`	This object is used to set the pending value for the mode of IP discovery for the given target. The possible values are `static(1)` or `dynamic(2)`. Static values can be specified by setting the other pending properties in this table: `ilomCtrlNetworkPendingIp Address`, `ilomCtrlNetworkPendingIp Gateway`, and `ilomCtrlNetworkPendingIp Netmask`. If dynamic is specified, the other pending properties should not be set. This property does not take effect until the `ilomCtrlNetworkCommitPending` MIB object is set to `true` for the given row.	`static(1)`, `dynamic(2)`	Integer	None
`ilomCtrlNetworkPendingIpNetmask`	This object is used to set the pending IP netmask for the given target. This property does not take effect until the `ilomCtrlNetworkCommitPending` property is set to `true` for the given row.	*pending_ip_netmask*	String	none
`ilomCtrlNetworkCommitPending`	This object is used to commit pending properties for the given row. Setting this property to `true(1)` will cause the network to be reconfigured according to the values specified in the other pending properties.	`true(1)`, `false(2)`	Integer	None

Configuring Secure Shell Settings

Topics

Description	Links
Configure Secure Shell settings	• "View the Current Key and Key Length" on page 53 • "Enable and Disable SSH" on page 54 • "Generate a New SSH Key" on page 55 • "Restart the SSH Server" on page 56

▼ View the Current Key and Key Length

Note – You can use `get` commands to view current key and key length information. For a description of the MIB objects used in this procedure, see "RSA and DSA Current Key and Key Length MIB Objects" on page 54.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username*`@`*snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - For RSA keys, to view the current key and key length, type the following:

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlSshRsaKeyFingerprint.0
% snmpget -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlSshRsaKeyLength.0
```

   - For DSA keys, to view the current key and key length, type the following:

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlSshDsaKeyFingerprint.0
% snmpget -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlSshDsaKeyLength.0
```

RSA and DSA Current Key and Key Length MIB Objects

You use the following MIB objects to view key information.

TABLE 3-6 Valid MIB Objects, Values, and Types for the Key Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlSshRsaKey Fingerprint`	The fingerprint of the RSA key used for the SSH protocol.	Size: 0..255	String	None
`ilomCtrlSshRsaKey Length`	The length of the RSA key used for the SSH protocol.	Range: 0..65535	Integer	None
`ilomCtrlSshDsaKey Fingerprint`	The fingerprint of the DSA key used for the SSH protocol.	Size: 0..255	String	None
`ilomCtrlSshDsaKey Length`	The length of the DSA key used for the SSH protocol.	Range: 0..65535	Integer	None

▼ Enable and Disable SSH

Note – You can use the `set` command enable and disable SSH. For a description of the MIB objects used in this procedure, see "SSH Enabled MIB Object" on page 55.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username*`@`*snmp_manager_ipaddress*

   `Password:` *password*

2. **To enable or disable SSH, type the following command to set the** `ilomCtrlSshEnabled` **MIB object to** `1` **(enabled) or** `2` **(disabled):**

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlSshEnabled.0 i 1|2
```

SSH Enabled MIB Object

Use the following MIB object to enable or disable SSH.

TABLE 3-7 Valid MIB Object, Value, and Type for SSH Enabled Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlSsh Enabled`	Specifies whether or not the SSH is enabled.	`true(1), false(2)`	Integer	Enabled

▼ Generate a New SSH Key

Note – You can use the `set` command to generate a new SSH key. For a description of the MIB objects used in this procedure, see "SSH Key MIB Objects" on page 56.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ipaddress*

   `Password:` *password*

2. **To set the SSH key type to RSA, type:**

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlSshGenerateNewKeyType.0 i 2
```

3. **To generate a new RSA key, type:**

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlSshGenerateNewKeyAction.0 i 1
```

Note – The fingerprint and key will look different. The new key will take effect immediately for new connections.

SSH Key MIB Objects

The following MIB objects, values, and types are valid for generating SSH keys.

TABLE 3-8 Valid MIB Objects, Values, and Types for Generating SSH Keys

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlSsh GenerateNewKey Action`	This MIB object is used to initiate a new public key generation.	`true(1), false(2)`	Integer	None
`ilomCtrlSsh GenerateNewKey Type`	This MIB object is used to specify the type of SSH key to generate.	`none(1), rsa(2), dsa(3)`	Integer	None

▼ Restart the SSH Server

Note – For a description of the MIB object used in this procedure, see "Restart SSH MIB Object" on page 56. Restarting SSH will end any existing SSH connections.A new key will not take effect until the SSH server is restarted. You can use the `set` command to restart SSH.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ipaddress*

   `Password:` *password*

2. **To restart the SSH server, type:**

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlSshRestartSshAction.0 i 1
```

Restart SSH MIB Object

The following MIB object, value, and type are valid for restarting SSH.

TABLE 3-9 Valid MIB Object, Value, and Type for Restarting SSH

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlSshRestart SshdAction`	This object is used to initiate an SSHD restart.	`true(1), false(2)`	Integer	None

CHAPTER 4

Manage User Accounts Using SNMP

Topics

Description	Links
Review the prerequisites	• "Before You Begin" on page 60
Configure user accounts	• "Configure User Accounts" on page 61 • "Configure Single Sign On" on page 63
Configure Active Directory settings	• "View and Configure Active Directory Settings" on page 65 • "View and Configure Active Directory Administrator Groups Settings" on page 70 • "View and Configure Active Directory Operator Groups Settings" on page 72 • "View and Configure Active Directory Custom Groups Settings" on page 73 • "View and Configure Active Directory User Domain Settings" on page 76 • "View and Configure Active Directory Alternate Server Settings" on page 77 • "View and Configure Redundancy Settings" on page 81 • "View and Configure Active Directory DNS Locator Settings" on page 82 • "View and Configure DNS Name Server Settings" on page 84

Topics

Description	Links
Configure LDAP settings	• "Configure LDAP Settings" on page 86
Configure LDAP/SSL settings	• "View and Configure LDAP/SSL Administrator Groups Settings" on page 93 • "View and Configure LDAP/SSL Operator Groups Settings" on page 95 • "View and Configure LDAP/SSL Custom Groups Settings" on page 96 • "View and Configure LDAP/SSL User Domain Settings" on page 98 • "View and Configure LDAP/SSL Alternate Server Settings" on page 100
Configure RADIUS settings	• "Configure RADIUS Settings" on page 103

Related Topics

For ILOM	Section	Guide
• Concepts	• User Account Management	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide* (820-6410)
• Web	• Managing User Accounts	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide* (820-6411)
• CLI	• Managing User Accounts	*Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide* (820-6412)

The ILOM 3.0 Documentation Collection is available at:
`http://docs.sun.com/app/docs/prod/int.lights.mgr30#hic`

Before You Begin

Prior to performing the procedures in this chapter, you must ensure that the following requirements are met:

- To view user account information, you need the Read Only (o) role enabled.
- To configure user account information, you need the User Management (u) role enabled.

- To execute the `snmpset` command, you need to use an SNMP v1 or v2c community or an SNMP v3 user account with read/write (rw) privileges.
- For examples of SNMP commands, see "SNMP Command Examples" on page 295.

Note – The example SNMP commands presented in this section are based on the Net-SNMP sample applications and, therefore, will only work as presented if you have Net-SNMP and the Net-SNMP sample applications installed.

Configuring User Accounts

Topics

Description	Links
Configure user accounts	• "Configure User Accounts" on page 61 • "Configure Single Sign On" on page 63

▼ Configure User Accounts

Note – You can use `get` and `set` commands to configure user account MIB object settings. For a description of the MIB objects used in this procedure, see "User Account MIB Objects" on page 62.

---

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ipaddress*

   `Password:` *password*

2. **To create a new user account with a user role of Operator, type:**

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLocalUserRowStatus.'user1' i 4
ilomCtrlLocalUserRoles.'user1' s "operator"
ilomCtrlLocalUserPassword.'user1' s "password"
```

3. To delete a user account, type:

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLocalUserRowStatus.'user1' i 6
```

User Account MIB Objects

The following MIB objects, properties, values, and types are valid for local user accounts.

TABLE 4-1 Valid MIB Objects, Properties, Values, and Types for Local User Accounts

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlLocal UserUsername`	A local user use rname. It must start with an alphabetical letter and may contain alphabetical letters, digits, hyphens and underscores, but cannot contain spaces. It cannot be the same as the password.	*username*	String	None
`ilomCtrlLocal UserPassword`	A local user password.	*password*	String	None

TABLE 4-1 Valid MIB Objects, Properties, Values, and Types for Local User Accounts *(Continued)*

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlLocal UserRoles`	Specifies the role that is associated with a user. The roles can be assigned for the legacy roles of 'Administrator' or 'Operator', or any of the individual role IDs of 'a', 'u', 'c', 'r', 'o' and 's'. The role IDs can be joined together. For example, 'aucros', where a=admin, u=user, c=console, r=reset, o=read-only, s= service.	`administrator`, `operator`, `admin(a)`, `user(u)`, `console(c)`, `reset(r)`, `read-only(o)`, `service(s)`	String	None
ilomCtrlLocal UserRowStatus	This object is used to create a new row or to delete an existing row in the table. This property can be set to either `createAndWait(5)` or `destroy(6)`, to create and remove a user respectively.	`active(1)`, `notInService(2)`, `notReady(3)`, `createAndGo(4)`, `createAndWait(5)`, `destroy(6)`	Integer	None
ilomCtrlLocal UserCLIMode	An enumerated value that describes the possible CLI modes. The `default` mode corresponds to the ILOM DMTF CLP. The `alom` mode corresponds to the ALOM CMT.	`default(1)`, `alom(2)`	Integer	None

▼ Configure Single Sign On

Single Sign On is a convenient authentication service that reduces the number of times you need to enter a password to gain access to ILOM. Single Sign On is enabled by default. As with any authentication service, authentication credentials are passed over the network. If this is not desirable, consider disabling the Single Sign On authentication service.

Note – You can use the `set` command to configure single sign on MIB object settings. For a description of the MIB object used in this procedure, see "Single Sign On MIB Object" on page 64.

---

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   `Password:` *password*

2. **To enable Single Sign On, type:**

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlSingleSignonEnabled.0 i 1
```

Single Sign On MIB Object

The following MIB object, value, and type are valid for Single Sign On.

TABLE 4-2 Valid MIB Object, Value, and Type for Single Sign On

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlSingl SignonEnabled`	Specifies whether Single Sign On (SSO) authentication should be enabled on the device. SSO allows tokens to be passed so that it is not necessary to re-enter passwords between different applications. This allows SSO between the system controller (SC) web interface and the service processor (SP) web interface, between the SC command-line interface and the SP command-line interface, and between the SC and SP interfaces and the Java Remote Console application.	`true(1), false(2)`	Integer	None

Configuring Active Directory Settings

Topics

Description	Links
Configure Active Directory Settings	• "View and Configure Active Directory Settings" on page 65 • "View and Configure Active Directory Administrator Groups Settings" on page 70 • "View and Configure Active Directory Operator Groups Settings" on page 72 • "View and Configure Active Directory Custom Groups Settings" on page 73 • "View and Configure Active Directory User Domain Settings" on page 76 • "View and Configure Active Directory Alternate Server Settings" on page 77 • "View and Configure Active Directory DNS Locator Settings" on page 82

▼ View and Configure Active Directory Settings

Note – You can use the `get` and `set` commands to view and configure Active Directory settings. For a description some of the MIB objects used in this procedure, see "Active Directory MIB Objects" on page 68. For descriptions of the other MIB objects, see the SUN-ILOM-CONTROL-MIB.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To view the Active Directory state, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryEnabled.0
```

- To enable the Active Directory, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryEnabled.0 i 1
```

- To view the Active Directory port number, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryPortNumber.0
```

- To set the Active Directory port number, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryPortNumber.0 i portnumber
```

- To view the Active Directory default user roles, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryDefaultRoles.0
```

- To set the Active Directory default user roles, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryDefaultRoles.0 s acro
```

- To view the Active Directory certificate file URI, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryCertFileURI.0
```

- To set the Active Directory certificate file URI, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryCertFileURI.0 s URI
```

- To view the Active Directory time out, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryTimeout.0
```

- To set the Active Directory time out, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryTimeout.0 i 6
```

- To view the Active Directory certificate validation mode, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryStrictCertEnabled.0
```

- To set the Active Directory certificate validation mode, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryStrictCertEnabled.0 i 1
```

- To view the Active Directory certificate file status, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryCertFileStatus.0
```

- To view the event log setting for the amount of messages sent to the event log, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryLogDetail.0
```

- To configure the event log setting so that only the highest priority messages are sent to the event log, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryLogDetail.0 i 2
```

- To view the role that user1 is to have when authenticated via Active Directory, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryDefaultRoles.'user1'
```

- To specify the Admin (a) role for user1 when authenticated via Active Directory, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryDefaultRoles.'user1' s a
```

- To view and clear the certificate information associated with the server when it is set to true, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryCertClear.0
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryCertClear.0 i 0
```

- To view the version of the certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryCertVersion.0
```

- To view the serial number of the certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryCertserialNo.0
```

- To view the issuer of the certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryCertIssuer.0
```

- To view the subject of the certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryCertSubject.0
```

- To view the valid start date of the certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryCertValidBegin.0
```

- To view the valid end date of the certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirectoryCertValidEnd.0
```

Active Directory MIB Objects

The following MIB objects, values, and types are valid for the Active Directory

TABLE 4-3 Valid MIB Objects, Values, and Types for Active Directory

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlActive Directory Enabled`	Specifies whether the Active Directory client is enabled.	`true(1), false(2)`	Integer	true
`ilomCtrlActive DirectoryIP`	The IP address of the Active Directory server used as a name service for user accounts.	*ipaddress*	String	None
`ilomCtrlActive Directory PortNumbe`	Specifies the port number for the Active Directory client. Specifying zero as the port means auto-select while specifying 1 to 65535 configures the actual port.	portnumber Range: 0 to 65535	Integer	None
`ilomCtrl Active Directory DefaultRoles`	Specifies the role that a user authenticated via Active Directory should have. Setting this property to legacy roles of 'Administrator' or 'Operator', or any of the individual role IDs of 'a', 'u', 'c', 'r', 'o' and 's' will cause the Active Directory client to ignore the schema stored on the Active Directory server. Setting this to 'none' clears the value and indicates that the native Active Directory schema should be used. The role IDs can be joined together. For example, 'aucros,' where a= admin, u=user, c=console, r= reset, o=read-only, and s= service.	`administrator, operator, admin(a), user(u), console(c), reset(r), read-only(o), service(s), none`	String	None
`ilomCtrlActive Directory CertFileURI`	This is the URI of a certificate file needed when Strict Certificate Mode is enabled. Setting the URI causes the transfer of the file, making the certificate available immediately for certificate authentication.	*URI*	String	None

TABLE 4-3 Valid MIB Objects, Values, and Types for Active Directory *(Continued)*

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlActiveDirectoryTimeout`	Specifies the number of seconds to wait before timing out if the Active Directory server is not responding.	Range: 1 to 20 seconds	Integer	4
`ilomCtrlActiveDirectoryStrictCertEnabled`	Specifies whether the Strict Certificate Mode is enabled for the Active Directory client. If enabled, the Active Directory certificate must be uploaded to the SP so that certificate validation can be performed when communicating with the Active Directory server.	`true(1)`, `false(2)`	Integer	true
`ilomCtrlActiveDirectoryCertFileStatus`	A string indicating the status of the certificate file. This is useful in determining whether a certificate file is present or not.	*status*	String	None

▼ View and Configure Active Directory Administrator Groups Settings

Note – If you were using the Net-SNMP sample applications, you could use the `snmpget` and `snmpset` commands to configure the Active Directory Administrator Groups settings. For a description of the MIB objects used in this procedure, see "Active Directory Administrator Groups MIB Objects" on page 71.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username*`@`*snmp_manager_ipaddress*

   `Password:` *password*

2. **To view the name of Active Directory administrator group ID number 2, type:**

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirAdminGroupName.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirAdminGroupName.2 = STRING:
CN=spAdmins,DC=spc,DC=north,DC=sun,DC=com
```

3. **To set the name of Active Directory administrator group ID number 2 to CN=spAdmins,DC=spc,DC=south,DC=sun,DC=com, type:**

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirAdminGroupName.2 s CN=spAdmins,DC=spc,DC=
south,DC=sun,DC=com
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirAdminGroupName.2 = STRING:
CN=spAdmins,DC=spc,DC=south,DC=sun,DC=com
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirAdminGroupName.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirAdminGroupName.2 = STRING:
CN=spAdmins,DC=spc,DC=south,DC=sun,DC=com
```

Active Directory Administrator Groups MIB Objects

The following MIB objects, values, and types are valid for Active Directory Administrator Groups settings.

TABLE 4-4 Valid MIB Objects, Values, and Types for Active Directory Administrator Groups Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlActiveDirAdminGroupId`	An integer identifier of the Active Directory Administrator Groups entry.	1 to 5 **Note -** This object is not accessible for reading or writing.	Integer	None
`ilomCtrlActiveDirAdminGroupName`	This string should contain a Distinguished Name that exactly matches one of the group names on the Active Directory server. Any user belonging to one of these groups in this table will be assigned the ILOM role of Administrator.	*name* (maximum of 255 characters)	String	None

▼ View and Configure Active Directory Operator Groups Settings

---

Note – You can use the `get` and `set` commands to configure the Active Directory Operator Groups settings. For a description of the MIB objects used in this procedure, see "Active Directory Operator Groups MIB Objects" on page 72.

---

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ipaddress*

   `Password:` *password*

2. **To view the name of Active Directory operator group ID number 2, type:**

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirOperatorGroupName.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirOperatorGroupName.2 =
STRING: ad-oper-group-ent-2
```

3. **To set the name of Active Directory operator group ID number 2 to new-name-2, type:**

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirOperatorGroupName.2 s new-name-2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirOperatorGroupName.2 =
STRING: new-name-2
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirOperatorGroupName.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirOperatorGroupName.2 =
STRING: new-name-2
```

Active Directory Operator Groups MIB Objects

The following MIB objects, values, and types are valid Active Directory Operator Groups settings.

TABLE 4-5 Valid MIB Objects, Values, and Types for Active Directory Operator Groups Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlActiveDirOperatorGroupId`	An integer identifier of the Active Directory Operator Groups entry.	1 to 5 **Note -** This object is not accessible for reading or writing.	Integer	None
`ilomCtrlActiveDirOperatorGroupName`	This string should contain a Distinguished Name that exactly matches one of the group names on the Active Directory server. Any user belonging to one of these groups in this table will be assigned the ILOM role of Operator.	*name* (maximum of 255 characters)	String	None

▼ View and Configure Active Directory Custom Groups Settings

Note – You can use the `get` and `set` commands to configure the Active Directory Custom Groups settings. For a description of the MIB objects used in this procedure, see "Active Directory Custom Groups MIB Objects" on page 75.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ipaddress*

   `Password:` *password*

2. **To view the name of Active Directory custom group ID number 2, type:**

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirCustomGroupName.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirCustomGroupName.2 =
STRING: CN=SpSuperCust,OU=Groups,DC=johns,DC=sun,DC=com
```

3. **To set the name of Active Directory custom group ID number 2 to CN= SpSuperCust,OU=Groups,DC=bills,DC=sun,DC=com, type:**

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirCustomGroupName.2 s CN=SpSuperCust,OU=Groups,DC=
bills,DC=sun,DC=com
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirCustomGroupName.2 =
STRING: CN=SpSuperCust,OU=Groups,DC=bills,DC=sun,DC=com
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirCustomGroupName.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirCustomGroupName.2 =
STRING: CN=SpSuperCust,OU=Groups,DC=bills,DC=sun,DC=com
```

4. **To view the roles of Active Directory custom group ID number 2, type:**

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirCustomGroupRoles.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirCustomGroupRoles.2 =
STRING: "aucro"
```

5. **To set the roles of Active Directory custom group ID number 2 to User Management and Read Only (u,o), type:**

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirCustomGroupRoles.2 s "uo"
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirCustomGroupRoles.2 =
STRING: "uo"
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirCustomGroupRole.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirCustomGroupRoles.2 =
STRING: "uo"
```

Active Directory Custom Groups MIB Objects

The following MIB objects, values, and types are valid for Active Directory Custom Groups settings.

TABLE 4-6 Valid MIB Objects, Values, and Types for Active Directory Custom Groups Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlActiveDirCustomGroupId`	An integer identifier of the Active Directory Custom Groups entry.	1 to 5 This object is not accessible for reading or writing.	Integer	None
`ilomCtrlActiveDirCustomGroupName`	This string should contain a Distinguished Name that exactly matches one of the group names on the Active Directory server. Any user belonging to one of these groups in this table will be assigned the ILOM role based on the entry's configuration for roles.	*name* (maximum of 255 characters)	String	None
`ilomCtrlActiveDirCustomGroupRoles`	Specifies the role that a user authenticated via Active Directory should have. Setting this property to legacy roles of 'Administrator' or 'Operator', or any of the individual role IDs of 'a', 'u', 'c', 'r', 'o' and 's' will cause the Active Directory client to ignore the schema stored on the Active Directory server. Setting this object to 'none' clears the value and indicates that the native Active Directory schema should be used. The role IDs can be joined together. For example, 'aucros,' where a= admin, u=user, c=console, r= reset, o=read-only, and s= service.	`administrator`, `operator`, `admin(a)`, `user(u)`, `console(c)`, `reset(r)`, `read-only(o)`, `service(s)`, `none`	String	None

▼ View and Configure Active Directory User Domain Settings

Note – You can use the `get` and `set` commands to configure the Active Directory User Domain settings. For a description of the MIB objects used in this procedure, see "Active Directory User Domain MIB Objects" on page 77.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   `Password:` *password*

2. **To view the name of Active Directory user domain ID number 2, type:**

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirUserDomain.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirUserDomain.2 = STRING:
<USERNAME>@davidc.example.sun.com
```

3. **To set the name of Active Directory user domain ID number 2 to <USERNAME>@johns.example.sun.com, type:**

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirUserDomain.2 s "<USERNAME>@johns.example.sun.com"
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirUserDomain.2 = STRING:
<USERNAME>@johns.example.sun.com
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirUserDomain.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirUserDomain.2 = STRING:
<USERNAME>@johns.example.sun.com
```

Active Directory User Domain MIB Objects

The following MIB objects, values, and types are valid for Active Directory User Domain settings.

TABLE 4-7 Valid MIB Objects, Values, and Types for Active Directory User Domain Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlActiveDirUserDomainId`	An integer identifier of the Active Directory domain.	1 to 5 This object is not accessible for reading or writing.	Integer	None
`ilomCtrlActiveDirUserDomain`	This string should match exactly with an authentication domain on the Active Directory server. This string should contain a substitution string (<USERNAME>), which will be replaced with the user's login name during authentication. Either the principle or Distinguished Name format is allowed.	*name* (maximum of 255 characters)	String	None

▼ View and Configure Active Directory Alternate Server Settings

Note – You can use the `get` and `set` commands to set the values of MIB object properties to configure the Active Directory Alternate Server settings. For a description of the MIB objects used in this procedure, see "Active Directory Alternate Server MIB Objects" on page 80.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username*`@`*snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

- To view the IP address of Active Directory alternate server ID number 2, type:

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerIp.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirAlternateServerIp.2 =
IpAddress: 10.7.143.236
```

- To set the IP address of Active Directory alternate server ID number 2 to 10.7.143.246, type:

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerIp.2 a 10.7.143.246
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirAlternateServerIp.2 =
IpAddress: 10.7.143.246
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerIp.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirAlternateServerIp.2 =
IpAddress: 10.7.143.246
```

- To view the port number of Active Directory alternate server ID number 2, type:

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerPort.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirAlternateServerPort.2 =
INTEGER: 636
```

- To set the port number of Active Directory alternate server ID number 2 to 639, type:

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerPort.2 i 639
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirAlternateServerPort.2 =
INTEGER: 639
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerIp.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirAlternateServerPort.2 =
INTEGER: 639
```

- To view the certificate status of Active Directory alternate server ID number 2, type:

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerCertStatus.2
SUN-ILOM-CONTROL-
MIB::ilomCtrlActiveDirAlternateServerCertStatus.2 = STRING:
certificate not present
```

- To view the certificate URI of Active Directory alternate server ID number 2, type:

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerCertURI.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirAlternateServerCertURI.2 =
STRING: none
```

- To clear the certificate information associated with the server when it is set to true, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerCertClear.0 i 1
```

- To view the certificate version of the alternate server certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerCertVersion.0
```

- To view the serial number of the alternate server certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerCertSerialNo.0
```

- To view the issuer of the alternate server certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerCertIssuer.0
```

- To view the subject of the alternate server certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerCertSubject.0
```

- To view the valid start date of the alternate server certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerCertValidBegin.0
```

- To view the valid end date of the alternate server certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlActiveDirAlternateServerCertValidEnd.0
```

Active Directory Alternate Server MIB Objects

The following MIB objects, values, and types are valid for Active Directory Alternate Server settings.

TABLE 4-8 Valid MIB Objects, Values, and Types for Active Directory Alternate Server Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlActive DirAlternate ServerId`	An integer identifier of the Active Directory alternate server table.	1 to 5 This object is not accessible for reading or writing.	Integer	None
`ilomCtrlActive DirAlternate ServerIP`	The IP address of the Active Directory alternate server used as a name service for user accounts.	*ipaddress*	String	None
`ilomCtrlActive DirAlternate ServerPort`	Specifies the port number for the Active Directory alternate server. Specifying 0 as the port indicates that auto-select will use the well known port number. Specifying 1-65535 is used to explicitly set the port number.	*portnumber* (range: 0 to 65535)	Integer	None
`ilomCtrlActive DirAlternate ServerCert Status`	A string indicating the status of the certificate file. This is useful in determining whether a certificate file is present or not.	*status* (maximum size: 255 characters)	String	None
`ilomCtrlActive DirAlternate ServerCertURI`	This is the URI of a certificate file needed when Strict Certificate Mode is enabled. Setting the URI causes the transfer of the file, making the certificate available immediately for certificate authentication. Additionally, either `remove` or `restore` are supported for direct certificate manipulation.	*URI*	String	None

▼ View and Configure Redundancy Settings

Note – You can use the `get` and `set` commands to view and configure redundancy settings. For a description of the MIB objects used in these commands, see the SUN-ILOM-CONTROL-MIB.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To view the status of the server in a redundant configuration, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlRedundancyStatus.0
```

   - To view the property that controls whether the server is to be promoted or demoted from active or standby status, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlRedundancyAction.0
```

   - To promote a redundant server from standby to active status, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlRedundancyAction.0 i 2
```

   - To view the FRU name of the chassis monitoring module (CMM) on which this agent is running, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlRedundancyFRUName.0
```

▼ View and Configure Active Directory DNS Locator Settings

Note – You can use the `get` and `set` commands to configure the Active Directory DNS Locator settings. For a description of the MIB objects used in this procedure, see "Active Directory DNS Locator MIB Objects" on page 83.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username*`@`*snmp_manager_ipaddress*

   `Password:` *password*

2. **To view the state of Active Directory DNS Locator, type:**

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirDnsLocatorEnabled.0
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirDnsLocatorEnabled.0 =
INTEGER: false(2)
```

3. **To set the state of Active Directory DNS Locator ID number 2 to enabled, type:**

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirDnsLocatorEnabled.0 i 1
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirDnsLocatorEnabled.0 =
INTEGER: true(1)
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirDnsLocatorEnabled.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirDnsLocatorEnabled.2 =
INTEGER: true(1)
```

4. **To view the service name of Active Directory DNS Locator ID number 2, type:**

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirDnsLocatorQueryService.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirDnsLocatorQueryService.2 =
STRING: _ldap._tcp.dc._msdcs.<DOMAIN>.<PORT:636>
```

5. **To set the service name and port number of Active Directory DNS Locator ID number 2, type:**

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirDnsLocatorQueryService.2 s
"_ldap._tcp.pdc._msdcs.<DOMAIN>.<PORT:936>"
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirDnsLocatorQueryService.2 =
STRING: _ldap._tcp.pdc._msdcs.<DOMAIN>.<PORT:936>
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlActiveDirDnsLocatorQueryService.2
SUN-ILOM-CONTROL-MIB::ilomCtrlActiveDirDnsLocatorQueryService.2 =
STRING: _ldap._tcp.pdc._msdcs.<DOMAIN>.<PORT:936>
```

Active Directory DNS Locator MIB Objects

The following MIB objects, values, and types are valid for Active Directory DNS Locator settings.

TABLE 4-9 Valid MIB Objects, Values, and Types for Active Directory DNS Locator Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlActiveDirDnsLocatorEnabled`	Specifies whether or not the Active Directory DNS Locator functionality is enabled.	`true(1), false(2)`	Integer	false
`ilomCtrlActiveDirDnsLocatorQueryId`	An integer identifier of the Active Directory DNS Locator Query entry.	1 to 5 This object is not accessible for reading or writing.	Integer	None
`ilomCtrlActiveDirDnsLocatorQueryService`	The service name that is used to perform the DNS query. The name may contain '<DOMAIN>' as a substitution marker, being replaced by the domain information associated for the user at the time of authentication. The service name may also contain '<PORT:> ', which can be used to override any learned port information, if necessary. For example, <PORT:636> may be specified for the standard LDAP/SSL port 636.	*name* (maximum of 255 characters)	String	None

Configuring DNS Name Server

Topics

Description	Links
Configure DNS Name Server	• "View and Configure DNS Name Server Settings" on page 84

▼ View and Configure DNS Name Server Settings

Note – You can use the `get` and `set` commands to view and configure DNS name server settings. For a description of the MIB objects used in these commands, see the SUN-ILOM-CONTROL-MIB.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To view and specify the name server for DNS, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlDNSNameServers.0
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlDNSNameServers.0 s 'nameservername'
```

   - To view and specify the search path for DNS, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlDNSSearchPath.0
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlDNSSearchPath.0 s 'searchpath'
```

- To view state of DHCP autodns for DNS, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlDNSdhcpAutoDns.0
```

- To set the state of DHCP autodns for DNS to enabled, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlDNSdhcpAutoDns.0 i 1
```

- To view the number of seconds to wait before timing out if the server does not respond, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlDNSTimeout.0
```

- To set the number of seconds to wait before timing out if the server does not respond to 5, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlDNSTimeout.0 i 5
```

- To view the number of times a request is attempted again after a timeout, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlDNSRetries.0
```

- To set the number of times a request is attempted again after a timeout to 5, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlDNSRetries.0 i 5
```

Configuring ILOM for LDAP

Topics

Description	Links
Configure ILOM for LDAP	• "Configure LDAP Settings" on page 86

▼ Configure LDAP Settings

Note – You can use the `get` and `set` commands to configure ILOM for LDAP. For a description of the MIB objects used in this procedure, see "ILOM for LDAP MIB Objects" on page 88.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To view whether the LDAP server is enabled to authenticate LDAP users, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapEnabled.0
```

   - To set the LDAP server state to enabled to authenticate LDAP users, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapEnabled.0 i 1
```

   - To view the LDAP server IP address, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapServerIP.0
```

   - To set the LDAP server IP address, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapServerIP.0 a ipaddress
```

   - To view the LDAP server port number, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapPortNumber.0
```

   - To set the LDAP server port number, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapPortNumber.0 i 389
```

- To view the LDAP server Distinguished Name, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapBindDn.0
```

- To set the LDAP server Distinguished Name, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapBindDn.0 s ou=people,ou=sales,dc=sun,dc=com
```

- To view the LDAP server password, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapBindPassword.0
```

- To set the LDAP server password, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapBindPassword.0 s password
```

- To view the branch of your LDAP server on which user searches are made, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSearchBase.0
```

- To set the branch of your LDAP server on which to search for users, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSearchBase.0 s ldap_server_branch
```

- To view the LDAP server default role, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapDefaultRoles.0
```

- To set the LDAP server default role to Administrator, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapDefaultRoles.0 s administrator
```

ILOM for LDAP MIB Objects

The following MIB objects, values, and types are valid for ILOM for LDAP settings.

TABLE 4-10 Valid MIB Objects, Values, and Types for LDAP Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlLdap Enabled`	Specifies whether the LDAP client is enabled.	`true(1), false(2)`	Integer	false
`ilomCtrlLdap ServerIP`	The IP address of the LDAP server used as a name service for user accounts.	*ipaddress*	String	None
`ilomCtrlLdap PortNumber`	Specifies the port number for the LDAP client.	Range: 0..65535	Integer	389
`ilomCtrlLdap BindDn`	The Distinguished Name (DN) for the read-only proxy user used to bind to the LDAP server. For example: cn=proxyuser,ou= people,dc=sun,dc=com"	*distinguished_name*	String	None
`ilomCtrlLdap BindPassword`	The password of a read-only proxy user which is used to bind to the LDAP server. This property is essentially write-only. The write-only access level is no longer supported as of SNMPv2. This property must return a null value when read.	*password*	String	None
`ilomCtrlLdap SearchBase`	A search base in the LDAP database below which to find users. For example: “ou= people,dc=sun,dc=com"	The branch of your LDAP server on which to search for users	String	None
`ilomCtrlLdap DefaultRoles`	Specifies the role that a user authenticated via LDAP should have. This property supports the legacy roles of ’Administrator’ or ’Operator’, or any of the individual role ID combinations of ’a’, ’u’, ’c’, ’r’, ’o’ and ’s’. For example, ’aucros’, where a=admin, u=user, c= console, r=reset, o=read-only, and s=service.	`administrator, operator, admin(a), user(u), console(c), reset(r), read-only(o), service(s)`	String	None

Configuring ILOM for LDAP/SSL

Topics

Description	Links
Configure LDAP/SSL settings	• "Configure LDAP/SSL Settings" on page 89 • "View and Configure LDAP/SSL Certificate Settings" on page 92 • "View and Configure LDAP/SSL Administrator Groups Settings" on page 93 • "View and Configure LDAP/SSL Operator Groups Settings" on page 95 • "View and Configure LDAP/SSL Custom Groups Settings" on page 96 • "View and Configure LDAP/SSL User Domain Settings" on page 98 • "View and Configure LDAP/SSL Alternate Server Settings" on page 100

▼ Configure LDAP/SSL Settings

Note – You can use the `get` and `set` commands to configure the LDAP/SSL settings. For a description of the MIB objects used in this procedure, see "LDAP/SSL MIB Objects" on page 91.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To set the LDAP/SSL state to Enabled to authenticate LDAP/SSL users, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslEnabled.0 i 1
```

- To set the LDAP/SSL IP address, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslIP.0 a ipaddress
```

- To set the LDAP/SSL port number, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslPortNumber.0 i portnumber
```

- To set the LDAP/SSL default user role, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslDefaultRoles.0 s operator
```

- To set the LDAP/SSL certificate file URI, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslCertFileURI.0 s URI
```

- To set the LDAP/SSL timeout, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslTimeout.0 i 6
```

- To set the LDAP/SSL strict certificate enabled value, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslStrictCertEnabled.0 s true
```

- To set the LDAP/SSL certificate file status, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslCertFileStatus.0 s status
```

- To set the LDAP/SSL log detail value to medium, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslLogDetail.0 i 3
```

LDAP/SSL MIB Objects

The following MIB objects, values, and types are valid for LDAP/SSL settings.

TABLE 4-11 Valid MIB Objects, Values, and Types (Global Variables) for LDAP/SSL Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlLdapSslEnabled`	Specifies whether or not the LDAP/SSL client is enabled.	`true(1)`, `false(2)`	Integer	true
`ilomCtrlLdapSslIP`	The IP address of the LDAP/SSL server used as a directory service for user accounts.	*ipaddress*	String	None
`ilomCtrlLdapSslPort Number`	Specifies the port number for the LDAP/SSL client. Specifying 0 as the port means auto-select while specifying 1-65535 configures the actual port value.	*portnumber* (range: 0 to 65535)	Integer	389
`ilomCtrlLdapSslDefault Roles`	Specifies the role that a user authenticated via LDAP/SSL should have. Setting this property to legacy roles of 'Administrator' or 'Operator', or any of the individual role IDs of 'a', 'u', 'c', 'r', 'o' and 's' will cause the LDAP/SSL client to ignore the schema stored on the LDAP server. Setting this object to 'none' clears the value and indicates that the native LDAP/SSL schema should be used. The individual role IDs can be joined together in any combination of two or more roles. For example, this object can be set to 'aucros', where a=admin, u=user, c=console, r=reset, o= read-only, and s=service.	`administrator`, `operator`, `admin(a)`, `user(u)`, `console(c)`, `reset(r)`, `read-only(o)`, `service(s)`, `none`	String	None
`ilomCtrlLdapSslCertFile URI`	The TFTP URI of the LDAP/SSL server's certificate file that should be uploaded in order to perform certificate validation. Setting the URI causes the transfer of the specified file, making the certificate available immediately for certificate authentication. The server certificate file is needed when Strict Certificate Mode is enabled. Additionally, either `remove` or `restore` are supported for direct certificate manipulation.	*URI*	String	None
`ilomCtrlLdapSslTimeout`	Specifies the number of seconds to wait before timing out if the LDAP/SSL server is not responding.	Range: 1 to 20	Integer	4

TABLE 4-11 Valid MIB Objects, Values, and Types (Global Variables) for LDAP/SSL Settings *(Continued)*

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlLdapSslStrictCertEnabled`	Specifies whether or not the Strict Certificate Mode is enabled for the LDAP/SSL Client. If enabled, the LDAP/SSL server's certificate must be uploaded to the SP so that certificate validation can be performed when communicating with the LDAP/SSL server.	`true(1)`, `false(2)`	Integer	true
`ilomCtrlLdapSslCertFileStatus`	A string indicating the status of the certificate file. This is useful in determining whether a certificate file is present or not.	*status* (maximum size: 255 characters)	String	None
`ilomCtrlLdapSslLogDetail`	Controls the amount of messages sent to the event log. The high priority has the least number of messages going to the log, while the lowest priority 'trace' has the most messages logged. When this object is set to `none`, no messages are logged.	`none(1)`, `high(2)`, `medium(3)`, `low(4)`, `trace(5)`	Integer	None

▼ View and Configure LDAP/SSL Certificate Settings

Note – You can use the `get` and `set` commands to view and configure LDAP/SSL certificate settings. For a description of the MIB objects used in these commands, see the SUN-ILOM-CONTROL-MIB.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To clear the certificate information associated with the server when it is set to true, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslCertFileClear.0 i 0
```

   - To view the certificate version of the certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslCertFileVersion.0
```

- To view the serial number of the certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslCertFileSerialNo.0
```

- To view the issuer of the certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslCertFileIssuer.0
```

- To view the subject of the certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslCertFileSubject.0
```

- To view the valid start date of the certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslCertFileValidBegin.0
```

- To view the valid end date of the certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslCertFileValidEnd.0
```

▼ View and Configure LDAP/SSL Administrator Groups Settings

Note – You can use the `get` and `set` commands to configure the LDAP/SSL Administrator Groups settings. For a description of the MIB objects used in this procedure, see "LDAP/SSL Administrator Groups MIB Objects" on page 94.

---

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

- To view the name of LDAP/SSL administrator group ID number 3, type:

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslAdminGroupName.3
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslAdminGroupName.3 = STRING:
CN=SpSuperAdmin,OU=Groups,DC=davidc,DC=example,DC=sun,DC=com
```

- To set the name of LDAP/SSL administrator group ID number 3 to CN= SpSuperAdmin,OU=Groups,DC=tomp,DC=example,DC=sun,DC=com, type:

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslAdminGroupName.3 s CN=SpSuperAdmin,OU=
Groups,DC=tomp,DC=example,DC=sun,DC=com
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslAdminGroupName.3 = STRING:
CN=SpSuperAdmin,OU=Groups,DC=tomp,DC=example,DC=sun,DC=com
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslAdminGroupName.3
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslAdminGroupName.3 = STRING:
CN=SpSuperAdmin,OU=Groups,DC=tomp,DC=example,DC=sun,DC=com
```

LDAP/SSL Administrator Groups MIB Objects

The following MIB objects, values, and types are valid for LDAP/SSL Administrator Groups settings.

TABLE 4-12 Valid MIB Objects, Values, and Types for LDAP/SSL Administrator Groups Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlLdapSslAdminGroupId`	An integer identifier of the LDAP/SSL AdminGroup entry.	1 to 5 **Note -** This object is not accessible for reading or writing.	Integer	None
`ilomCtrlLdapSslAdminGroupName`	This string should contain a Distinguished Name that exactly matches one of the group names on the LDAP/SSL server. Any user belonging to one of these groups in this table will be assigned the ILOM role of Administrator.	*name* (maximum of 255 characters)	String	None

▼ View and Configure LDAP/SSL Operator Groups Settings

Note – You can use the `get` and `set` commands to configure the LDAP/SSL Operator Groups settings. For a description of the MIB objects used in this procedure, see "LDAP/SSL Operator Groups MIB Objects" on page 96.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To view the name of LDAP/SSL operator group ID number 3, type:

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslOperatorGroupName.3
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslOperatorGroupName.3 =
STRING: CN=SpSuperOper,OU=Groups,DC=davidc,DC=example,DC=
sun,DC=com
```

   - To set the name of Active Directory operator group ID number 3 to CN=SpSuperAdmin,OU=Groups,DC=tomp,DC=example,DC=sun,DC=com, type:

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslOperatorGroupName.3 s CN=SpSuperOper,OU=
Groups,DC=tomp,DC=example,DC=sun,DC=com
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslOperatorGroupName.3 =
STRING: CN=SpSuperOper,OU=Groups,DC=tomp,DC=example,DC=sun,DC=
com
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslOperatorGroupName.3
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslOperatorGroupName.3 =
STRING: CN=SpSuperOper,OU=Groups,DC=tomp,DC=example,DC=sun,DC=
com
```

LDAP/SSL Operator Groups MIB Objects

The following MIB objects, values, and types are valid for LDAP/SSL Operator Groups settings.

TABLE 4-13 Valid MIB Objects, Values, and Types for LDAP/SSL Operator Groups Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlLdapSslOperatorGroupId`	An integer identifier of the LDAP/SSL Operator Group entry.	1 to 5 **Note -** This object is not accessible for reading or writing.	Integer	None
`ilomCtrlLdapSslOperatorGroupName`	This string should contain a Distinguished Name that exactly matches one of the group names on the LDAP/SSL server. Any user belonging to one of these groups in this table will be assigned the ILOM role of Operator.	*name* (maximum of 255 characters)	String	None

▼ View and Configure LDAP/SSL Custom Groups Settings

Note – You can use the `get` and `set` commands to configure the LDAP/SSL Custom Groups settings. For a description of the MIB objects used in this procedure, see "LDAP/SSL Custom Groups MIB Objects" on page 98.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username*`@`*snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**
   - To view the name of LDAP/SSL custom group ID number 2, type:

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslCustomGroupName.2
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslCustomGroupName.2 = STRING:
CN=SpSuperCust,OU=Groups,DC=johns,DC=sun,DC=com
```

- To set the name of LDAP/SSL custom group ID number 2 to CN= SpSuperCust,OU=Groups,DC=bills,DC=sun,DC=com, type:

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslCustomGroupName.2 s CN=SpSuperCust,OU=Groups,DC=
bills,DC=sun,DC=com
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslCustomGroupName.2 = STRING:
CN=SpSuperCust,OU=Groups,DC=bills,DC=sun,DC=com
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslCustomGroupName.2
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslCustomGroupName.2 = STRING:
CN=SpSuperCust,OU=Groups,DC=bills,DC=sun,DC=com
```

- To view the roles of LDAP/SSL custom group ID number 2, type:

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslCustomGroupRoles.2
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslCustomGroupRoles.2 = STRING:
"aucro"
```

- To set the roles of LDAP/SSL custom group ID number 2 to User Management and Read Only (u,o), type:

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslCustomGroupRoles.2 s "uo"
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslCustomGroupRoles.2 = STRING:
"uo"
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslCustomGroupRoles.2
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslCustomGroupRoles.2 = STRING:
"uo"
```

LDAP/SSL Custom Groups MIB Objects

The following MIB objects, values, and types are valid LDAP/SSL Custom Groups settings.

TABLE 4-14 Valid MIB Objects, Values, and Types for LDAP/SSL Custom Groups Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlLdapSslCustomGroupId`	An integer identifier of the LDAP/SSL custom group entry.	1 to 5 **Note -** This object is not accessible for reading or writing.	Integer	None
`ilomCtrlLdapSslCustomGroupName`	This string should contain a Distinguished Name that exactly matches one of the group names on the LDAP/SSL server. Any user belonging to one of these groups in this table will be assigned the ILOM role based on the entry's configuration for roles.	*name* (maximum of 255 characters)	String	None
`ilomCtrlLdapSslCustomGroupRoles`	Specifies the role that a user authenticated via LDAP/SSL should have. Setting this property to legacy roles of 'Administrator' or 'Operator', or any of the individual role IDs of 'a', 'u', 'c', 'r', 'o' and 's' will cause the LDAP/SSL client to ignore the schema stored on the LDAP/SSL server. Setting this object to 'none' clears the value and indicates that the native LDAP/SSL schema should be used. The role IDs can be joined together. For example, 'aucros,' where a=admin, u=user, c=console, r=reset, o=read-only, and s=service.	`administrator`, `operator`, `admin(a)`, `user(u)`, `console(c)`, `reset(r)`, `read-only(o)`, `service(s)`, `none`	String	None

▼ View and Configure LDAP/SSL User Domain Settings

Note – You can use the `get` and `set` commands to configure the LDAP/SSL User Domain settings. For a description of the MIB objects used in this procedure, see "LDAP/SSL User Domain MIB Objects" on page 99.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username*`@`*snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To view the name of LDAP/SSL user domain ID number 3, type:

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslUserDomain.3
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslUserDomain.3 = STRING: CN=
<USERNAME>,CN=Users,DC=davidc,DC=example,DC=sun,DC=com
```

   - To set the name of LDAP/SSL user domain ID number 3 to CN= <USERNAME>, CN=Users,DC=tomp,DC=example,DC=sun,DC=com, type:

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslUserDomain.3 s CN=<USERNAME>,CN=Users,DC=
tomp,DC=example,DC=sun,DC=com
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslUserDomain.3 = STRING: CN=
<USERNAME>,CN=Users,DC=tomp,DC=example,DC=sun,DC=com
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslUserDomain.3
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslUserDomain.3 = STRING: CN=
<USERNAME>,CN=Users,DC=tomp,DC=example,DC=sun,DC=com
```

LDAP/SSL User Domain MIB Objects

The following MIB objects, values, and types are valid for LDAP/SSL User Domain settings.

TABLE 4-15 Valid MIB Objects, Values, and Types for LDAP/SSL User Domain Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlLdapSslUserDomainId`	An integer identifier of the LDAP/SSL domain.	1 to 5 **Note -** This object is not accessible for reading or writing.	Integer	None
`ilomCtrlLdapSslUserDomain`	This string should match exactly with an authentication domain on the LDAP/SSL server. This string should contain a substitution string (<USERNAME>), which will be replaced with the user's login name during authentication. Either the principle or Distinguished Name format is allowed.	*name* (maximum of 255 characters)	String	None

▼ View and Configure LDAP/SSL Alternate Server Settings

Note – You can use the `get` and `set` commands to configure the LDAP/SSL Alternate Server settings. For a description of the MIB objects used in this procedure, see "LDAP/SSL Alternate Server MIB Objects" on page 102 and the SUN-ILOM-CONTROL MIB.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To view the IP address of LDAP/SSL alternate server ID number 3, type:

```
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslAlternateServerIp.3
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslAlternateServerIp.3 =
IpAddress: 10.7.143.236
```

   - To set the IP address of LDAP/SSL alternate server ID number 3 to 10.7.143.246, type:

```
% snmpset -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslAlternateServerIp.3 a 10.7.143.246
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslAlternateServerIp.3 =
IpAddress: 10.7.143.246
% snmpget -v1 -cprivate -mALL SNMP_agent_ipaddress
ilomCtrlLdapSslAlternateServerIp.3
SUN-ILOM-CONTROL-MIB::ilomCtrlLdapSslAlternateServerIp.3 =
IpAddress: 10.7.143.246
```

   - To view and clear the certificate information associated with the alternate server when it is set to true, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslAlternateServerCertClear.0
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslAlternateServerCertClear.0 i 0
```

- To view the alternate server certificate version of the certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslAlternateServerCertVersion.0
```

- To view the serial number of the alternate server certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslAlternateServerCertSerialNo.0
```

- To view the issuer of the alternate server certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslAlternateServerCertIssuer.0
```

- To view the subject of the alternate server certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslAlternateServerCertSubject.0
```

- To view the valid start date of the alternate server certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslAlternateServerCertValidBegin.0
```

- To view the valid end date of the alternate server certificate file, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlLdapSslAlternateServerCertValidEnd.0
```

LDAP/SSL Alternate Server MIB Objects

The following MIB objects, values, and types are valid for LDAP/SSL Alternate Server settings.

TABLE 4-16 Valid MIB Objects, Values, and Types for LDAP/SSL Alternate Server Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlLdapSslAlternateServerId`	An integer identifier of the LDAP/SSL alternate server table.	1 to 5 **Note -** This object is not accessible for reading or writing.	Integer	None
`ilomCtrlLdapSslAlternateServerIP`	The IP address of the LDAP/SSL alternate server used as directory server for user accounts.	*ipaddress*	String	None
`ilomCtrlLdapSslAlternateServerPort`	Specifies the port number for the LDAP/SSL alternate server. Specifying zero as the port indicates that auto-select will use the well known port number. Specifying 1-65535 is used to explicitly set the port number.	*portnumber* (range: 0 to 65535)	Integer	None
`ilomCtrlLdapSslAlternateServerCertStatus`	A string indicating the status of the certificate file. This is useful in determining whether a certificate file is present or not.	*status* (maximum size: 255 characters)	Sting	None
`ilomCtrlLdapSslAlternateServerCertURI`	This is the URI of a certificate file needed when Strict Certificate Mode is enabled. Setting the URI causes the transfer of the file, making the certificate available immediately for certificate authentication. Additionally, either `remove` or `restore` are supported for direct certificate manipulation.	*URI*	String	None

Configuring RADIUS Settings

Topics

Description	Links
Configure ILOM for LDAP	• "Configure RADIUS Settings" on page 103

▼ Configure RADIUS Settings

Note – Before completing this procedure, collect the appropriate information about your RADIUS environment. You can use the `get` and `set` commands to configure RADIUS. For a description of the MIB objects used in this procedure, see "RADIUS MIB Objects" on page 104.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To view whether the RADIUS server is enabled to authenticate RADIUS users, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlRadiusEnabled.0
```

   - To set the RADIUS server state to Enabled to authenticate RADIUS users, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlRadiusEnabled.0 i 1
```

   - To view the RADIUS server IP address, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlRadiusServerIP.0
```

   - To set the RADIUS server IP address, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlRadiusServerIP.0 a ipaddress
```

   - To view the RADIUS server port number, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlRadiusPortNumber.0
```

   - To set the RADIUS server port number, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlRadiusPortNumber.0 i portnumber
```

- To view the RADIUS server shared secret, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlRadiusSecret.0
```

- To set the RADIUS server shared secret, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlRadiusSecret.0 s secret
```

- To view the RADIUS server default user roles, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlRadiusDefaultRoles.0
```

- To set the RADIUS server default user roles to console, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlRadiusDefaultRoles.0 s c
```

RADIUS MIB Objects

The following MIB objects, values, and types are valid for RADIUS settings.

TABLE 4-17 Valid MIB Objects, Values, and Types for RADIUS Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlRadiusEnabled`	Specifies whether or not the RADIUS client is enabled.	`true(1)`, `false(2)`	Integer	false
`ilomCtrlRadiusServerIP`	The IP address of the RADIUS server used as a name service for user accounts.	*ipaddress*	String	None

TABLE 4-17 Valid MIB Objects, Values, and Types for RADIUS Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlRadius PortNumber`	Specifies the port number for the RADIUS client.	*portnumber* (range: 0 to 65535)	Integer	1812
`ilomCtrlRadius Secret`	The shared secret encryption key that is used to encypt traffic between the RADIUS client and server.	*secret* (maximum length: 255 characters)	Sting	None
`ilomCtrlRadius DefaultRoles`	Specifies the role that a user authenticated via RADIUS should have. This property supports the legacy roles of 'Administrator' or 'Operator', or any of the individual role ID combinations of 'a', 'u', 'c', 'r', 'o' and 's'. For example, 'aucro', where a=admin, u=user, c=console, r=reset, o=read-only, and s=service.	`administrator`, `operator`, `admin(a)`, `user(u)`, `console(c)`, `reset(r)`, `read-only(o)`, `service(s)`	String	None

CHAPTER 5

Monitor Component Information and Manage Inventory Using SNMP

Topics

Description	Links
Review the prerequisites	• "Before You Begin" on page 108
View component information and manage inventory	• "View Component Information" on page 109 • "View and Set Clock Settings" on page 111 • "View and Clear the ILOM Event Log" on page 112 • "Configure Remote Syslog Receiver IP Addresses" on page 114
Manage alert rules	• "Configure an Alert Rule" on page 115
Configure SMTP client for Email notification alerts	• "Configure SMTP Client for Email Notification Alerts" on page 118
Configure alerts	• "View and Configure Email Alert Settings" on page 119
Configure Telemetry Harness Daemon	• "View and Configure Telemetry Harness Daemon Settings" on page 121

Related Topics

For ILOM	Section	Guide
• Concepts	• System Monitoring and Alert Management	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide* (820-6410)
• CLI	• Managing Alerts	*Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide* (820-6412)
• Web	• Managing Alerts	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide* (820-6411)

The ILOM 3.0 Documentation Collection is available at:
`http://docs.sun.com/app/docs/prod/int.lights.mgr30#hic`

Before You Begin

- Before you can use SNMP to view and configure ILOM settings, you must configure SNMP. For more information, see "Preparing Your System to Use SNMP" on page 5.
- When executing the `snmpset` command, you need to use a v1/v2c community or a v3 user with read/write (rw) privileges.
- For examples of SNMP commands, see "SNMP Command Examples" on page 295.

Note – The example SNMP commands presented in this section are based on the Net-SNMP sample applications and, therefore, will only work as presented if you have Net-SNMP and the Net-SNMP sample applications installed.

Viewing Component Information

Topics

Description	Links
View the component information	• "View Component Information" on page 109 • "Component MIB Objects" on page 110

▼ View Component Information

Note – You can use `get` commands to view component information. For a description of the MIB objects used in this procedure, see "Component MIB Objects" on page 110.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ip_address*

   `Password:` *password*

2. **To view the firmware revision, type:**

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ip_address
entPhysicalFirmwareRev.1
```

Component MIB Objects

TABLE 5-1 lists several of the MIB objects provided by the ENTITY-MIB that you can use to view components.

TABLE 5-1 MIB Objects, Values, and Types for Component Settings

MIB Object	Description	Values	Type	Default
`entPhysicalName`	The textual name of the physical entity.	Size: 0..255	String	Zero-length string
`entPhysicalDescr`	A textual description of physical entity.	Size: 0..255	String	None
`entPhysicalContainedIn`	The value of entPhysicalIndex for the physical entity that *contains* this physical entity. A value of zero indicates this physical entity is not contained in any other physical entity.	Range: 0..2147483647	Integer	None
`entPhysicalClass`	An indication of the general hardware type of the physical entity.	`other(1)`, `unknown(2)`, `chassis(3)`, `backplane(4)`, `container(5)`, `powerSupply(6)`, `fan(7)`, `sensor(8)`, `module(9)`, `port(10)`, `stack(11)`	Integer	None
`entPhysicalFirmwareRev`	The vendor-specific firmware revision string for the physical entity.	Size: 0..255	String	Zero-length string

Monitoring System Sensors, Indicators, and ILOM Event Log

Topics

Description	Links
View and set clock settings	• "View and Set Clock Settings" on page 111
View and clear the ILOM event log	• "View and Clear the ILOM Event Log" on page 112
Configure remote syslog receiver IP addresses	• "Configure Remote Syslog Receiver IP Addresses" on page 114
Configure alert rules	• "Configure an Alert Rule" on page 115

▼ View and Set Clock Settings

Note – You can use the `get` and `set` commands to view and set clock settings with respect to Network Time protocol (NTP) synchronization. For a description of the MIB objects used in this procedure, see "ILOM Clock Setting MIB Objects" on page 112.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ip_address*

   `Password:` *password*

2. **Refer to the following SNMP commands for examples:**

   - To view the NTP server state, type:

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlNTPEnabled.0
```

   - To set the NTP server state to enabled, type:

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlNTPEnabled.0 i 1
```

- To view the date and time of the device, type:

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlDateAndTime.0
```

- To set the date and time of the device, type:

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlDateAndTime.0 s 2008-3-24,4:59:47.0
```

ILOM Clock Setting MIB Objects

The following MIB objects, values, and types are valid for ILOM clock settings.

TABLE 5-2 Valid MIB Objects, Values, and Types for ILOM Clock Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlDateAndTime`	The date and time of the device.	*date/time*	String	None
`ilomCtrlNTPEnabled`	Specifies whether the Network Time Protocol is enabled.	`true(1)`, `false(2)`	Integer	false
ilomCtrlTimezone	The configured timezone string.	Size: 0..255	String	None

▼ View and Clear the ILOM Event Log

Note – You can use the `get` command to view the ILOM event log and the `set` command to configure the ILOM event log. For a description of the MIB objects used in this procedure, see "ILOM Event Log MIB Objects" on page 113.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username*`@`*snmp_manager_ip_address*

   `Password:` *password*

2. **To view the ILOM event log type for an event log with a record ID of 2, type:**

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlEventLogType.2
```

3. To clear the ILOM event log, type:

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlEventLogClear.0 i 1
```

ILOM Event Log MIB Objects

The following MIB objects, values, and types are valid for ILOM event log settings.

TABLE 5-3 MIB Objects, Values, and Types for Event Log Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlEventLog RecordID`	The record number for a given event log entry. **Note -** This object is not accessible.	Range: 1..10000	Integer	None
`ilomCtrlEventLog Type`	An integer representing the type of event. **Note -** This object is read-only.	`log(1), action2), fault(3), state(4), repair(5)`	Integer	None
`ilomCtrlEventLog Timestamp`	The date and time that the event log entry was recorded. **Note -** This object is read-only.	*date/time*	String	None
`ilomCtrlEventLog Class`	An integer representing the class of event. **Note -** This object is read-only.	`audit(1), ipmi(2), chassis(3), fma(4), system(5) pcm(6)`	Integer	None

TABLE 5-3 MIB Objects, Values, and Types for Event Log Settings *(Continued) (Continued)*

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlEventLog Severity`	The event severity corresponding to the given log entry. **Note -** This object is read-only.	`disable(1), critical(2), major(3), minor(4), down(5`	Integer	None
`ilomCtrlEventLog Description`	A textual description of the event. **Note -** This object is read-only.	*description*	String	None
`ilomCtrlEventLog Clear`	Setting this object to true clears the event log.	`true(1), false(2)`	Integer	None

▼ Configure Remote Syslog Receiver IP Addresses

Note – You can use the `get` and `set` commands to view and set IP addresses for a remote Syslog receiver. For a description of the MIB objects used in this procedure, see "Remote Syslog Receiver IP Addresses MIB Objects" on page 115.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username@snmp_manager_ip_address*

   `Password:` *password*

2. **To view a remote syslog destination IP address, type:**

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlRemoteSyslogDest1.0
```

3. **To set a remote syslog destination IP address, type:**

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlRemoteSyslogDest1.0 s ip_address
```

Remote Syslog Receiver IP Addresses MIB Objects

The following MIB objects, values, and types are valid for remote syslog receiver IP addresses.

TABLE 5-4 MIB Objects, Values, and Types for Remote Syslog Receiver IP Addresses

MIB Object	Description	Values	Type	Default
`ilomCtrlRemoteSyslogDest1`	The IP address of the first remote syslog destination (log host).	*ip_address*	String	None
`ilomCtrlRemoteSyslogDest2`	The IP address of the second remote syslog destination (log host).	*ip_address*	String	None

▼ Configure an Alert Rule

Note – You can use the `get` and `set` commands to view and configure alert rule configurations. For a description of the MIB objects used in this procedure, see "Alert Rule Configuration MIB Objects" on page 116.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username*`@`*snmp_manager_ip_address*

   `Password:` *password*

2. **To view the severity level for the alert rule with an AlertID of 2, type:**

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlAlertSeverity.2
```

3. **To set the severity level to critical for the alert rule with an AlertID of 2, type:**

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlAlertSeverity.2 i 2
```

Alert Rule Configuration MIB Objects

The following MIB objects, values, and types are valid for alert rule settings.

TABLE 5-5 MIB Objects, Values, and Types for Alert Rule Settings

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlAlertID`	An integer ID associated with a given alert rule. **Note -** This object is not accessible.	Range: 0..65535	Integer	None
`ilomCtrlAlertSeverity`	Specifies the mininum event severity that should trigger an alert for a given class.	`disable(1)`, `critical(2)`, `major(3)`, `minor(4)`, `down(5)`	Integer	None

`ilomCtrlAlertType`	Specifies the type of notification for a given alert. If the type is `snmptrap(2)` or `ipmipet(3)`, the `ilomCtrlAlertDestinationip` must be specified. If the type is `email(1)`, the `ilomCtrlAlertDestinationEmail` must be specified.	`email(1)`
		`snmptrap(2)`
		`ipmipet(3)`
		remotesyslog (4)
			Integer	None

`ilomCtrlAlertDestinationip`	Specifies the IP address to send alert notifications when the alert type is `snmptrap(2)`, `ipmipet(3)`, or `remotesyslog(4)`.	*ip_address*	String	None
`ilomCtrlAlertDestinationEmail`	Specifies the email address to send alert notifications when the alert type is `email(1)`.	*email address*, size: 0..255	String	None
`ilomCtrlAlertSNMPVersion`	Specifies the version of SNMP trap that should be used for the given alert rule.	`v1(1)`, `v2c(2)`, `v3(3)`	Integer	None

TABLE 5-5 MIB Objects, Values, and Types for Alert Rule Settings *(Continued)*

MIB Object	Description	Allowed Values	Type	Default
`ilomCtrlAlert SNMPCommunity OrUsername`	Specifies the community string to be used when the `ilomCtrlAlertSNMPVersion` property is set to `v1(1)` or `v2c(2)`. Specifies the SNMP user name to use when the `ilomCtrlAlertSNMPVersion` is set to `v3(3)`.	Size: 0..255	String	None
ilomCtrlAlert EmailEvent ClassFilter	A class name or `all` to filter emailed alerts on.	Size: 0..255	String	None
`ilomCtrlAlert EmailEventType Filter`	A class name or `all` to filter emailed alerts on.	Size 0..255	String	None

Configuring SMTP Client for Email Alert Notifications

Topics

Description	Links
Configure SMTP Client for Email and alert notification	• "Configure SMTP Client for Email Notification Alerts" on page 118

Before You Begin

- To generate configured Email Notification alerts, you must enable the ILOM client to act as an SMTP client to send the email alert messages. To enable the ILOM client as an SMTP client, you must specify the IP address and port number of an outgoing SMTP email server that will process the email notifications.
- Prior to enabling the ILOM client as an SMTP client, gather the IP address and port number of the outgoing SMTP email server.
- You can use the `get` and `set` commands to configure the SMTP client. For a description of the MIB objects used in this procedure, see "SMTP Client MIB Objects" on page 119 and the SUN-ILOM-CONTROL-MIB.

▼ Configure SMTP Client for Email Notification Alerts

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ip_address*

   `Password:` *password*

2. **Refer to the following SNMP commands for examples:**

   - To view a SMTP client state, type:

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlSMTPEnabled.0
```

   - To set a SMTP client state to enabled, type:

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlSMTPEnabled.0 i 1
```

   - To view a SMTP server IP address, type:

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlSMTPServerip.0
```

   - To set a SMTP server IP address, type:

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlSMTPServerip.0 s ip_address
```

   - To view a SMTP client port number, type:

```
% snmpget -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlSMTPPortNumber.0
```

   - To set a SMTP client port number, type:

```
% snmpset -v2c -cprivate -mALL SNMP_agent_ip_address
ilomCtrlSMTPPortNumber.0 i 25
```

   - To view an optional format to identify the sender or the 'from' address, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSMTPCustomSender.0
```

- To configure an optional format to identify the sender or the 'from' address, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSMTPCustomSender.0 s 'ilom-alert@HOSTNAME.abc.com'
```

SMTP Client MIB Objects

The following MIB objects, values, and types are valid settings for SMTP clients.

TABLE 5-6 Valid MIB Objects, Values, and Types for SMTP Clients

MIB Object	Property	Allowed Values	Type	Default
`ilomCtrlSMTP Enabled`	Specifies whether or not the SMTP client is enabled.	`true(1), false(2)`	Integer	false
`ilomCtrlSMTP Serverip`	The IP address of the SMTP server used as a name service for user accounts.	*ip_address*	String	None
`ilomCtrlSMTP PortNumber`	Specifies the port number for the SMTP client.	Range: 0..65535	Integer	None

Configuring Email Alert Settings

Topics

Description	Links
Configure email alert settings	• "View and Configure Email Alert Settings" on page 119

▼ View and Configure Email Alert Settings

Note – You can use the `get` and `set` commands to view and configure email alert settings. For a description of the MIB objects used in these commands, see the SUN-ILOM-CONTROL-MIB.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To view the optional format used to identify the sender or the 'from' address, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlAlertEmailCustomSender.0
```

   - To set the optional format used to identify the sender or the 'from' address, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlAlertEmailCustomSender.0 s 'ilom-
alert@HOSTNAME.abc.com'
```

   - To view an optional string that can be added to the beginning of the message body, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlAlertEmailMessagePrefix.0
```

   - To define an optional string (for example: BeginMessage) that can be added to the beginning of the message body, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlAlertEmailMessagePrefix.0 s 'BeginMessage'
```

Configuring Telemtry Harness Daemon

Topics

Description	Links
Configure telemetry harness daemon settings	• "View and Configure Telemetry Harness Daemon Settings" on page 121

▼ View and Configure Telemetry Harness Daemon Settings

Note – You can use the `get` and `set` commands to view and configure Telemetry Harness Daemon (THD) settings. For a description of the MIB objects used in these commands, see the SUN-ILOM-CONTROL-MIB.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username***`@`***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**
   - To view the state of the THD daemon, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlThdState.0
```

   - To view the control action for THD daemon, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlThdAction.0
```

   - To set the control action for THD daemon to `suspend`, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlThdAction.0 i 1
```

- To view the description of the THD module named `THDMod1`, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlThdModuleDesc.'THDMod1'
```

- To view the state of the THD module named `THDMod1`, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlThdModuleState.'THDMod1'
```

- To view the control action for the THD module named `THDMod1`, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlThdModuleAction.'THDMod1'
```

- To set the control action for the THD module named `THDMod1` to `suspend`, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlThdModuleAction.0 i 1
```

- To view the state of the THD instance named `myTHDinstance` that is in the THD class named `myTHDclase`, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlThdInstanceState.'myTHDclass.myTHDinstance'
```

- To view the action of the THD instance named `myTHDinstance` that is in the THD class named `myTHDclase`, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlThdInstanceAction.'myTHDclass.myTHDinstance'
```

- To set the action of the THD instance named `myTHDinstance` that is in the THD class named `myTHDclase` to `resume`, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlThdInstanceAction.'myTHDclass.myTHDinstance' i 2
```

CHAPTER 6

Monitor and Manage Power Consumption Using SNMP

Topics

Description	Links
Review the prerequisites	• "Before You Begin" on page 124
Monitor the power consumption interfaces	• "Monitor System Total Power Consumption" on page 125 • "Monitor Actual Power Consumption" on page 126 • "Monitor Individual Power Supply Consumption" on page 126 • "Monitor Available Power" on page 128 • "Monitor Hardware Configuration Maximum Power Consumption" on page 128 • "Monitor Permitted Power Consumption" on page 128 • "Monitor Power Management Settings" on page 128
View and set power policy	• "View and Set the Power Policy" on page 129

Related Topics

For ILOM	Section	Guide
• Concepts	• Power Monitoring and Management Interfaces	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide* (820-6410)
• Web	• Monitoring Power Consumption	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide* (820-6411)
• CLI	• Monitoring Power Consumption	*Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide* (820-6412)

The ILOM 3.0 Documentation Collection is available at:
`http://docs.sun.com/app/docs/prod/int.lights.mgr30#hic`

Before You Begin

Prior to performing the procedures in this chapter, you should ensure that the following requirements are met.

- Before you can use SNMP to view and configure ILOM settings, you must configure SNMP. For more information, see "Preparing Your System to Use SNMP" on page 5.
- To execute the `snmpset` command, you need to use an SNMP v1 or v2c community or an SNMP v3 user with read/write (rw) privileges.
- For examples of SNMP commands, see "SNMP Command Examples" on page 295.

Note – The example SNMP commands presented in this section are based on the Net-SNMP sample applications and, therefore, will only work as presented if you have Net-SNMP and the Net-SNMP sample applications installed.

Monitoring the Power Consumption Interfaces

Topics

Description	Links
Monitor the power consumption interfaces	• "Monitor System Total Power Consumption" on page 125 • "Monitor Actual Power Consumption" on page 126 • "Monitor Individual Power Supply Consumption" on page 126 • "Monitor Available Power" on page 128 • "Monitor Hardware Configuration Maximum Power Consumption" on page 128 • "Monitor Permitted Power Consumption" on page 128
View and set power policy	• "View and Set the Power Policy" on page 129

Note – The power consumption interfaces described in this chapter might or might not be implemented on the platform that you are using. See the platform-specific ILOM Supplement, Platform Administration guide, or Product Notes for implementation details. You can find the ILOM Supplement, Platform Administration guide and Product Notes within the documentation set for your system.

Note – The ability to view and set the power policy is not available on SPARC platforms using ILOM 3.0 or ILOM 3.0.2. The power policy setting is available on some SPARC platforms starting with ILOM 3.0.3.

▼ Monitor System Total Power Consumption

- **To view total system power consumption using SNMP, type this command:**

```
% snmpget -v2c -cprivate -mALL snmp_agent_ipaddress
entPhysicalName.308
```

▼ Monitor Actual Power Consumption

- **To view actual power consumption using SNMP, type this command:**

```
% snmpget -v2c -cprivate -mALL snmp_agent_ipaddress
sunHwCtrlPowerMgmtActual.0
```

▼ Monitor Individual Power Supply Consumption

Before you can use SNMP to monitor individual power supply consumption, you must determine the `entPhysicalName` index numbers that correspond to the output and input power sensors for a particular power supply.

- **To view the individual power supply consumption, type a command similar to the following command.**

  For example, if you know that the `entPhysicalIndex` of `/SYS/VPS` is 303, you can view total output power consumption by typing the following command:

```
% snmpget -v2c -cprivate -mALL snmp_agent_ipaddress \
  entPhysicalName.303 \
  entPhysicalClass.303 \
  entPhysicalDescr.303 \
  sunPlatNumericSensorBaseUnits.303 \
  sunPlatNumericSensorExponent.303 \
  sunPlatNumericSensorCurrent.303 \
  sunPlatNumericSensorLowerThresholdNonCritical.303 \
  sunPlatNumericSensorUpperThresholdNonCritical.303 \
  sunPlatNumericSensorLowerThresholdCritical.303 \
  sunPlatNumericSensorUpperThresholdCritical.303 \
  sunPlatNumericSensorLowerThresholdFatal.303 \
  sunPlatNumericSensorUpperThresholdFatal.303
```

TABLE 6-1 provides a brief description of each of the MIB objects included in the above command example. For more information, see the ENTITY-MIB and the SUN-PLATFORM-MIB.

TABLE 6-1 Individual Power Supply Consumption MIB Objects

MIB Object	MIB Name	Description
`entPhysicalName`	ENTITY-MIB	The textual name of the physical entity.
`entPhysicalClass`	ENTITY-MIB	The general hardware type of the physical entity.
`entPhysicalDescr`	ENTITY-MIB	A textual description of physical entity.

TABLE 6-1 Individual Power Supply Consumption MIB Objects *(Continued)*

MIB Object	MIB Name	Description
sunPlatNumeric SensorBaseUnits	SUN-PLATFORM-MIB	The base unit of the values returned by this sensor as per CIM_NumericSensor.BaseUnits.
sunPlatNumeric SensorExponent	SUN-PLATFORM-MIB	The exponent to be applied to the units returned by this sensor as for CIM_NumericSensor.UnitModifier.
sunPlatNumeric SensorCurrent	SUN-PLATFORM-MIB	The sunPlatDiscreteSensorStatesIndex of a row in the sunPlatDiscreteSensorStatesTable that corresponds to the current reading of the sensor.
sunPlatNumeric SensorLower ThresholdNon Critical	SUN-PLATFORM-MIB	The lower threshold at which a NonCritical condition occurs as defined for CIM_NumericSensor.LowerThreshold NonCritical.
sunPlatNumeric SensorUpper ThresholdNon Critical	SUN-PLATFORM-MIB	The upper threshold at which a NonCritical condition occurs as defined for CIM_NumericSensor.UpperThreshold NonCritical.
sunPlatNumeric SensorLower ThresholdCritical	SUN-PLATFORM-MIB	The lower threshold at which a Critical condition occurs as defined for CIM_NumericSensor.LowerThreshold Critical.
sunPlatNumeric SensorUpper ThresholdCritical	SUN-PLATFORM-MIB	The upper threshold at which a Critical condition occurs as defined for CIM_NumericSensor.UpperThreshold Critical.
sunPlatNumeric SensorLower ThresholdFatal	SUN-PLATFORM-MIB	The lower threshold at which a Fatal condition occurs as defined for CIM_NumericSensor.LowerThreshold Fatal.
sunPlatNumeric SensorUpper ThresholdFatal	SUN-PLATFORM-MIB	The upper threshold at which a Fatal condition occurs as defined for CIM_NumericSensor.UpperThreshold Fatal.

▼ Monitor Available Power

- **To view total available power using SNMP, type this command:**

```
% snmpget -v2c -cprivate -mALL snmp_agent_ipaddress
sunHwCtrlPowerMgmtAvailablePower.0
```

▼ Monitor Hardware Configuration Maximum Power Consumption

- **To view the hardware configuration maximum power consumption using SNMP, type this command:**

```
% snmpget -v2c -cprivate -mALL snmp_agent_ipaddress
sunHwCtrlPowerMgmtHWConfigPower.0
```

▼ Monitor Permitted Power Consumption

- **To view permitted power consumption using SNMP, type this command:**

```
% snmpget -v2c -cprivate -mALL snmp_agent_ipaddress
sunHwCtrlPowerMgmtPermittedPower.0
```

▼ Monitor Power Management Settings

Note – You can use the `get` command to view power management settings. For a description of the MIB objects used in these commands, see the SUN-HW-CTRL-MIB.

---

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username*@*snmp_manager_ipaddress*

   Password: *password*

2. **Refer to the following SNMP command examples:**

- To view the name of the power management policy for `PowerMgmtTable` index number 5, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
sunHwCtrlPowerMgmtName.5
```

- To view the units for the value of the power management policy for `PowerMgmtTable` index number 5, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
sunHwCtrlPowerMgmtUnits.5
```

- To view the value of the power management policy for `PowerMgmtTable` index number 5, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
sunHwCtrlPowerMgmtValue.5
```

Using the Power Consumption Control Interfaces

Topics

Description	Links
View and set power policy	• "View and Set the Power Policy" on page 129

▼ View and Set the Power Policy

Note – You can use the `get` and `set` commands to view and set power policy.

1. To view the power policy using SNMP, type this command:

```
% snmpget -v2c -cprivate -mALL snmp_agent_ipaddress
sunHwCtrlPowerMgmtPolicy.0
```

2. **To set the power policy, use the** `snmpset` **command.**

   For example, to set this MIB object to `performance`, type this command:

```
% snmpset -v2c -cprivate -mALL snmp_agent_ipaddress
sunHwCtrlPowerMgmtPolicy.0 i 3
```

TABLE 6-2 shows the MIB object type and values that are supported by the `sunHwCtrlPowerMgmtPolicy` MIB object.

TABLE 6-2 Valid Values and Type for the sunHwCtrlPowerMgmtPolicy MIB Object

MIB Object	Values	Type	Default
`sunHwCtrlPowerMgmtPolicy`	`notsupported(1), unknown(2), performance(3), elastic(4)`	Integer	None

CHAPTER 7

Configure ILOM Firmware Settings Using SNMP

Topics

Description	Links
Review the prerequisites	• "Before You Begin" on page 132
Configuring ILOM firmware interfaces	• "View and Configure ILOM Firmware Settings" on page 132

Related Topics

For ILOM	Section	Guide
• Concepts	• Configuration Management and Firmware Updates	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide* (820-6410)
• Web	• Updating ILOM Firmware	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide* (820-6411)
• CLI	• Updating ILOM Firmware	*Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide* (820-6412)

The ILOM 3.0 Documentation Collection is available at:
http://docs.sun.com/app/docs/prod/int.lights.mgr30#hic

Before You Begin

Prior to performing the procedures in this chapter, you should ensure that the following requirements are met.

- Before you can use SNMP to view and configure ILOM settings, you must configure SNMP. For more information, see "Preparing Your System to Use SNMP" on page 5.
- To execute the `snmpset` command, you need to use an SNMP v1 or v2c community or an SNMP v3 user with read/write (rw) privileges.
- For examples of SNMP commands, see "SNMP Command Examples" on page 295.

Note – The example SNMP commands presented in this section are based on the Net-SNMP sample applications and, therefore, will only work as presented if you have Net-SNMP and the Net-SNMP sample applications installed.

Configuring ILOM Firmware Interfaces

▼ View and Configure ILOM Firmware Settings

Note – You can use the `get` and `set` commands to view and configure ILOM firmware settings. For a description of the MIB objects used in these commands, see the SUN-ILOM-CONTROL-MIB.

---

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   `ssh` *username*`@`*snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**
   - To view the version of the current firmware image, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwareMgmtVersion.0
```

- To view the build number of the current firmware image, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwareMgmtBuildNumber.0
```

- To view the build date and time of the current firmware image, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwareMgmtBuildDate.0
```

- To view the IP address of the TFTP server that will be used to download the firmware image, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwareTFTPServerIP.0
```

- To set the IP address of the TFTP server that will be used to download the firmware image, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwareTFTPServerIP.0 s ipaddress
```

- To view the relative path of the new firmware image file on the TFTP server, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwareTFTPFileName.0
```

- To set the relative path of the new firmware image file on the TFTP server, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwareTFTPFileName.0 s 'tftpfilename'
```

- To view the property that determines whether the previous configuration of the server should be preserved after a firmware update, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwarePreserveConfig.0
```

- To set the `PreservConfig` property to true so that the previous configuration of the server is preserved after a firmware update, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwarePreserveConfig.0 i 1
```

- To view the property that indicates the status of a firmware update, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwareMgmtStatus.0
```

- To view the property that is used to initiate a firmware update using the values of the other firmware management properties as parameters, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwareMgmtAction.0
```

- To set the property so as to initiate a firmware update using the values of the other firmware management properties as parameters, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwareMgmtAction.0 i 2
```

- To clear the values of the other firmware management properties used if and when a firmware update is initiated, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwareMgmtAction.0 i 1
```

- To view the version of the current firmware management file system, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwareMgmtFilesystemVersion.0
```

- To view the property that is used to postpone the BIOS upgrade until the next server power off, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwareDelayBIOS.0
```

- To set the `DelayBIOS` property to postpone the BIOS upgrade until the next server power off, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlFirmwareDelayBIOS.0 i 1
```

CHAPTER 8

Manage ILOM Configuration Management Interfaces Using SNMP

Topics

Description	Links
Review the prerequisites	• "Before You Begin" on page 136
Configuring ILOM configuration management interfaces	• "View and Configure Policy Settings" on page 137 • "Configure Power Setting" on page 137 • "View and Configure Backup and Restore Settings" on page 138 • "Configure the Reset Setting" on page 139

Related Topics

For ILOM	Section	Guide
• Concepts	• Configuration Management and Firmware Updates	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide* (820-6410)
• Web	• Backing Up and Restoring the ILOM Configuration	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide* (820-6411)
• CLI	• Backing Up and Restoring the ILOM Configuration	*Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide* (820-6412)

The ILOM 3.0 Documentation Collection is available at:
http://docs.sun.com/app/docs/prod/int.lights.mgr30#hic

Before You Begin

Prior to performing the procedures in this chapter, you should ensure that the following requirements are met.

- Before you can use SNMP to view and configure ILOM settings, you must configure SNMP. For more information, see "Preparing Your System to Use SNMP" on page 5.
- To execute the `snmpset` command, you need to use an SNMP v1 or v2c community or an SNMP v3 user with read/write (rw) privileges.
- For examples of SNMP commands, see "SNMP Command Examples" on page 295.

Note – The example SNMP commands presented in this section are based on the Net-SNMP sample applications and, therefore, will only work as presented if you have Net-SNMP and the Net-SNMP sample applications installed.

Configuring ILOM Configuration Management Interfaces

Topics

Description	Links
Configure ILOM configuration management interfaces	• "View and Configure Policy Settings" on page 137 • "Configure Power Setting" on page 137 • "View and Configure Backup and Restore Settings" on page 138 • "Configure the Reset Setting" on page 139

▼ View and Configure Policy Settings

Note – You can use the `get` and `set` commands to view and configure policy settings. For a description of the MIB objects used in these commands, see the SUN-ILOM-CONTROL-MIB.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**
   - To view a short description of the policy for policy ID number 2, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlPolicyShortStr.2
```

   - To view a verbose description of the policy for policy ID number 2, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlPolicyLongStr.2
```

   - To view the status of the policy for policy ID number 2, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlPolicyEnabled.2
```

   - To set the status of the policy for policy ID number 2 enabled, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlPolicyEnabled.2 i 1
```

▼ Configure Power Setting

Note – You can use the `set` command to configure the power setting. For a description of the MIB object used in this command, see the SUN-ILOM-CONTROL-MIB.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command example:**

   - To specify the action "powerOn" and apply it to the power control target named '/SYS', type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlPowerAction.'/SYS' i 1
```

▼ View and Configure Backup and Restore Settings

Note – You can use the `get` and `set` commands to view and configure backup and restore settings. For a description of the MIB objects used in these commands, see the SUN-ILOM-CONTROL-MIB.

---

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To restore the configuration on the SP to the original factory default state, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlResetToDefaultsAction.0 i 3
```

   - To view the target destination of configuration XML file during backup and restore operation, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
lomCtrlBackupAndRestoreTargetURI.0
```

- To set the target destination of configuration XML file during the backup and restore operation to tftp://10.8.136.154/remotedir/config_backup.xml, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
lomCtrlBackupAndRestoreTargetURI.0 s
'tftp://10.8.136.154/remotedir/config_backup.xml'
```

- To set the passphrase to encrypt or decrypt sensitive data during the backup and restore operation, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlBackupAndRestorePassphrase.0 s 'passphrase'
```

- To view the property used to issue a action, either backup or restore, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlBackupAndRestoreAction.0
```

- To issue a restore action using the `ilomCtrlBackupAndRestoreAction` MIB object, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlBackupAndRestoreAction.0 i 2
```

- To monitor the current status of backup or restore operation, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlBackupAndRestoreActionStatus.0
```

▼ Configure the Reset Setting

Note – You can use the `set` command to configure the reset setting. For a description of the MIB objects used in this command, see the SUN-ILOM-CONTROL-MIB.

---

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command example:**

- To specify the action “reset” and apply it to the reset control target named ‘/SP’, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlResetAction.’/SP’ i 1
```

CHAPTER 9

Manage SPARC System Configuration Management Interfaces Using SNMP

Topics

Description	Links
Review the prerequisites	• "Before You Begin" on page 142
SPARC management interfaces	• "View and Configure SPARC Diagnostic Settings" on page 143 • "View and Configure SPARC Host Settings" on page 146 • "View and Configure SPARC Boot Mode Settings" on page 149 • "View and Configure SPARC Keyswitch Setting" on page 150

Related Topics

For ILOM	Section	Guide
• Concepts	• Remote Host Management Options	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide* (820-6410)
• Web	• Managing Remote Hosts	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide* (820-6411)
• CLI	• Managing Remote Hosts	*Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide* (820-6412)

The ILOM 3.0 Documentation Collection is available at:
http://docs.sun.com/app/docs/prod/int.lights.mgr30#hic

Before You Begin

Prior to performing the procedures in this chapter, you should ensure that the following requirements are met.

- Before you can use SNMP to view and configure ILOM settings, you must configure SNMP. For more information, see "Preparing Your System to Use SNMP" on page 5.
- To execute the `snmpset` command, you need to use an SNMP v1 or v2c community or an SNMP v3 user with read/write (rw) privileges.
- For examples of SNMP commands, see "SNMP Command Examples" on page 295.

Note – The example SNMP commands presented in this section are based on the Net-SNMP sample applications and, therefore, will only work as presented if you have Net-SNMP and the Net-SNMP sample applications installed.

Configuring SPARC Management Interfaces

Topics

Description	Links
Review the prerequisites	• "Before You Begin" on page 142
SPARC management interfaces	• "View and Configure SPARC Diagnostic Settings" on page 143 • "View and Configure SPARC Host Settings" on page 146 • "View and Configure SPARC Boot Mode Settings" on page 149 • "View and Configure SPARC Keyswitch Setting" on page 150

▼ View and Configure SPARC Diagnostic Settings

Note – You can use the `get` and `set` commands to view and configure SPARC diagnostic settings. For a description of the MIB objects used in these commands, see the SUN-ILOM-CONTROL-MIB.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To view the triggers of embedded diagnostics for the host, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsTrigger.0
```

   - To set the triggers of embedded diagnostics for the host to "powerOnReset", type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsTrigger.0 i 4
```

   - To view the modes for POST, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsMode.0
```

   - To set the POST mode to `service`, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsMode.0 i 3
```

   - To view the level of embedded diagnostics that should be run on the host during a boot for the power-on-reset trigger, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsPowerOnLevel.0
```

- To set the level of embedded diagnostics that should be run on the host during a boot for the power-on-reset trigger to `normal`, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsPowerOnLevel.0 i 3
```

- To view the level of embedded diagnostics that should be run on the host during a boot for the user-reset trigger, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsUserResetLevel.0
```

- To set the level of embedded diagnostics that should be run on the host during a boot for the user-reset trigger to `normal`, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsUserResetLevel.0 i 3
```

- To view the level of embedded diagnostics that should be run on the host during a boot for the error-reset trigger, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsErrorResetLevel.0
```

- To set the level of embedded diagnostics that should be run on the host during a boot for the error-reset trigger to `normal`, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsErrorResetLevel.0 i 3
```

- To view the verbosity level of embedded diagnostics that should be run on the host during a boot, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsPowerOnVerbosity.0
```

- To set the verbosity level of embedded diagnostics that should be run on the host during a boot to `maximum`, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsPowerOnVerbosity.0 i 4
```

- To view the verbosity level of embedded diagnostics that should be run on the host during a boot for user-reset trigger, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsUserResetVerbosity.0
```

- To set the verbosity level of embedded diagnostics that should be run on the host during a boot for user-reset trigger to `maximum`, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsUserResetVerbosity.0 i 4
```

- To view the verbosity level of embedded diagnostics that should be run on the host during a boot for error-reset trigger, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsErrorResetVerbosity.0
```

- To set the verbosity level of embedded diagnostics that should be run on the host during a boot for error-reset trigger to `maximum`, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsErrorResetVerbosity.0 i 4
```

- To view the progress of POST diagnostics on the host, expressed as a percentage, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsStatus.0
```

- To view the property that shows the action to control the POST diagnostics on the host, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsAction.0
```

- To set the property to take control of the POST diagnostics running on the host to `start`, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCDiagsAction.0 i 2
```

▼ View and Configure SPARC Host Settings

Note – You can use the `get` and `set` commands to view and configure SPARC host settings. For a description of the MIB objects used in these commands, see the SUN-ILOM-CONTROL-MIB.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To view the starting MAC address for the host, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostMACAddress.0
```

   - To view the version string for OpenBoot PROM (OBP), type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostOBPVersion.0
```

   - To view the version string for POST, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostPOSTVersion.0
```

   - To view the option that determines whether the host should continue to boot in the event of a non-fatal POST error, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostAutoRunOnError.0
```

   - To configure the host to continue to boot in the event of a non-fatal POST error, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostAutoRunOnError.0 i 1
```

   - To view the string that describes the status of POST, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostPOSTStatus.0
```

- To view the option that determines what action the SP will take when it discovers that the host is hung, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostAutoRestartPolicy.0
```

- To configure the SP to reset when it discovers that the host is hung, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostAutoRestartPolicy.0 i 2
```

- To view the string that describes the boot status of host operating system, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostOSBootStatus.0
```

- To view the boot timer time-out value, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostBootTimeout.0
```

- To set the boot timer time-out value to 30 seconds, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostBootTimeout.0 i 30
```

- To view the property that determines what action the SP will take when the boot timer expires, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostBootRestart.0
```

- To configure the SP to reset when the boot timer expires, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostBootRestart.0 i 2
```

- To view the maximum number of boot failures allowed by the SP, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostMaxBootFail.0
```

- To set the maximum number of boot failures allowed by the SP to 10, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostMaxBootFail.0 i 10
```

- To view the property that determines what action the SP will take when the maximum number of boot failures is reached, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostBootFailRecovery.0
```

- To configure the SP to power cycle the host when the maximum number of boot failures is reached, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostBootFailRecovery.0 i 2
```

- To view the version string for the Hypervisor, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostHypervisorVersion.0
```

- To view the version string for the system firmware (SysFw), type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostSysFwVersion.0
```

- To view the property that determines the break action that SP will send, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostSendBreakAction.0
```

- To configure the SP to send a `dumpcore` break action, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostSendBreakAction.0 i 3
```

- To view the property that determines the host I/O reconfiguration policy to apply on next host power-on, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostIoReconfigurePolicy.0
```

- To configure the SP to execute the host I/O reconfiguration policy on the next power-on, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCHostIoReconfigurePolicy.0 i 3
```

▼ View and Configure SPARC Boot Mode Settings

Note – You can use the `get` and `set` commands to view and configure SPARC boot mode settings. For a description of the MIB objects used in these commands, see the SUN-ILOM-CONTROL-MIB.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username***@***snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**

   - To view the boot mode state for the host, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCBootModeState.0
```

   - To configure the host to retain current NVRAM variable settings, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCBootModeState.0 i 1
```

   - To view the boot script to use when the boot mode state is set to script, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCBootModeScript.0
```

   - To specify the boot script to use when the boot mode state is set to `'setenv diag-switch'`, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCBootModeScript.0 s 'setenv diag-switch'
```

- To view date and time when the boot mode configuration will expire, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCBootModeExpires.0
```

- To view the string that refers to the LDOM configuration name, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCBootModeLDOMConfig.0
```

- To set the LDOM configuration name to `default`, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCBootModeLDOMConfig.0 s default
```

▼ View and Configure SPARC Keyswitch Setting

Note – You can use the `get` and `set` commands to view and configure SPARC key switch settings. For a description of the MIB objects used in these commands, see the SUN-ILOM-CONTROL-MIB.

1. **Log in to a host that has an SNMP tool and the ILOM MIBs installed. For example, type:**

   ssh *username*@*snmp_manager_ipaddress*

   `Password:` *password*

2. **Refer to the following SNMP command examples:**
   - To view the current state of the virtual key switch, type:

```
% snmpget -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCKeySwitchState.0
```

   - To set the state of the virtual key switch to `standby`, type:

```
% snmpset -mALL -v2c -cprivate SNMP_agent_ipaddress
ilomCtrlSPARCKeySwitchState.0 i 2
```

PART II IPMI

Part II of this document provides an overview of the Intelligent Platform Management Interface (IPMI), and descriptions of the procedures you can perform to access ILOM functions.

CHAPTER 10

Server Management Using IPMI

Topics

Description	Links
Learn about IPMI	• "About Intelligent Platform Management Interface" on page 154 • "IPMItool" on page 155 • "IPMI Alerts" on page 155 • "IPMI Administrator and Operator Roles" on page 156
Learn how to configure the IPMI state and how to use IPMItool	• "Configuring the IPMI State" on page 157 • "Performing ILOM Tasks Using IPMItool" on page 160
Learn about the IPMI commands	• "IPMI Commands" on page 169

Related Topics

For ILOM	Section	Guide
• Concepts	• ILOM Overview	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide* (820-6410)
• CLI	• CLI Overview	*Oracle Integrated Lights Out Manager (ILOM) 3.0 CLI Procedures Guide* (820-6412)

Related Topics

For ILOM	Section	Guide
• Web interface	• Web Interface Overview	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Web Interface Procedures Guide* (820-6411)
• SNMP	• SNMP Overview	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Management Protocols Reference Guide* (820-6413)

The ILOM 3.0 Documentation Collection is available at:
http://docs.sun.com/app/docs/prod/int.lights.mgr30#hic

About Intelligent Platform Management Interface

ILOM supports the Intelligent Platform Management Interface (IPMI), which enables you to monitor and control your server platform, as well as to retrieve information about your server platform.

IPMI is an open, industry-standard interface that was designed for the management of server systems over a number of different types of networks. IPMI functionality includes field-replaceable unit (FRU) inventory reporting, system monitoring, logging of system events, system recovery (including system resets and power on and power off capabilities), and alerting.

The monitoring, logging, system recovery, and alerting functions available through IPMI provide access to the manageability that is built into the platform hardware.

ILOM is compliant with IPMI v1.5 and v2.0.

An Oracle-provided Windows port of IPMItool is available at
http://www.sun.com/system-management/tools.jsp

Additional information, including detailed specifications about IPMI, is available at the following sites:

- http://www.intel.com/design/servers/ipmi/spec.htm
- http://openipmi.sourceforge.net

The service processors (SPs) on your servers and server modules (blades) are IPMI v2.0 compliant. You can access IPMI functionality through the command line using the IPMItool utility either in-band (using the host operating system running on the

server) or out-of-band (using a remote system). Additionally, you can generate IPMI-specific traps from the ILOM web interface, or manage the SP's IPMI functions from any external management solution that is IPMI v1.5 or v2.0 compliant.

IPMItool

IPMItool is an open-source, simple command-line interface (CLI) utility for managing and configuring IPMI-enabled devices. IPMItool can be used to manage the IPMI functions of either the local system or a remote system. You can use the IPMItool utility to perform IPMI functions with a kernel device driver or over a LAN interface. You can download IPMItool from this site:

`http://ipmitool.sourceforge.net/`

You can do the following with IPMItool:

- Read the Sensor Data Record (SDR) repository.
- Print sensor values.
- Display the contents of the system event log (SEL).
- Print field-replaceable unit (FRU) inventory information.
- Read and set LAN configuration parameters.
- Perform remote chassis power control.

Detailed information about IPMItool is provided in a man page that is available from this site:

`http://ipmitool.sourceforge.net/manpage.html`

IPMItool supports a feature that enables you to enter ILOM command-line interface (CLI) commands just as though you were using the ILOM CLI directly. CLI commands can be scripted and then the script can be run on multiple service processor (SP) instances.

IPMI Alerts

ILOM supports alerts in the form of IPMI Platform Event Trap (PET) alerts. Alerts provide advance warning of possible system failures. Alert configuration is available from the ILOM SP on your server or server module .IPMI PET alerts are supported on all Oracle Sun server platforms and modules, with the exception of the chassis monitoring module (CMM). For more information about the types of IPMI alerts, see "Alert Management" in the *Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide.*

IPMI Administrator and Operator Roles

The *IPMI Administrator role* maps to these user roles in ILOM: `aucro`. The *IPMI Operator role* maps to these user roles in ILOM: `cro`. A brief explanation of these ILOM roles appears in TABLE 10-1.

TABLE 10-1 IPMI Administrator and Operator Roles in ILOM

IPMI Role	Enabled ILOM Role Privileges	Description
Administrator	• Admin (a) • User Management (u) • Console (c) • Reset and Host Console (r) • Read-Only (o)	These user roles enable read and write privileges to these management features in ILOM: system management configuration properties, user account properties, remote console management properties, remote power management properties, and reset and host control management properties.
Operator	• Console (c) • Reset and Host Console (r) • Read-Only (o)	These user roles enable read and write privileges to these management features in ILOM: remote console management properties, remote power management properties, and reset and host control management properties. The Read-Only role also provides read-access to system management configuration properties and user management properties.

For more information about ILOM roles and privileges, see the chapter about User Management in the *Oracle Integrated Lights Out Manager (ILOM) 3.0 Concepts Guide* (820-6410).

Configuring the IPMI State

You can enable or disable the IPMI state using either the CLI or the web interface.

Topics

Description	Links
Enable the IPMI state	• "Before You Begin" on page 157 • "Enable IPMI State Using the CLI" on page 157 • "Enable IPMI State Using the Web Interface" on page 157

Before You Begin

- To enable the IPMI state using the ILOM CLI or web interface, you need to have IPMI Administrator privileges enabled in ILOM. For more information, see "IPMI Administrator and Operator Roles" on page 156.

▼ Enable IPMI State Using the CLI

1. **Log in to the ILOM CLI.**
2. **At the command prompt, type:**

   -> **set /SP/services/ipmi servicestate=enabled**

```
-> set /SP/services/ipmi servicestate=enabled
Set 'servicestate' to 'enabled'
```

▼ Enable IPMI State Using the Web Interface

1. **Log in to the ILOM web interface.**
2. **Select Configuration --> System Management Access --> IPMI.**

   The IPMI Settings page appears.
3. **Click the check box to enable or disable the IPMI state.**

Using IPMItool to Run ILOM CLI Commands

The IPMItool CLI is a convenient alternative method to executing ILOM CLI commands. It enables you to enter ILOM CLI commands just as if you were using the ILOM CLI directly. Most ILOM CLI commands are supported.

Topics

Description	Links
Use ipmitool to run CLI commands	• "Access the ILOM CLI From IPMItool" on page 158 • "Script ILOM CLI Commands With IPMItool" on page 159

Before You Begin

- To use the ILOM CLI through `ipmitool`, you must be using `ipmitool` version 1.8.9.4 or later. To check the version number of `ipmitool`, type `ipmitool -V`.
- Ensure that you have the proper user roles assigned in ILOM when using the IPMItool CLI to execute ILOM commands. For more information, see "IPMI Administrator and Operator Roles" on page 156.

▼ Access the ILOM CLI From IPMItool

1. **To enable the ILOM CLI using IPMItool, type:**

   # **ipmitool -H** *hostname* **-U** *username* **-P** *userpassword* **sunoem cli**

   The ILOM CLI prompt appears as follows:

```
Connected. Use ^D to exit.
->
```

2. **To use the CLI, type CLI commands.**

▼ Script ILOM CLI Commands With IPMItool

A key benefit of using ILOM CLI from IPMItool is that the CLI commands can be scripted and then the script can be run on multiple SP instances. Scripting is possible because the CLI commands can be included on the IPMItool command line where each argument on the command line is treated as a separate ILOM CLI command. Command separation is archived by including quotation marks at the beginning and end of each ILOM CLI command. The following example shows how to include two CLI commands on the `ipmitool` command line. In the example, notice that each ILOM CLI command begins and ends with quotation marks.

```
# ipmitool -H hostname -U username -P userpassword sunoem cli "show
/SP/services" "show /SP/logs"
Connected. Use ^D to exit.
-> show /SP/services
/SP/services
    Targets:
        http
        https
        servicetag
        snmp
        ssh
        sso

    Properties:

    Commands:
        cd
        show

-> show /SP/logs
 /SP/logs
    Targets:
        event

    Properties:

    Commands:
        cd
        show

->Session closed
Disconnected
```

Performing ILOM Tasks Using IPMItool

Topics

Description	Links
Perform various ILOM functions using IPMItool	• "View a List of Sensors and Their Values" on page 161 • "View Details About a Single Sensor" on page 162 • "Power On the Host" on page 162 • "Power Off the Host" on page 162 • "Power Cycle the Host" on page 162 • "Set Power-Limit Budget Activation State" on page 163 • "Get Power-Limit Budget Wattage" on page 164 • "Set Power-Limit Budget Wattage" on page 165 • "Shut Down the Host Gracefully" on page 162 • "Set Power-Limit Budget Activation State" on page 163 • "Get Power-Limit Budget Wattage" on page 164 • "Set Power-Limit Budget Wattage" on page 165 • "View Manufacturing Information for FRUs" on page 167 • "View the System Event Log" on page 168

Before You Begin

- Ensure that you have the proper user roles assigned in ILOM when using the IPMItool CLI to execute ILOM commands. For more information, see "IPMI Administrator and Operator Roles" on page 156.

▼ View a List of Sensors and Their Values

```
$ ipmitool -H 1.2.3.4 -I lanplus -U username -P userpassword sdr list
/SYS/T_AMB       | 24 degrees C       | ok
/RFM0/FAN1_SPEED | 7110 RPM           | ok
/RFM0/FAN2_SPEED | 5880 RPM           | ok
/RFM1/FAN1_SPEED | 5880 RPM           | ok
/RFM1/FAN2_SPEED | 6360 RPM           | ok
/RFM2/FAN1_SPEED | 5610 RPM           | ok
/RFM2/FAN2_SPEED | 6510 RPM           | ok
/RFM3/FAN1_SPEED | 6000 RPM           | ok
/RFM3/FAN2_SPEED | 7110 RPM           | ok
/RFM4/FAN1_SPEED | 6360 RPM           | ok
/RFM4/FAN2_SPEED | 5610 RPM           | ok
/RFM5/FAN1_SPEED | 5640 RPM           | ok
/RFM5/FAN2_SPEED | 6510 RPM           | ok
/RFM6/FAN1_SPEED | 6180 RPM           | ok
/RFM6/FAN2_SPEED | 6000 RPM           | ok
/RFM7/FAN1_SPEED | 6330 RPM           | ok
/RFM7/FAN2_SPEED | 6330 RPM           | ok
/RFM8/FAN1_SPEED | 6510 RPM           | ok
/RFM8/FAN2_SPEED | 5610 RPM           | ok
```

Note – If `bimetal` is not configured to support the `-P` option, which enables the password to be entered in the command line, you will be prompted to enter the password.

Note – The above output was shortened. The actual output displays 163 sensors.

▼ View Details About a Single Sensor

```
$ ipmitool -H 1.2.3.4 -v -I lanplus -U username -P userpassword sensor get /SYS/T_AMB
Locating sensor record...
Sensor ID              : /SYS/T_AMB (0x8)
 Entity ID             : 41.0
 Sensor Type (Analog)  : Temperature
 Sensor Reading        : 24 (+/- 0) degrees C
 Status                : ok
 Lower Non-Recoverable : 0.000
 Lower Critical        : 4.000
 Lower Non-Critical    : 10.000
 Upper Non-Critical    : 35.000
 Upper Critical        : 40.000
 Upper Non-Recoverable : 45.000
 Assertions Enabled    : lnc- lcr- lnr- unc+ ucr+ unr+
 Deassertions Enabled  : lnc- lcr- lnr- unc+ ucr+ unr+
```

▼ Power On the Host

$**ipmitool -H 1.2.3.4 -v -I lanplus -U** *username* **-P** *userpassword* **chassis power on**

▼ Power Off the Host

$ **ipmitool -H 1.2.3.4 -v -I lanplus -U** *username* **-P** *userpassword* **chassis power off**

▼ Power Cycle the Host

$ **ipmitool -H 1.2.3.4 -v -I lanplus -U** *username* **-P** *userpassword* **chassis power cycle**

▼ Shut Down the Host Gracefully

$ **ipmitool -H 1.2.3.4 -v -I lanplus -U** *username* **-P** *userpassword* **chassis power soft**

▼ Set Power-Limit Budget Activation State

Note – The power-limit budget activation state commands described below enable you to set the activation-state for the power management budget interfaces.

- IPMI command to activate the power-limit budget state:

```
$ ipmitool -H <localhost|IP address> -U <username> -P <password>
raw 0x2e 0x49 0x00 0x01 0xFF 0xFF

dc
```

- IPMI command to deactivate the power-limit budget state:

```
$ ipmitool -H <localhost|IP address> -U <username> -P <password>
raw 0x2e 0x49 0x00 0x00 0xFF 0xFF

dc
```

TABLE 10-2 identifies the byte order and the field format that is used to activate or deactivate the state for the power-limit budget property.

TABLE 10-2 Byte Order and Field Format - Power-Limit Budget Activation State

	Byte	Data Fields:
Request Data	1	Sun OEM command group number “0x2e”.
	2	Command code “0x49” sets the power-limit activation state.
	3	Group extension identification “0x00”. The value for this field is ignored.
	4	Sub-commands for power-limit activation: 0x00 - Deactivate Power Limit 0x01 - Activate Power Limit
	5-6	Reserved fields: 0xFF. The values for this field are ignored.
Response Data	1	Completion code consumed by ipmitool. The system does not display a status for successful completion code. However, if the result of the completion code is anything other than 'successful', a failure message appears.
	2	Group extension identification “dc” appears upon command completion.

▼ Get Power-Limit Budget Wattage

Note – It is recommended to do a Get Power Limit Budget Wattage command prior to setting the power-limit budget wattage property.

```
$ ipmitool -H <localhost|IP address> -U <username> -P <password> raw
0x2e 0x4A 0x00 0x00 0x00

dc 01 b3 00 02 fa 00 00 00 00 01 e9 00 00
```

TABLE 10-3 identifies the byte order and the field format that is used to obtain the power limit budget wattage property.

TABLE 10-3 Byte Order and Field Format - Get Power Limit Budget Wattage

	Byte	Data Field
Request Data	1	SUN OEM command group number "0x2e."
	2	Command code "0x4A" gets Power Budget settings
	3	Group extension identification: 0x00. The value for this field is ignored.
	4-5	Reserved fields: 0x00. Values for this field are ignored.

TABLE 10-3 Byte Order and Field Format - Get Power Limit Budget Wattage *(Continued)*

	Byte	Data Field
Response Data	1	Completion code that is consumed by ipmitool. The system does not display a status for successful completion code. However, if the result of the completion code is anything other than 'successful', a failure message appears.
	2	Group extension identification "dc" appears upon command completion.
	3	Activation State. 00 - deactivated 01 - activated
	4	Reserved field: b3. The value for this field can be ignored.
	5	Exception action is taken if the power limit is exceeded and is unable to be controlled within the correction time limit. 00 - none 01 - hard power-off
	6-7	Power limit in watts: 02 fa
	8-11	Correction timelimit in milliseconds: 00 00 00 00.
	12	A flag to indicate whether the correction timelimit is the system default timelimit. 00 - not default. 01 - default.
	13	Reserved field: e9. The value for this field can be ignored..
	14-15	Reserved fields: "00 00". The value for this field can be ignored.

▼ Set Power-Limit Budget Wattage

Note – The set power limit commands sets the power limit budget wattage for the system. You should use this command to set the upper limit of system power usage. However, you should not use this command as an interface for dynamic or frequently changing power limit. The power limit set should always be persistent across AC and DC cycles.

```
$ ipmitool -H <localhost|IP address> -U <username> -P <password> raw
0x2e 0x4B 0xdc 0xff 0xff 0xff 0x01 0x02 0xaa 0x00 0x00 0x1b 0x58 0x00
0xff 0x00 0x00
dc
```

TABLE 10-4 identifies the byte order and the field format that is used to set the power limit budget wattage property.

TABLE 10-4 Byte Order and Field Format - Set Power Limit Budget Wattage

	Byte	Data Field
Request Data	1	SUNOEM command group number: `0x2e`
	2	Command code `0x4B` sets power budget settings
	3	Group extension identification: `0xdc`. The value for this field is ignored.
	4-6	Reserved fields: `0xff 0xff 0xff`. The values for this field are ignored.
	7	Exception action taken: 00 - none 01 - hard power-off
	8-9	Power limit in watts. For example: `0x2a 0xaa`
	10-13	Correction time limit in milliseconds. For example: "0x00 0x00 0x1b 0x58". This value is ignored if timelimit is set to default, see next byte.
	14	A flag indicating whether to use the system default timelimit. Correction timelimit in bytes 10-13 will be ignored. 0x00 - not default 0x01 - default
	15	Reserved field: `0xff`. The value for this field is ignored.
	16-17	Reserved field: `0x00 0x00`. The value for this field is ignored.
Response Data	1	Completion code that is consumed by ipmitool. The system does not display a status for successful completion code. However, if the result of the completion code is anything other than successful, a failure message appears.
	2	Group extension identification "`dc`" appears upon command completion.

▼ View Manufacturing Information for FRUs

```
$ ipmitool -H 1.2.3.4 -v -I lanplus -U username -P userpassword fru print
FRU Device Description : Builtin FRU Device (ID 0)
 Board Product         : ASSY,ANDY,4SKT_PCI-E,BLADE
 Board Serial          : 0000000-7001
 Board Part Number     : 501-7738-01
 Board Extra           : AXX_RevE_Blade
 Product Manufacturer  : ORACLE
 Product Name          : ILOM

FRU Device Description : /SYS (ID 4)
 Chassis Type          : Rack Mount Chassis
 Chassis Part Number   : 541-0251-05
 Chassis Serial        : 00:03:BA:CD:59:6F
 Board Product         : ASSY,ANDY,4SKT_PCI-E,BLADE
 Board Serial          : 0000000-7001
 Board Part Number     : 501-7738-01
 Board Extra           : AXX_RevE_Blade
 Product Manufacturer  : ORACLE
 Product Name          : SUN BLADE X8400 SERVER MODULE
 Product Part Number   : 602-0000-00
 Product Serial        : 0000000000
 Product Extra         : 080020ffffffffffffff0003baf15c5a

FRU Device Description : /P0 (ID 5)
 Product Manufacturer  : ADVANCED MICRO DEVICES
 Product Part Number   : 0F21
 Product Version       : 2

FRU Device Description : /P0/D0 (ID 6)
 Product Manufacturer  : MICRON TECHNOLOGY
 Product Name          : 1024MB DDR 400 (PC3200) ECC
 Product Part Number   : 18VDDF12872Y-40BD3
 Product Version       : 0300
 Product Serial        : D50209DA
 Product Extra         : 0190
 Product Extra         : 0400

FRU Device Description : /P0/D1 (ID 7)
 Product Manufacturer  : MICRON TECHNOLOGY
 Product Name          : 1024MB DDR 400 (PC3200) ECC
 Product Part Number   : 18VDDF12872Y-40BD3
 Product Version       : 0300
 Product Serial        : D50209DE
 Product Extra         : 0190
 Product Extra         : 0400
```

▼ View the System Event Log

```
$ ipmitool -H 1.2.3.4 -I lanplus -U username -P userpassword sel list
 100 | Pre-Init Time-stamp   | Power Unit #0x78 | State Deasserted
 200 | Pre-Init Time-stamp   | Power Supply #0xa2 | Predictive Failure Asserted
 300 | Pre-Init Time-stamp   | Power Supply #0xba | Predictive Failure Asserted
 400 | Pre-Init Time-stamp   | Power Supply #0xc0 | Predictive Failure Asserted
 500 | Pre-Init Time-stamp   | Power Supply #0xb4 | Predictive Failure Asserted
 600 | 04/05/2007 | 12:03:24 | Power Supply #0xa3 | Predictive Failure Deasserted
 700 | 04/05/2007 | 12:03:25 | Power Supply #0xaa | Predictive Failure Deasserted
 800 | 04/05/2007 | 12:03:25 | Power Supply #0xbc | Predictive Failure Deasserted
 900 | 04/05/2007 | 12:03:26 | Power Supply #0xa2 | Predictive Failure Asserted
 a00 | 04/05/2007 | 12:03:26 | Power Supply #0xa8 | Predictive Failure Deasserted
 b00 | 04/05/2007 | 12:03:26 | Power Supply #0xb6 | Predictive Failure Deasserted
 c00 | 04/05/2007 | 12:03:26 | Power Supply #0xbb | Predictive Failure Deasserted
 d00 | 04/05/2007 | 12:03:26 | Power Supply #0xc2 | Predictive Failure Deasserted
 e00 | 04/05/2007 | 12:03:27 | Power Supply #0xb0 | Predictive Failure Deasserted
 f00 | 04/05/2007 | 12:03:27 | Power Supply #0xb5 | Predictive Failure Deasserted
1000 | 04/05/2007 | 12:03:27 | Power Supply #0xba | Predictive Failure Asserted
1100 | 04/05/2007 | 12:03:27 | Power Supply #0xc0 | Predictive Failure Asserted
1200 | 04/05/2007 | 12:03:28 | Power Supply #0xa9 | Predictive Failure Deasserted
1300 | 04/05/2007 | 12:03:28 | Power Supply #0xae | Predictive Failure Deasserted
1400 | 04/05/2007 | 12:03:28 | Power Supply #0xb4 | Predictive Failure Asserted
1500 | 04/05/2007 | 12:03:28 | Power Supply #0xbe | Predictive Failure Deasserted
```

IPMI Commands

You can download the IPMItool utility at:

http://ipmitool.sourceforge.net/

After you install the IPMItool package, you can access detailed information about command usage and syntax from the man page that is installed. The following table summarizes available IPMItool commands.

TABLE 10-5 IPMItool commands

IPMI Command	Function
`sunoem sshkey set`	Configure an SSH key for a remote shell user.
`ipmitool sunoem sshkey del`	Remove an SSH key from a remote shell user.
`ipmitool sunoem led get`	Read LED status.
`ipmitool sunoem led set`	Set LED status.
`ipmitool sunoem cli`	Enter ILOM CLI commands as if you were using the ILOM CLI directly. The LAN/LANplus interface should be used.
`ipmitool sunoem CLI force`	Available as of ILOM 3.0.10, a `force` option can be invoked as an argument to the `sunoem` CLI command.
`ipmitool raw`	Execute raw IPMI commands.
`ipmitool lan print`	Print the current configuration for the given channel.
`ipmitool lan set (1) (2)`	Set the given parameter on the given channel.
`ipmitool chassis status`	Display information regarding the high-level status of the system chassis and main power subsystem.
`ipmitool chassis power`	Perform a chassis control command to view and change the power state.
`ipmitool chassis identify`	Control the front panel identify light. Default is 15. Use 0 to turn off.
`ipmitool chassis restart_cause`	Query the chassis for the cause of the last system restart.
`ipmitool chassis bootdev (1)`	Request the system to boot from an alternate boot device on next reboot.
`ipmitool chassis bootparam (1)`	Set the host boot parameters.

TABLE 10-5 IPMItool commands *(Continued)*

IPMI Command	Function
`ipmitool chassis selftest`	Display the BMC Self Test results.
`ipmitool power`	Return the BMC Self Test results.
`ipmitool event`	Send a predefined event to the system event log.
`ipmitool mc (1) (2)`	Instruct the BMC to perform a warm or cold reset.
`ipmitool sdr`	Query the BMC for sensor data records (SDR) and extract sensor information of a given type, then query each sensor and print its name, reading, and status.
`ipmitool sensor`	List sensors and thresholds in a wide table format.
`ipmitool fru print`	Read all field-replaceable unit (FRU) inventory data and extract such information as serial number, part number, asset tags, and short strings describing the chassis, board, or product.
`ipmitool sel`	View the ILOM SP system event log (SEL).
`ipmitool pef info`	Query the BMC and print information about the PEF supported features.
`ipmitool pef status`	Print the current PEF status (the last SEL entry processed by the BMC, etc).
`ipmitool pef list`	Print the current PEF status (the last SEL entry processed by the BMC, etc).
`ipmitool user`	Display a summary of userid information, including maximum number of userids, the number of enabled users, and the number of fixed names defined.
`ipmitool session`	Get information about the specified session(s). You can identify sessions by their ID, by their handle number, by their active status, or by using the keyword “all” to specify all sessions.
`ipmitool firewall (1)`	Enable/disable individual command and command sub-functions; determine which commands and command sub-functions can be configured on a given implementation.
`ipmitool set (1)`	Set the runtime options including session host name, user name, password and privilege level.
`ipmitool exec`	Execute IPMItool commands from file name. Each line is a complete command.

PART III WS-Management and CIM

Part III of this document identifies the requirements for managing an Oracle Sun server platform using WS-Management and a Common Information Model (CIM). Information includes overview material, a procedure for enabling the WS-Management state, and a list of the supported CIM profiles.

CHAPTER 11

Server Management Using WS-Management and CIM

Topics

Description	Links
Learn about support for WS-Management and CIM	• "WS-Management and CIM Overview" on page 174
Learn how to configure the state for WS-Management	• "Configuring Support for WS-Management in ILOM" on page 176
Learn about the supported CIM profiles and Oracle's Sun specific classes	• "Supported DMTF SMASH Profiles and CIM Classes" on page 180

Related Topics

For ILOM	Section	Guide
• Version 3.0.8	• New Features	*Oracle Integrated Lights Out Manager (ILOM) 3.0 Feature Updates and Release Notes* (820-7329)

The ILOM 3.0 Documentation Collection is available at:
http://docs.sun.com/app/docs/prod/int.lights.mgr30#hic

WS-Management and CIM Overview

As of version 3.0.8, ILOM supports the use of the Distributed Management Task Force (DMTF) Web Services for Management (WS-Management) protocol and Common Information Model (CIM). The support for these DMTF standards in ILOM enable developers to build and deploy network management applications to monitor and manage information about Oracle's Sun system hardware.

Topics described in this section, include:

Note – The WS-Management and CIM features discussed in this section are common to all Oracle Sun server platforms supporting ILOM 3.0.8 or later. For more information about additional features (or limitations) that might be applicable to your server regarding these features, see the ILOM Supplement guide or Platform Administration guide provided with your server.

WS-Management

WS-Management is based on the Simple Object Access Protocol (SOAP) specification that promotes interoperability between managed applications and managed resources. It enables you to:

- Discover the presence of management resources, as well as provide navigation among them.
- View and write to individual management resources, such as settings and dynamic values.
- Obtain a list for contents of containers and collections, such as system components and log entries.
- Run management methods.

For further details about implementing and deploying a WS-Management environment to remotely manage system hardware across your IT infrastructure, see: `http://www.dmtf.org/standards/wsman`

For more information about how to configure support for WS-Management in ILOM, see "Configuring Support for WS-Management in ILOM" on page 176.

Common Information Model (CIM)

CIM is an object-oriented information model that provides a common definition for managing system hardware data. These common definitions enable you to exchange semantically rich management information among systems on your network.

CIM supplies a set of classes that provide a framework to organize the information about the managed environment. Specifically, these classes enable you to create or use another application other than ILOM to monitor and manage Oracle's Sun hardware.

System Management Architecture for Server Management (SMASH)

Oracle's Sun hardware supports a relevant subset of SMASH profiles. For more information about DMTF SMASH profiles, consult the specification for this standard at: `http://www.dmtf.org/standards/mgmt/smash`

For more information about support SMASH profiles and CIM classes, see "Supported DMTF SMASH Profiles and CIM Classes" on page 180.

Configuring Support for WS-Management in ILOM

Refer to the following sections that describe the prerequisites and procedures for configuring support for WS-Management in ILOM.

Before You Begin

- To edit the configuration properties for WS-Management in ILOM, you must have Admin (a) role privileges.

▼ Edit the WS-Management Service State, Transport Mode, and Port Number Using the CLI

1. **Log in to the ILOM SP CLI.**

Note – Alternatively, you can log in to the ILOM CMM CLI then navigate to the SP target where you want to enable or disable the KVMS lock option for the ILOM Remote Console.

---

2. **To view all the properties associated with the management of the SP WS-Man service, type:**

   ```
   -> help /SP/services/wsman
   ```

   The following help output appears for the WS-Management service:

```
/SP/services/wsman : Management of the WSMAN service
   Targets:

   Properties:
       http_port : WSMAN http port
       http_port : User role required for set = a

       https_port : WSMAN https port
       https_port : User role required for set = a

       mode : WSMAN mode
       mode : User role required for set = a

       state : WSMAN state
       state : User role required for set = a
```

3. Perform the tasks described in the following table to navigate and manage the SP WS-Management target properties.

Task	Instructions
Navigate to the WS-Management service target.	• To navigate to the WS-Management service target, type the following command: -> **cd /SP/services/wsman** **Note -** You must navigate to the wsman target prior to viewing or configuring the properties associated with the WS-Management service.

Task	Instructions
View the WS-Management CLI properties and commands.	• To view the WS-Management properties and commands, type the following command: -> **show** The current properties and available commands that are associated with the management of the SP WS-Management service appear. For example: -> **cd /SP/services/wsman** /SP/services/wsman -> **show** /SP/services/wsman Targets: Properties: http_port = 7783 https_port = 7782 mode = http state = enabled Commands: cd set show
Set the WS-Management service state.	• To enable or disable support for the WS-Management service in ILOM, type the following command to set the service state: -> **set state=enabled** or -> **set state=disabled** **Note -** The service state for WS-Management in ILOM 3.0.8 is, by default, disabled. For all other ILOM versions, the service state is, by default, enabled.
Set the WS-Management transport mode (HTTP or HTTPS).	• To set the transport mode (HTTP or HTTPS) for the WS-Management service in ILOM, type one of the following commands: -> **set mode=http** or -> **set mode=https**

Task	Instructions
Set the WS-Management transport mode port number.	• To set the transport mode port number for the WS-Management, service in ILOM, type one of the following commands: -> **`set http_port=`**`####` or -> **`set https_port=`**`####` Where #### equals the port number to be assigned to the specified transport mode (`HTTP` or `HTTPS`). For example, to set the default port number for `HTTP` or `HTTPS`, you would type: •For `HTTP`: `set http_port=`*8889* •For `HTTPS`: `set https_port=`*8888*

4. Type `exit` to exit the ILOM CLI.

▼ Edit WS-Management State, Transport Mode, and Port Number Using the Web Interface

1. Log in to the ILOM SP web interface.

2. In the ILOM SP web interface, click the Configuration --> System Management Access --> WS-Man tab.

3. In the WS-Man page, configure the following WS-Man settings:

Settings	Instructions
Enable or disable the WS-Management service state.	• In the `State Enabled` check box, select (enable) or clear (disable) the check box. By default, this setting is disabled in ILOM.
Select a WS-Management transport mode (HTTP or HTTPS).	• In the `Mode` list box, select `HTTP` or `HTTPS`. By default, this setting is set to `HTTP`.
Set the WS-Management transport mode port number.	• In the `HTTP` or `HTTPS` text box, specify the transport mode port number for the WS-Management service. The default port number settings for `HTTP` or `HTTPS` are as follows: • `HTTP`: *8889* • `HTTPS`: *8888*

4. Click Save to apply the changes made to the WS-Man settings.

Supported DMTF SMASH Profiles and CIM Classes

Oracle-supported CIM classes provide a common information model interface for developers building management applications. With Oracle-specific CIM class properties, developers can use standards-based CIM-compliant applications to manage Oracle's Sun hardware.

For a list of the supported DMTF profiles, Oracle-specific CIM classes, as well as the supported CIM indications in ILOM, see these sections:

- Supported DMTF SMASH profiles and CIM classes, see TABLE 11-1, "Supported SMASH Profiles and CIM Classes" on page 181.
- Supported CIM indications in ILOM, see "Supported CIM Indications" on page 182.

Note – Oracle supports CIM schema version 2.18.1. For DMTF CIM schema details, see `http://www.dmtf.org/standards/cim/cim_schema_v2181/`

Note – Use name space `http://schemas.oracle.com/wbem/wscim/1/cim-schema/2` when using Oracle-specific CIM class. For example: `http://schemas.oracle.com/wbem/wscim/1/cim-schema/2/Oracle_ComputerSystem`

Note – As of ILOM 3.0.14, the Oracle Sun supported CIM classes have been renamed from Sun_xxx to Oracle_xxx. Prior to ILOM 3.0.14, the Oracle Sun CIM classes should be referenced as Sun_xxxx and not Oracle_xxx as described in this guide. For further details about Oracle-supported CIM classes, see Appendix A.

TABLE 11-1 Supported SMASH Profiles and CIM Classes

Supported DMTF Profiles	Oracle-Supported CIM Classes	Oracle-Derived Classes
Base Server http://www.dmtf.org/standards/published_documents/DSP1004.pdf	• CIM ComputerSystem • CIM_EnabledLogicalElementCapabilities • CIM_ElementCapabilities • CIM_ComputerSystemPackage • CIM_ElementConformsToProfile • CIM_SystemDevice • CIM_UseOfLog	• Oracle_ComputerSystem • Oracle_EnabledLogicalElementCapabilities • Oracle_ElementCapabilities • Oracle_Computer SystemPackage • Oracle_ElementConformsToProfile • Oracle_SystemDevice • Oracle_UseOfLog
Service Processor	• CIM_ComputerSystem • CIM_EnabledLogicalElementCapabilities • CIM_ElementCapabilities • CIM_SystemComponent	• Oracle_ComputerSystem • Oracle_EnabledLogicalElementCapabilities • Oracle_ElementCapabilities • Oracle_SystemComponent
Physical Asset http://www.dmtf.org/standards/published_documents/DSP1011.pdf	• CIM_Chip • CIM_PhysicalMemory • CIM_Chassis • CIM_PhysicalPackage • CIM_PhysicalAssetCapabilities • CIM_Container • CIM_Realizes • CIM_ComputerSystemPackage • CIM_ElementCapabilities	• Oracle_Chip • Oracle_PhysicalMemory • Oracle_Chassis • Oracle_PhysicalPackage • Oracle_PhysicalAssetCapabilities • Oracle_Container • Oracle_Realizes • Oracle_ComputerSystemPackage • Oracle_ElementCapabilities
Sensors http://www.dmtf.org/standards/published_documents/DSP1009.pdf	• CIM_Sensor • CIM_NumericSensor • CIM_AssociatedSensor • CIM_SystemDevice	• Oracle_Sensor • Oracle_NumericSensor • Oracle_AssociatedSensor • Oracle_SystemDevice
CPU http://www.dmtf.org/standards/published_documents/DSP1022.pdf	• CIM_Processor • CIM_Realizes • CIM_SystemDevice	• Oracle_Processor • Oracle_Realizes • Oracle_SystemDevice
System Memory http://www.dmtf.org/standards/published_documents/DSP1026.pdf	• CIM_Memory • CIM_Realizes • CIM_SystemDevice	• Oracle_Memory • Oracle_Realizes • Oracle_SystemDevice

TABLE 11-1 Supported SMASH Profiles and CIM Classes *(Continued)*

Supported DMTF Profiles	Oracle-Supported CIM Classes	Oracle-Derived Classes
Indicator LED (DSP0835.pdf) http://www.dmtf.org/standards/published_documents	• CIM_SystemDevice • Not Applicable* • Not Applicable* **Note -** *The CIM schema version 1.18.1 does not have the CIM_IndicatorLED and CIM_AssociatedIndicatorLED defined. The CIM_IndicatorLED and CIM_AssociatedIndicatorLED are required by the Indicator LED profile.	• CIM_SystemDevice • Oracle_IndicatorLED* • Oracle_AssociatedIndicatorLED* **Note -** *Use the CIM_IndicatorLED and CIM_AssociatedIndicatorLED from the experimental schema for CIM schema version 2.18.1 and rename them Oracle_IndicatorLED and Oracle_AssociatedIndicatorLED
Record Log (DSP0810.pdf) http://www.dmtf.org/standards/published_documents	• CIM_RecordLog • CIM_LogEntry • CIM_LogManagesRecord • CIM_UseOfLog	• Oracle_RecordLog • Oracle_LogEntry • Oracle_LogManagesRecord • Oracle_UseOfLog
Profile Registration http://www.dmtf.org/standards/published_documents/DSP1033_1.0.0.pdf	• CIM_RegisteredProfile • CIM_ElementConformsToProfile • CIM_ReferenceProfile	• Oracle_RegisteredProfile • Oracle_ElementConformsToProfile • Oracle_ReferenceProfile

Supported CIM Indications

As of ILOM 3.0.8, ILOM can generate CIM indications for the following conditions:

- Sensor crosses a threshold (CIM_ThresholdIndication).
- Hardware component changes operational state or health state (CIM_InstModification).
- Hardware component is inserted into the chassis (CIM_InstCreation).
- Hardware component is removed from the chassis (CIM_InstDeletion).

TABLE 11-2 identifies the CIM classes supported in ILOM for CIM indications.

TABLE 11-2 Oracle's Sun-Supported CIM Classes for Sensor Indications

Oracle's Sun-Supported CIM Classes for Sensor Indications	Oracle' Derived Classes for Sensor Indications
• CIM_InstCreation	• Oracle_InstCreation

TABLE 11-2 Oracle's Sun-Supported CIM Classes for Sensor Indications *(Continued)*

Oracle's Sun-Supported CIM Classes for Sensor Indications	Oracle' Derived Classes for Sensor Indications
• CIM_InstDeletion	• Oracle_InstDeletion
• CIM_InstModification	• Oracle_HWCompErrorOkIndication
• CIM_ThresholdIndication	• Oracle_ThresholdIndication

In addition, ILOM defines two static instances of `CIM_IndicationFilter`, in `/root/interop namespace` that a client can subscribe to in order to receive indication for when a threshold is crossed or for when a hardware component health state changes. TABLE 11-3 identifies the key properties and ILOM values supported for these conditions.

TABLE 11-3 Key Properties and Values for Static CIM_IndicationFilter Instances

Key Property	ILOM Value
Subscription for sensor crossing threshold	
• CreationClassName	• CIM_IndicationFilter
• Name	• ORCL:ILOM:SensorCrossingThresholdFilter (as of ILOM 3.0.14) • JAVA:ILOM:SensorCrossingThresholdFilter (prior to ILOM 3.0.14)
• SystemCreationClassName	• CIM_ComputerSystem
• SystemName	• localhost
Subscription for hardware component changes health state	
• CreationClassName	• CIM_IndicationFilter
• Name	• ORCL:ILOM:HWComponentErrorFilter (as of ILOM 3.0.14) • JAVA:ILOM:HWComponentErrorFilter (prior to ILOM 3.0.14)
• SystemCreationClassName	• CIM_ComputerSystem
• SystemName	• localhost

APPENDIX **A**

Oracle's Sun-Supported CIM Classes

This appendix identifies the properties for Oracle supported CIM classes

Topics include:

Document Conventions

The following document conventions apply to this appendix:

- Each class table within this appendix describes only the properties supported by ILOM. For all possible properties of a class, see the corresponding DMTF CIM Schema 2.18.1 at:

  http://www.dmtf.org/standards/cim/cim_schema_v2181/

- An Oracle Sun-specific property (added by Oracle Sun-derived classes) will have the word *Sun-specific*.
- *Key*-property rows are shown first in each class table, in alpha-numerical ascending order.
- Non-key-property rows are after key-property rows, in alpha-numerical ascending order.
- The term ***controller*** refers to the hardware entity on which management software resides, for example, the service processor (SP) or chassis monitoring module (CMM). The term ***controllee*** refers to the hardware entity that is controlled by the controller, for example, the host system (SYS) or the chassis (CH).
- As of ILOM 3.0.14, the Oracle Sun CIM classes have been renamed from Sun_xxx to Oracle_xxx. Prior to ILOM 3.0.14, the Oracle Sun CIM classes should be referenced as Sun_xxxx and *not* Oracle_xxx as described in this guide.

Oracle_AssociatedIndicatorLED

Description:	The `Oracle_AssociatedIndicatorLED` class associates an LED to a physical element.
Inheritance:	`CIM_Dependency`
Properties:	For a description of the supported properties for the `Oracle_AssociatedIndicatorLED` class, see TABLE A-1**.** **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-1**)**, see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Indicator LED

Note – The Indicator LED profile specifies the `CIM_AssociatedIndicatorLED` class. However, the `CIM_AssociatedIndicatorLED` class does not exist in version 2.1.8.1 of the CIM Schema. Therefore, Oracle uses the `CIM_AssociatedIndicatorLED` class that is identified in the Experimental CIM Schema 2.18.1 and renamed it to `Oracle_AssociatedIndicatorLED`.

TABLE A-1 Properties for Oracle_AssociatedSensor

Property	Data Type	Description	ILOM Value
`Antecedent`	`CIM_ManagedSystem` REF	The `Antecedent` property is a mandatory *key* property. Indicates the `ManagedSystemElement` that has an associated LED.	Object path to an instance of `CIM_ManagedSystemElement`.
`Dependent`	`Oracle_IndicatorLED` REF	The `Dependent` property is a mandatory *key* property. Represents the indicator LED of the managed element.	Object path to an instance of `Oracle_IndicatorLED`.

Oracle_AssociatedSensor

Description:	The `Oracle_AssociatedSensor` class associates a sensor to the physical element.
Inheritance:	`CIM_AssociatedSensor`
Properties:	For a description of the supported properties for the `Oracle_AssociatedSensor` class, see TABLE A-2**.** **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-2**)**, see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Sensor

TABLE A-2 Properties for Oracle_AssociatedSensor

Property	Data Type	Description	ILOM Value
`Antecedent`	`CIM_Sensor` REF	The `Antecedent` property is a mandatory *key* property. Represents the sensor for the managed element.	Object path to an instance of `CIM_Sensor`.
`Dependent`	`CIM_PhysicalElement` REF	The `Dependent` property is a mandatory *key* property. The `ManagedSystemElement` for which information is measured by the sensor.	`Object path to an instance of the CIM_PhysicalElement that the sensor belongs.`

Oracle_Chassis

Description:	The `Oracle_Chassis` class represents the physical elements that enclose other elements.
Inheritance:	`CIM_Chassis`
Properties:	For a description of the supported properties for the `Oracle_Chassis` class, see TABLE A-3. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-3), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Physical Asset

TABLE A-3 Properties for Oracle_Chassis

Property	Data Type	Description	ILOM Value
`CreationClassName`	string	The `CreationClassName` property is a mandatory *key* property `CreationClassName` indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	Set to `Oracle_Chassis`

TABLE A-3 Properties for Oracle_Chassis *(Continued)*

Property	Data Type	Description	ILOM Value
`Tag`	string	The `Tag` property is a mandatory *key* property. The `Tag` property is an arbitrary string that uniquely identifies the physical element and serves as the key of the element. The `Tag` property can contain information such as asset tag or serial number data. The *key* for `PhysicalElement` is placed very high in the object hierarchy in order to independently identify the hardware or entity, regardless of physical placement in or on cabinets, adapters, and so on. For example, a hot-swappable or removable component can be taken from its containing (scoping) Package and be temporarily unused. The object still continues to exist and can be inserted into a different scoping container. Therefore, the *key* for `PhysicalElement` is an arbitrary string and is defined independently of any placement or location-oriented hierarchy.	Set to component NAC name
`CanBeFRUed`	boolean	The `CanBeFRUed` property is a boolean that indicates whether this `PhysicalElement` can be FRUed (TRUE) or not (FALSE).	Will be set to `TRUE` or `FALSE` depending on whether the component is considered to be a FRU by the platform.

TABLE A-3 Properties for Oracle_Chassis *(Continued)*

Property	Data Type	Description	ILOM Value
ChassisPackageType	uint16[]	The ChassisPackageType property indicates the physical form factor for the type of chassis. This property may have a value when the PackageType property contains the value 3 Chassis Frame. A value of 28 Blade Enclosure indicates the Chassis is designed to contain one or more PhysicalPackage(s) of PackageType 16 "Blade" or PackageType 17 "Blade Expansion". Definition type values include any of the following: {Unknown, Other, SMBIOS Reserved, Desktop, Low Profile Desktop, Pizza Box, Mini Tower, Tower, Portable, LapTop, Notebook, Hand Held, Docking Station, All in One, Sub Notebook, Space-Saving, Lunch Box, Main System Chassis, Expansion Chassis, SubChassis, Bus Expansion Chassis, Peripheral Chassis, Storage Chassis, SMBIOS Reseved, Sealed-Case PC, SMBIOS Reserved, CompactPCI, AdvancedTCA, Blade Enclosure, DMTF Reserved, Vendor Reserved} Values for the definition types are: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, .., 0x8000..0xFFFF}	Will be set to 17 (Main System Chassis)
ChassisTypeDescription	string	The ChassisTypeDescription is a string providing more information on the ChassisPackageType.	Will have appropriate description.
Description	string	Textual description of the object.	Will have appropriate description.

TABLE A-3 Properties for Oracle_Chassis *(Continued)*

Property	Data Type	Description	ILOM Value
ElementName	string	The ElementName property is a user-friendly name. This property allows each instance to define a user-friendly name in addition to its key properties, identity data, and description information. **Note -** The Name property of ManagedSystemElement is also defined as a user-friendly name. But, it is often subclassed to be a *Key*. It is not reasonable that the same property can convey both identity and a user-friendly name, without inconsistencies. Where Name exists and is *not a Key* (such as for instances of LogicalDevice), the same information can be present in both the Name and ElementName properties.	Set to component NAC name.

TABLE A-3 Properties for Oracle_Chassis *(Continued)*

Property	Data Type	Description	ILOM Value
HealthState	uint16[]	Indicates the current health of the element. This attribute expresses the health of this element but not necessarily that of its subcomponents. The following values apply: • 0 (Unknown) - The implementation cannot report on HealthState at this time. • 5 (OK) - The element is fully functional and is operating within normal operational parameters and without error. • 10 (Degraded/Warning) - The element is in working order and all functionality is provided. However, the element is not working to the best of its abilities. For example, the element might not be operating at optimal performance or it might be reporting recoverable errors. • 15 (Minor Failure) - All functionality is available but some might be degraded. • 20 (Major Failure) - The element is failing. It is possible that some or all of the functionality of this component is degraded or not working. • 25 (Critical Failure) - The element is non-functional and recovery might not be possible. • 30 (Non-recoverable Error) - The element has completely failed, and recovery is not possible. All functionality provided by this element has been lost. DMTF has reserved the unused portion of the continuum for additional health states in the future.	Will have appropriate value depending on whether the component is in error state or not.
Manufacturer	string	The Manufacturer property is the name of the organization responsible for producing the PhysicalElement. This organization might be the entity from whom the Element is purchased, but this is not necessarily true. The latter information is contained in the vendor property of CIM_Product.	Will have appropriate value if the chassis is considered a FRU by the platform.
Model	string	The Model property is the name by which the PhysicalElement is generally known.	Will have appropriate value if the chassis is considered a FRU by the platform.

TABLE A-3 Properties for Oracle_Chassis *(Continued)*

Property	Data Type	Description	ILOM Value
OperationalStatus	uint16[]	The OperationalStatus property indicates the current statuses of the element. Various operational statuses are defined. Many of the enumeration's values are self-explanatory. Enumeration definitions can include any of the following: {Unknown, Other, OK, Degraded, Stressed, Predictive Failure, Error, Non-Recoverable Error, Starting, Stopping, Stopped, In Service, No Contact, Lost Communication, Aborted, Dormant, Supporting Entity in Error, Completed, Power Mode, DMTF Reserved, Vendor Reserved} Values for the above definition are as follows: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, .., 0x8000..}	OperationalStatus[0] will have appropriate value depending on whether the component is in error state or not.
PartNumber	string	Part number assigned by the organization that is responsible for producing or manufacturing the PhysicalElement	Will have appropriate value if the chassis is considered a FRU by the platform.
SKU	string	The SKU property is the stock-keeping unit number for this PhysicalElement.	Will have appropriate value if the chassis is considered a FRU by the platform.
SerialNumber	string	The SerialNumber property is a manufacturer-allocated number used to identify the physical element.	Will have appropriate value if the chassis is considered a FRU by the platform.
StatusDescriptions	string	That StatusDescriptions property describes the various OperationalStatus array values. For example, if Stopping is the value assigned to OperationalStatus, then this property may contain an explanation as to why an object is being stopped. Note that entries in this array are correlated with those at the same array index in OperationalStatus.	StatusDescriptions[*0*] will have appropriate description on the reason for the value of OperationalStatus[*0*]

Oracle_ComputerSystem

Description:	The `Oracle_ComputerSystem` class represents a special collection of Sun system managed elements. This collection provides computer capabilities and serves as an aggregation point to associate one or more of the following elements: file system, operating system, processor and memory (volatile and non-volatile storage).
Inheritance:	`CIM_ComputerSystem`
Properties	For a description of the supported properties for the `Oracle_ComputerSystem` class, see TABLE A-4**.** **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-4**)**, see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profiles	• Base Server • Service Processor

TABLE A-4 Attributes for Oracle_ComputerSystem

Property	Data Type	Description	ILOM Value
`CreationClass Name`	string	The `CreationClassName` property is a mandatory *key* property. `CreationClassName` indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	Set value to: `Oracle_ComputerSystem`.
`Name`	string	The `Name` attribute is a mandatory *key* CIM property. The inherited `Name` serves as the key of a system instance in an enterprise environment.	Implementation-dependent value representing unique ID of the `ComputerSystem`.

TABLE A-4 Attributes for Oracle_ComputerSystem *(Continued)*

Property	Data Type	Description	ILOM Value
Dedicated[]	string	The Dedicated[] property enumerates the purpose(s) to which the ComputerSystem is dedicated, if any, and what functionality is provided. Functionality definitions can include any of the following: {Not Dedicated, Unknown, Other, Storage, Router, Switch, Layer 3 Switch, Central Office Switch, Hub, Access Server, Firewall, Print, I/O, Web Caching, Management, Block Server, File Server, Mobile User Device, Repeater, Bridge/Extender, Gateway, Storage Virtualizer, Media Library, ExtenderNode, NAS Head, Self-contained NAS, UPS, IP Phone, Management Controller, Chassis Manager, Host-based RAID controller, Storage Device Enclosure, Desktop, Laptop, Virtual Tape Library, Virtual Library System, DMTF Reserved, Vendor Reserved} Values for the above functionality definitions are as follows: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36..32567, 32568..65535}	For ComputerSystem instance representing the controller, the Dedicated[0] value will be set to 28 (Management Controller). For ComputerSystem instance representing the controllee, Dedicated[0] to 0 (Not Dedicated).
ElementName	string	The ElementName property is a user-friendly name. This property allows each instance to define a user-friendly name in addition to its key properties, identity data, and description information. **Note -** The Name property of ManagedSystemElement is also defined as a user-friendly name. But, it is often subclassed to be a *Key*. It is not reasonable that the same property can convey both identity and a user-friendly name, without inconsistencies. Where Name exists and is not a key (such as for instances of LogicalDevice), the same information can be present in both the Name and ElementName properties.	For ComputerSystem instance representing the controller, the ElementName will be set to the controller or hostname. For ComputerSystem instance representing the controllee, the ElementName will be set to the host product name.

TABLE A-4 Attributes for Oracle_ComputerSystem *(Continued)*

Property	Data Type	Description	ILOM Value
`EnabledDefault`	string	The `EnabledDefault` property is an enumerated value indicating an administrator's default or startup configuration for the Enabled State of an element. By default, the element is `Enabled` (value=2). Element definitions include any of the following: {Enabled, Disabled, Not Applicable, Enabled but Offline, No Default, Quiesce, DMTF Reserved, Vendor Reserved} Values for the element definitions are as follows: {2, 3, 5, 6, 7, 9, .., 32768..65535}	`EnabledDefault` will be set to default value `2` (`Enabled`)

TABLE A-4 Attributes for Oracle_ComputerSystem *(Continued)*

Property	Data Type	Description	ILOM Value
EnabledState	uint16[]	EnabledState is an integer enumeration that indicates the enabled and disabled states of an element. It can also indicate the transitions between these requested states. For example, Shutting Down (value=4) and Starting (value=10) are transient states between enabled and disabled. The following text briefly summarizes the various enabled and disabled states: • Enabled (2) indicates that the element is or could be executing commands, will process any queued commands, and queues new requests. • Disabled (3) indicates that the element will not execute commands and will drop any new requests. • Shutting Down (4) indicates that the element is in the process of going to a Disabled state. • Not Applicable (5) indicates the element does not support being enabled or disabled. • Enabled but Offline (6) indicates that the element might be completing commands, and will drop any new requests. • Test (7) indicates that the element is in a test state. • Deferred (8) indicates that the element might be completing commands, but will queue any new requests. • Quiesce (9) indicates that the element is enabled but in a restricted mode. • Starting (10) indicates that the element is in the process of going to an Enabled state. New requests are queued. The following values apply: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11..32767, 32768..65535} Value definitions include: {Unknown, Other, Enabled, Disabled, Shutting Down, Not Applicable, Enabled but Offline, In Test, Deferred, Quiesce, Starting, DMTF Reserved, Vendor Reserved}	For ComputerSystem instance representing the controller, the EnabledState value will be set to 2 (Enabled). For ComputerSystem instance representing the controllee, the EnabledState value will be set to appropriate value depending on the power state of the controllee.

TABLE A-4 Attributes for Oracle_ComputerSystem *(Continued)*

Property	Data Type	Description	ILOM Value
HealthState	uint16[]	Indicates the current health of the element. This attribute expresses the health of this element but not necessarily that of its subcomponents. The following values apply: • 0 (Unknown) - The implementation cannot report on HealthState at this time. • 5 (OK) - The element is fully functional and is operating within normal operational parameters and without error. • 10 (Degraded/Warning) - The element is in working order and all functionality is provided. However, the element is not working to the best of its abilities. For example, the element might not be operating at optimal performance or it might be reporting recoverable errors. • 15 (Minor Failure) - All functionality is available but some might be degraded. • 20 (Major Failure) - The element is failing. It is possible that some or all of the functionality of this component is degraded or not working. • 25 (Critical Failure) - The element is non-functional and recovery might not be possible. • 30 (Non-recoverable Error) - The element has completely failed, and recovery is not possible. All functionality provided by this element has been lost. DMTF has reserved the unused portion of the continuum for additional HealthStates in the future.	For ComputerSystem instance representing the controller EnabledState will be set to 5 (OK). For ComputerSystem instance representing the controllee HealthState will be set to appropriate value depending on the value of OperationalStatus property.
IdentifyingDescriptions	string	The IdentifyingDescriptions property is an array of free-form strings providing explanations and details behind the entries in the OtherIdentifyingInfo array. **Note -** Each entry of this array is related to the entry in OtherIdentifyingInfo that is located at the same index.	For ComputerSystem instance representing the controller, the IdentifyingDescriptions will not be set. For ComputerSystem instance representing the controllee, the IdentifyingDescriptions will be set to the following value: "CIM:Model:SerialNumber".

TABLE A-4 Attributes for Oracle_ComputerSystem *(Continued)*

Property	Data Type	Description	ILOM Value
OperationalStatus	uint16[]	The OperationalStatus indicates the current statuses of the element. Various operational statuses are defined. Many of the enumeration's values are self-explanatory. However, a few are not and are described here in more detail in the CIM_ComputerSystem.mof described in the DMTF CIM schema v2.18.1 Element definitions include any of the following: {Unknown, Other, OK, Degraded, Stressed, Predictive Failure, Error, Non-Recoverable Error, Starting, Stopping, Stopped, In Service, No Contact, Lost Communication, Aborted, Dormant, Supporting Entity in Error, Completed, Power Mode, DMTF Reserved, Vendor Reserved} Values for the above definitions are: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, .., 0x8000..}	For ComputerSystem instance representing the controller, the OperationalStatus[0] will be set to 2 (OK). For ComputerSystem instance representing the controllee, the OperationalStatus[0] will be set to appropriate value depending on the power status (starting, stopping) or whether the host has incurred an error or is unknown.
OtherEnabledState	string	The OtherEnabledState property is a string that describes the enabled or disabled state of the element when the EnabledState property is set to 1 (Other). This property must be set to null when EnabledState is any value other than 1.	Will be set to empty string.
OtherIdentifyingInfo	Sting[]	The OtherIdentifyingInfo property captures additional data, beyond System Name information, that could be used to identify a ComputerSystem. One example would be to hold the Fibre Channel World-Wide Name (WWN) of a node. Note that if only the Fibre Channel name is available and is unique (able to be used as the System key), then this property would be NULL and the WWN would become the System key, its data placed in the Name property.	For ComputerSystem instance representing the controller, the OtherIdentifyingInfo will not be set. For ComputerSystem instance representing the controllee, the OtherIdentifyingInfo[0] will be set to the following value: <product-name>:<SerialNumber> For more details, refer to the DMTF Base Server Profile.

TABLE A-4 Attributes for Oracle_ComputerSystem *(Continued)*

Property	Data Type	Description	ILOM Value
RequestedState	uint16[]	The RequestedState property is an integer enumeration that indicates the last requested or desired state for the element, irrespective of the mechanism through which it was requested. The actual state of the element is represented by EnabledState. This property is provided to compare the last requested and current enabled or disabled states. Element definitions include any of the following: {Unknown, Enabled, Disabled, Shut Down, No Change, Offline, Test, Deferred, Quiesce, Reboot, Reset, Not Applicable, DMTF Reserved, Vendor Reserved} Values for the above definitions are as follows: {0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, .., 32768..65535} **Note -** When EnabledState is set to 5 (Not Applicable), then this property has no meaning.	Will set to Not Applicable if there has not been any invocation of RequestStateChange() method. Will be set to the correct value of the incoming argument of RequestStateChange().
RequestStateChange()	UInt32	Method for client to request state change. The following state change operations are as follows: {2 , 3, 4, 6, 7, 8, 9, 10, 11, .., 32768..65535"} Definitions for the above values are as follows: {Enabled, Disabled, Shut Down, Offline,Test, Defer, Quiesce, eboot, Reset, DMTF Reserved, Vendor Reserved"}	For ComputerSystem instance representing the controller, will support 11 (Reset). For ComputerSystem instance representing the controllee, will support 2 (Enabled), 3 (Disabled), 4 (Shut Down). This operation is only supported if the user has admin role.

Oracle_ComputerSystemPackage

Description:	The `Oracle_ComputerSystemPackage` class is used to associate the instance of `Oracle_ComputerSystem` representing the controllee to the physical package chassis, `Oracle_Chassis`, which realizes the `Oracle_ComputerSystem`.
Inheritance:	`CIM_ComputerSystemPackage`
Properties:	For a description of the supported properties for the `Oracle_ComputerSystemPackage` class, see TABLE A-5. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-5), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	• Physical asset • Base server

TABLE A-5 Properties for Oracle_ComputerSystemPackage

Property	Data Type	Description	ILOM Value
`Antecedent`	`Oracle_Chassis` REF	The `Antecedent` property is a mandatory *key* property. The chassis that realizes a `Oracle_ComputerSystem`.	Object path to an instance of `Oracle_Chassis`.
`Dependent`	`Oracle_ComputerSystem` REF	The `Dependent` property is a mandatory *key* property. Represents the `Oracle_ComputerSystem`.	Object path to the instance of `Oracle_ComputerSystem` representing the controllee.

Oracle_Container

Description:	`Oracle_Container` is used to associate a physical package (`CIM_PhysicalPackage`) and a physical element (`CIM_PhyscalElement`) contained in the physical package.
Inheritance:	`CIM_Container`
Properties:	For a description of the supported properties for the `Oracle_Container` class, see TABLE A-6. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-6), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Physical asset

TABLE A-6 Properties for Oracle_Container

Property	Data Type	Description	ILOM Value
`GroupComponent`	`CIM_PhysicalPackage` REF	The `GroupComponent` property is a mandatory *key* property. The PhysicalPackage that contains other physical elements, including other packages.	Object path to an instance of `CIM_PhysicalPackage`.
`PartComponent`	`CIM_PhysicalElement` REF	The `PartComponent` property is a mandatory *key* property. The `PhysicalElement` which is contained in the package.	Object path to an instance of `CIM_ PhysicalElement`.

Oracle_ElementCapabilities

Description:	The `Oracle_ElementCapabilities` class is used to associate an instance of `ManagedElements` and its capabilities.
Inheritance:	`CIM_ElementCapabilities`
Properties:	For a description of the supported properties for the `Oracle_ElementCapabilities` class, see TABLE A-7. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-7), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Base server

TABLE A-7 Properties for Oracle_ElementCapabilities

Property	Data Type	Description	ILOM Value
`Capabilities`	`CIM_Capabilities REF`	The `Capabilities` property is a mandatory *key* property. The `Capabilities` object that is associated with the element.	Object path to an instance of `Oracle_EnabledLogicalElementCapabilities`.
`ManagedElement`	`CIM_ManagedElement REF`	The `ManagedElement` property is a mandatory *key* property. Identifies the managed element.	Object path to an instance of `Oracle_ComputerSystem`.

Oracle_ElementConformsToProfile

Description:	`Oracle_ElementConformsToProfile` associates the instance of `Oracle_ComputerSystem` representing the controllee to the instance of `Oracle_RegisteredProfile` representing the Base Server Profile.
Inheritance:	`CIM_ElementConformsToProfile`
Properties:	For a description of the supported properties for the `Oracle_ElementConformsToProfile` class, see TABLE A-8. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-8), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Profile registration Base server

TABLE A-8 Properties for Oracle_ElementConformsToProfile

Property	Data Type	Description	ILOM Value
`ElementConformsToProfile`	`Oracle_RegisteredProfile` REF	The `ElementconformsToProfile` property is a mandatory *key* property. The `RegisteredProfile` to which the `ManagedElement` conforms.	Object path to the instance of `Oracle_RegisteredProfile`.
`ManagedElement`)	`Oracle_ComputerSystem` REF	The `ManagedElement` property is a mandatory *key* property. The `Oracle_ComputerSystem`.	Object path to the instance of `Oracle_ComputerSystem` representing the controllee.

Oracle_EnabledLogicalElementCapabilities

Description:	`EnabledLogicalElementCapabilities` describes the capabilities supported for changing the state of the associated `EnabledLogicalElement`.
Inheritance:	`CIM_EnabledLogicalElementCapabilities`
Properties:	For a description of the supported properties for the `Oracle_EnabledLogicalElementCapabilities` class, see TABLE A-9. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-9), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Base server

TABLE A-9 Properties for Oracle_EnabledLogicalElementCapabilities

Property	Data Type	Description	ILOM Value
InstanceID	string	The InstanceID property is a mandatory *key* property. Within the scope of the instantiating Namespace, the InstanceID property uniquely identifies an instance of this class. The value of InstanceID should be constructed using the following preferred algorithm: <OrgID>:<LocalID> Where: • <OrgID> and <LocalID> are separated by a colon ':' • <OrgID> must include a copyrighted, trademarked or otherwise unique name that is owned by the business entity creating or defining the InstanceID, or is a registered ID that is assigned to the business entity by a recognized global authority (This is similar to the <Schema Name>_<Class Name> structure of Schema class names). • <OrgID> must not contain a colon (':'). The first colon to appear in InstanceID must appear between <OrgID> and <LocalID> . • <LocalID> is chosen by the business entity and should not be re-used to identify different underlying (real-world) elements. • For DMTF defined instances, the <OrgID> must be set to 'CIM'. If the above preferred algorithm is not used, the defining entity must ensure that the resultant InstanceID is not re-used across any instance IDs produced by this or other providers for this instance's NameSpace.	Represents the unique ID of the EnabledLogicalElementCapabilities.
Description	string	Textual description of the object.	Appropriate descriptions.
ElementName	string	The ElementName property is a user-friendly name. This property allows each instance to define a user-friendly name in addition to its key properties, identity data, and description information. Note that the Name property of ManagedSystemElement is also defined as a user-friendly name. But, it is often subclassed to be a Key. It is not reasonable that the same property can convey both identity and a user-friendly name, without inconsistencies. Where Name exists and is not a Key (such as for instances of LogicalDevice), the same information can be present in both the Name and ElementName properties.	Appropriate value.

TABLE A-9 Properties for Oracle_EnabledLogicalElementCapabilities *(Continued)*

Property	Data Type	Description	ILOM Value
`ElementNameEditSupported`	boolean	The boolean indicates whether the `ElementName` can be modified.	Set to False.
`RequestedStatesSupported`	uint16[]	Indicates the possible states that can be requested when using the method `RequestStateChange` on the `EnabledLogicalElement`. The following values apply: {2, 3, 4, 6, 7, 8, 9, 10, 11} Definitions for the above values are as follows: {Enabled, Disabled, Shut Down, Offline, Test, Defer, Quiesce, Reboot, Reset}	For the `EnabledLogicalElementCapabilities` instance representing the controller, `RequestedStatesSupported`[0] will be set to `11` (Reset). For the `EnabledLogicalElementCapabilities` instance representing the controllee, `RequestedStatesSupported`[] will be set to `2` (Enabled), `3` (Disabled), or `4` (Shut Down).

Oracle_HWCompErrorOkIndication

Description:	When a client creates an indication subscription in which the filter indicates that it looks for `CIM_InstModification` in which the modified object is a `PhysicalElement` (that is the query statement is `SourceInstance` ISA `CIM_PhysicalElement`), and it looks for changes in `SourceInstance.OperationalStatus` or `SourceInstance.HealthState` then ILOM CIM-subsystem will generate `Oracle_HWCompErrorOkIndication` indication when it is noticed that a hardware component goes from good to bad, or vice versa.
Inheritance:	`CIM_InstModification`
Properties:	For a description of the supported properties for the `Oracle_HWCompErrorOkIndication` class, see TABLE A-10. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-10), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	None

TABLE A-10 Properties for Oracle_HWCompErrorOkIndication

Property	Data Type	Description	ILOM Value
`PreviousInstance`	string	A copy of the previous instance whose change generated the Indication. `PreviousInstance` contains older values of an instance's properties (as compared to `SourceInstance`), selected by the `IndicationFilter`'s query.	String representation of the previous instance of `CIM_PhysicalElement` that is affected.
`SensorObjectPath` (SUN-specific)	string	Object path of the sensor that causes the hardware component to change operational state.	Appropriate value.

TABLE A-10 Properties for Oracle_HWCompErrorOkIndication *(Continued)*

Property	Data Type	Description	ILOM Value
SourceInstance	string	A copy of the instance that changed to generate the Indication. SourceInstance contains the current values of the properties selected by the indication filter's query. In the case of CIM_InstDeletion, the property values are copied before the instance is deleted.	String representation of the instance of CIM_PhysicalElement that is affected.
SourceInstanceHost	string	The host name or IP address of the SourceInstance.	Will have the value Oracle_ComputerSystem.Name of the instance of Oracle_ComputerSystem representating the controllee.
SourceInstanceModelPath	string	The model path of the SourceInstance. The following format *must* be used to encode the model path: <NamespacePath>:<ClassName>.<Prop1>=<Value1>, <Prop2>=\ "<Value2>, ..	String representation of the object path of the SourceInstance.

Oracle_IndicatorLED

Description:	The `Oracle_IndicatorLED` class models the logical aspects of an indicator LED.
Inheritance:	`CIM_IndicatorLED`
Properties:	For a description of the supported properties for the `Oracle_IndicatorLED` class, see TABLE A-11. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-11), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Indicator LED

Note – The Indicator LED profile specifies the `CIM_IndicatorLED` class. However, the `CIM_IndicatorLED` class does not exist in version 2.18.1 of the CIM Schema. Therefore, Oracle uses the `CIM_IndicatorLED` class that is identified in the Experimental CIM Schema version 2.18.1 and re-named it to `Oracle_IndicatorLED`.

TABLE A-11 Properties for Oracle_IndicatorLED

Property	Data Type	Description	ILOM Value
`CreationClassName`	string	The `CreationClassName` property is a mandatory *key* property. `CreationClassName` indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	Set to `Oracle_IndicatorLED`.
`DeviceID`	string	The `DeviceID` property is a mandatory *key* property. An address or other identifying information used to uniquely name the `LogicalDevice`.	Set to the NAC name of the of the LED.
`SystemCreationClassName`	string	The `SystemCreationClassName` property is a mandatory *key* property. The `SystemCreationClassName` of the scoping system.	Set to `Oracle_ComputerSystem`.

TABLE A-11 Properties for Oracle_IndicatorLED *(Continued)*

Property	Data Type	Description	ILOM Value
SystemName	string	The system name of the scoping system.	Will be set to Oracle_ComputerSystem.Name of the instance of Oracle_ComputerSystem that represents the controllee.
ActivationState	uint16[]	Indicates the current activity of an LED. An LED may exhibit behaviors that vary greatly in complexity. If the behavior of the LED is simple or a detailed depiction of the behavior is unnecessary to convey to client applications, values other than 5 (ControlPattern) may be used to indicate the behavior. If the behavior is complex and detailed information about the behavior is meaningful to a client, the value 5 (ControlPattern) can be used to indicate the ControlPattern property that describes the behavior. 2 (Lit) shall indicate the LED is continuously illuminated without variation in color or intensity. 3 (Blinking) shall indicate the LED is alternating between illuminated and not illuminated in a regular pattern without variation in color or intensity. The pattern is not indicated. 4 (Off) shall indicate the LED is not illuminated. 5 (ControlPattern) shall indicate the LED is exhibiting behavior that is described using the ControlPattern property. The following values apply: {2, 3, 4, 5, .., 32768..65535} Definitions for the above values are as follows: {Lit, Blinking, Off, Control Pattern, DMTF Reserved, Vendor Reserved}	Appropriate value.
Color	uint16[]	Indicates the current color of the LED. If the value of the ActivationState property is 4 (Off) this property will indicate the color of the LED the last time it was lit, or it will have the value 2 (Not Applicable).	Appropriate value.

TABLE A-11 Properties for Oracle_IndicatorLED *(Continued)*

Property	Data Type	Description	ILOM Value
ControlMode	uint16[]	Indicates the current control mode for the LED. 2 (Automatic) shall indicate the state of the LED is being controlled by the management infrastructure. 3 (Manual) shall indicate the state of the LED is being controlled by a management client. 4 (Test) shall indicate the LED is in a test mode. The following values apply: {2, 3, 4, .., 32768..65535} Definitions for the above values are as follows: {Automatic, Manual, Test, DMTF Reserved, Vendor Reserved}	Appropriate value.
ControlPattern	string	An LED may exhibit a range of behavior from very simple (ex. solid on) to very complicated (for example, a series of blinks of alternating color and duration). ControlPattern specifies the vendor or standard behavior exhibited by the LED if it cannot be described using one of the standard behaviors listed for the ActivationState property. If ActivationState has the value 5 (ControlPattern), the ControlPattern property shall not be NULL. The value of ControlPattern should be constructed using the following preferred algorithm: <OrgID>::<Pattern> Where: • <OrgID> and < Pattern> are separated by two colons (::) • <OrgID> includes a copyrighted, trademarked, or otherwise unique name that is owned by the business entity that is creating or defining the ControlPattern or that is a registered ID assigned to the business entity by a recognized global authority • If the definition of the value is specified by the DMTF, the value of <OrgID> must be "DMTF" • <Pattern> is chosen by the business entity and should not be reused to identify different underlying (real-world) behaviors. If the behavior specified for the LED adheres to a standard or proprietary specification, <Pattern> should be a uniquely assigned value identifying the behavior. If the behavior for the LED is described using a standard or proprietary grammar, <Pattern> should be prefixed with a uniquely assigned identifier for the grammar.	Appropriate value.

TABLE A-11 Properties for Oracle_IndicatorLED *(Continued)*

Property	Data Type	Description	ILOM Value
ElementName	string	Specifies an identifier for the LED. The value of ElementName should be constructed using the following preferred algorithm: <OrgID>::<LocalID> Where: • <OrgID> and <LocalID> are separated by two colons (::) • <OrgID> includes a copyrighted, trademarked, or otherwise unique name that is owned by the business entity that is creating or defining the ControlPattern or that is a registered ID assigned to the business entity by a recognized global authority • <LocalID> is chosen by the business entity and should not be reused to identify different underlying (real-world) elements.	Set to the NAC name of the LED.
EnabledDefault	uint16[]	An enumerated value indicating an administrator's default or startup configuration for the enabled state of an element. By default, the element is Enabled (value=2). Valid values are as follows: {2, 3, 5, 6, 7, 9, .., 32768..65535} Definitions for the valid values are: {Enabled, Disabled, Not Applicable, Enabled but Offline, No Default, Quiesce, DMTF Reserved, Vendor Reserved}	Set to default value 2 (Enabled).

TABLE A-11 Properties for Oracle_IndicatorLED *(Continued)*

Property	Data Type	Description	ILOM Value
EnabledState	uint16[]	Integer enumeration that indicates the enabled and disabled states of an element. It can also indicate the transitions between these requested states. For example, Shutting Down (value=4) and Starting (value=10) are transient states between enabled and disabled. The following values apply: • 0 (Unknown) • 1 (Other) • 2 (Enabled) - The element is or could be executing commands, will process any queued commands, and queues new requests. • 3 (Disabled) - The element will not execute commands and will drop any new requests • 4 (Shutting Down) - The element is in the process of going to a Disabled state. • 5 (Not Applicable) - The element does not support being enabled or disabled. • 6 (Enabled but Offline) - The element might be completing commands, and will drop any new requests. • 7 (Test) - The element is in a test state. • 8 (Deferred) - The element might be completing commands, but will queue any new requests. • 9 (Quiesce) - The element is enabled but in a restricted mode. • 10 (Starting) - The element is in the process of going to an Enabled state. New requests are queued. • 11..32767 (DMTF Reserved) • 32768..65539 (Vendor Reserved)	Appropriate value.

TABLE A-11 Properties for Oracle_IndicatorLED *(Continued)*

Property	Data Type	Description	ILOM Value
`HealthState`	uint16[]	Indicates the current health of the element. This attribute expresses the health of this element but not necessarily that of its subcomponents. The following values apply: • 0 (Unknown) - The implementation cannot report on `HealthState` at this time. • 5 (OK) - The element is fully functional and is operating within normal operational parameters and without error. • 10 (Degraded/Warning) - The element is in working order and all functionality is provided. However, the element is not working to the best of its abilities. For example, the element might not be operating at optimal performance or it might be reporting recoverable errors. • 15 (Minor Failure) - All functionality is available but some might be degraded. • 20 (Major Failure) - The element is failing. It is possible that some or all of the functionality of this component is degraded or not working. • 25 (Critical Failure) - The element is non-functional and recovery might not be possible. • 30 (Non-recoverable Error) - The element has completely failed, and recovery is not possible. All functionality provided by this element has been lost. DMTF has reserved the unused portion of the continuum for additional `HealthStates` in the future.	Appropriate value.

TABLE A-11 Properties for Oracle_IndicatorLED *(Continued)*

Property	Data Type	Description	ILOM Value
IndicatedConditions	uint16[]	The condition indicated by the LED. The following values apply: • 2 (Not Applicable) - The LED is currently not assigned an interpretation. • 3 (Location) - The LED is used to indicate the location of associated managed elements. • 4 (Attention) - The LED is used to indicate the associated managed elements requires the attention of service personnel. • 5 (Activity) -The LED is used to indicate activity is occurring for the associated managed elements. The type of activity indicated is specific to the associated managed elements. • 6 (Powered On) - The LED is used to indicate if the associated managed elements are receiving power. • 7 (Fault) - The LED is used to indicate if the associated managed elements are in a fault, error, or otherwise degraded state.	Appropriate value.

TABLE A-11 Properties for Oracle_IndicatorLED *(Continued)*

Property	Data Type	Description	ILOM Value
`OperationalStatus`	uint16[]	The `OperationalStatus` property indicates the current statuses of the element. Various operational statuses are defined. Many of the enumeration's values are self-explanatory. Enumeration values can include any of the following: {Unknown, Other, OK, Degraded, Stressed, Predictive Failure, Error, Non-Recoverable Error, Starting, Stopping, Stopped, In Service, No Contact, Lost Communication, Aborted, Dormant, Supporting Entity in Error, Completed, Power Mode, DMTF Reserved, Vendor Reserved} A list of valid values for the enumeration values include: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, .., 0x8000..}	Appropriate value.
`OtherIndicatedConditionDescription`	string	This property will have a value if `IndicatedCondition` contains the value `1` (Other).	Will have appropriate value if `IndicatedCondition` contains the value `1` (Other).
`RequestedState`	uint16[]	The `RequestedState` property is an integer enumeration that indicates the last requested or desired state for the element, irrespective of the mechanism through which it was requested. The actual state of the element is represented by `EnabledState`. This property is provided to compare the last requested and current enabled or disabled states. Element definitions include any of the following: {Unknown, Enabled, Disabled, Shut Down, No Change, Offline, Test, Deferred, Quiesce, Reboot, Reset, Not Applicable, DMTF Reserved, Vendor Reserved} Values for the above definitions include: {0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, .., 32768..65535} **Note -** When `EnabledState` is set to `5` ("Not Applicable"), then this property has no meaning.	Set to `12` (Not Applicable).

Oracle_InstCreation

Description:	When client creates an indication subscription in which the filter indicates that it looks for `CIM_InstCreation` and `SourceInstance` is a `PhysicalElement` (for example, the query statement contains `SourceInstance` ISA `CIM_PhysicalElement`) then ILOM CIM-subsystem will generate `Oracle_InstCreation` indication when it notices that a hardware component is hot inserted into the chassis.
Inheritance:	`CIM_InstCreation`
Properties:	For a description of the supported properties for the `Oracle_InstCreation`, see TABLE A-12. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-12), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	None

TABLE A-12 Properties for Oracle_InstCreation

Property	Data Type	Description	ILOM Value
`SourceInstance`	string	A copy of the instance that changed to generate the Indication. `SourceInstance` contains the current values of the properties selected by the Indication Filter's Query. In the case of `CIM_InstDeletion`, the property values are copied before the instance is deleted.	String representation of the instance of `CIM_PhysicalElement` that is hot-inserted.
`SourceInstance Host`	string	The host name or IP address of the `SourceInstance`.	Will have the value `Oracle_ComputerSystem.Name` of the instance of `Oracle_ComputerSystem` representing the controllee.
`SourceInstance ModelPath`	string	The model path of the `SourceInstance`. The following format must be used to encode the model path: <NamespacePath>:<ClassName>.<Prop1>=<Value1>, <Prop2>=\"<Value2>, ...	String representation of the object path of the `SourceInstance`.

Oracle_InstDeletion

Description:	When client creates an indication subscription in which the filter indicates that it looks for `CIM_InstDeletion` and `SourceInstance` is a `PhysicalElement` (for example, the query statement contains `SourceInstance` ISA `CIM_PhysicalElement`) then ILOM CIM-subsystem will generate `Oracle_InsDeletion` indication when it is noticed that a hardware component is hot removed from the chassis.
Inheritance:	`CIM_InstDeletion`
Properties:	For a description of the supported properties for the `Oracle_InstDeletion` class, see TABLE A-13. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-13), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	None

TABLE A-13 Properties for Oracle_InstDeletion

Property	Data Type	Description	ILOM Value
`SourceInstance`	string	A copy of the instance that changed to generate the Indication. `SourceInstance` contains the current values of the properties selected by the indication filter's query. In the case of `CIM_InstDeletion`, the property values are copied before the instance is deleted.	String representation of the instance of `CIM_PhysicalElement` that is hot-removed.
`SourceInstance Host`	string	The host name or IP address of the `SourceInstance`.	Will have the value `Oracle_ComputerSystem.Name` of the instance of `Oracle_ComputerSystem` representing the controllee.
`SourceInstance ModelPath`	string	The model path of the `SourceInstance`. The following format must be used to encode the model path: <NamespacePath>:<ClassName>.<Prop1>=<Value1>, <Prop2>=\"<Value2>, ...	String representation of the object path of the `SourceInstance`.

Oracle_LogEntry

Description:	`Oracle_LogEntry` is used to represent individual log records of IPMI SEL log.
Inheritance:	`CIM_LogEntry`
Properties:	For a description of the supported properties for the `Oracle_LogEntry` class, see TABLE A-14. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-14), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Record log

TABLE A-14 Properties for Oracle_LogEntry

Property	Data Type	Description	ILOM Value
InstanceID	string	The InstanceID property is a mandatory *key* property. Within the scope of the instantiating Namespace, InstanceID uniquely identifies an instance of this class. In order to ensure uniqueness within the NameSpace, the value of InstanceID should be constructed using the following preferred algorithm: <OrgID>:<LocalID> Where: • <OrgID> and <LocalID> are separated by a colon ':' • <OrgID> You must include a copyrighted, trademarked, or otherwise the unique name that is owned by the business entity creating or defining the InstanceID, or is a registered ID that is assigned to the business entity by a recognized global authority (This is similar to the <Schema Name>_<Class Name> structure of Schema class names.) • <OrgID> must not contain a colon (':'). When using this algorithm, the first colon to appear in InstanceID must appear between <OrgID> and <LocalID>. • <LocalID> is chosen by the business entity and should not be re-used to identify different underlying (real-world) elements. • For DMTF defined instances, the preferred algorithm must be used with the <OrgID> set to 'CIM' If the above preferred algorithm is not used, the defining entity must ensure that the resultant InstanceID is not re-used across any InstanceIDs produced by this or other providers for this instance's NameSpace.	Implementation dependent value representing unique ID.
CreationTimeStamp	datetime	A LogEntry may include a timestamp for the entry.	Appropriate value.
Description	string	Textual description of the object.	SEL event description.

TABLE A-14 Properties for Oracle_LogEntry *(Continued)*

Property	Data Type	Description	ILOM Value
ElementName	string	The ElementName property is a user-friendly name. This property allows each instance to define a user-friendly name in addition to its key properties, identity data, and description information. **Note -** The Name property of ManagedSystemElement is also defined as a user-friendly name. But, it is often subclassed to be a Key. It is not reasonable that the same property can convey both identity and a user-friendly name, without inconsistencies. Where Name exists and is not a Key (such as for instances of LogicalDevice), the same information can be present in both the Name and ElementName properties.	SEL event record ID.
LogInstance ID	string	The string containing the log's InstanceID.	Implementation dependent value representing unique ID of the associated Oracle_RecordLog.
LogName	string	The string containing the log's Name. This property is available for backwards continuity with CIM_LogRecord.	Will have the value "SEL Log".
RecordData	string	A string containing LogRecord data. If the corresponding RecordFormat property is <empty>, or cannot be parsed according to the recommended format, RecordData should be interpreted as a free-form string. If the RecordFormat property contains parseable format information (as recommended in the RecordFormat Description qualifier), the RecordData string should be parsed in accordance with this format. In this case, RecordData should begin with the delimiter character and this character should be used to separate substrings in the manner described. The RecordData string can then be parsed by the data consumer and appropriately typed.	Contents of the SEL event data.

TABLE A-14 Properties for Oracle_LogEntry *(Continued)*

Property	Data Type	Description	ILOM Value
RecordFormat	string	A string describing the data structure of the information in the property, RecordData. If the RecordFormat string is <empty>, RecordData should be interpreted as a free-form string. To describe the data structure of RecordData, the RecordFormat string should be constructed as follows: • The first character is a delimiter character and is used to parse the remainder of the string into sub-strings. • Each sub-string is separated by the delimiter character and should be in the form of a CIM property declaration (for example, datatype and property name). This set of declarations may be used to interpret the similarly delimited RecordData property. For example, using a '*' delimiter: RecordFormat = *"*string ThisDay*uint32 ThisYear*datetime SomeTime"* may be used to interpret: RecordData = "**This is Friday*2002*20020807141000.000000-300*".	Will have the format used for interpreting the RecordData property.
RecordID	string	Provides a representation of log entry ordering or pointers and handles for log entries	SEL event record ID.

Oracle_LogManagesRecord

Description:	`Oracle_LogManagesRecord` is used to associate the instance of `Oracle_RecordLog` representing the IPMI SEL log to an instance of SEL log record.
Inheritance:	`CIM_LogManagesRecord`
Properties:	For a description of the supported properties for the `Oracle_LogManagesRecord` class, see TABLE A-15. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-15), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Record log

TABLE A-15 Properties for Oracle_LogManagesRecord

Property	Data Type	Description	ILOM Value
`Log`	`Oracle_RecordLog` REF	The `Log` property is a mandatory *key* property. Indicates the `Oracle_RecordLog`.	Object path to the instance of `Oracle_RecordLog` representing the IPMI SEL log.
`Record`	`Oracle_LogEntry` REF	The `Record` property is a mandatory *key* property. Indicates the `Oracle_LogEntry`.	Object path to an instance of `Oracle_LogEntry`.

Oracle_Memory

Description:	Provides capabilities and management of memory-related LogicalDevices.
Inheritance:	`CIM_Memory`
Properties:	For a description of the supported properties for the `Oracle_Memory` class, see TABLE A-16. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-16), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	System Memory

TABLE A-16 Properties for Oracle_Memory

Property	Data Type	Description	ILOM Value
`CreationClassName`	string	The `CreationClassName` property is a mandatory *key* property. `CreationClassName` indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	Set to `Oracle_Memory`.
`DeviceID`	string	The `DeviceID` property is a mandatory *key* property. An address or other identifying information used to uniquely name the `LogicalDevice`.	Implementation dependent value representing unique ID.
`SystemCreationClassName`	string	The `SystemCreationClassName` property is a mandatory *key* property. Indicates the System`CreationClassName` of the scoping system.	Set to `Oracle_ComputerSystem`.
`SystemName`	string	The `SystemName` property is a mandatory *key* property. Indicates the `SystemName` of the scoping system.	Will be set to `Oracle_ComputerSystem.Name` of the instance of `Oracle_ComputerSystem` that represents the controllee.

TABLE A-16 Properties for Oracle_Memory *(Continued)*

Property	Data Type	Description	ILOM Value
Access	uint16[]	The Access property describes whether the media is *readable* (value=1), *writeable* (value=2), or both (value=3). *Unknown* (0) and *Write Once* (4) can also be defined. The following values apply: {0, 1, 2, 3, 4} Definitions for the above values are: {Unknown, Readable, Writeable, Read/Write Supported, Write Once}	Set to 3 (Read/Write Supported).
BlockSize	uint16[]	Size in bytes of the blocks that form this StorageExtent. If the block size is variable, then the maximum block size in bytes should be specified. If the block size is unknown or if a block concept is not valid (for example, for AggregateExtents, Memory or LogicalDisks), enter a 1.	Set to appropriate value if memory size can be computed.
ElementName	string	The ElementName property is a user-friendly name. This property allows each instance to define a user-friendly name in addition to its *key* properties, identity data, and description information. **Note -** The Name property of ManagedSystemElement is also defined as a user-friendly name. But, it is often subclassed to be a *Key*. It is not reasonable that the same property can convey both identity and a user-friendly name, without inconsistencies. Where Name exists and is not a *Key* (such as for instances of LogicalDevice), the same information can be present in both the Name and ElementName properties.	Appropriate value.
EnabledDefault	uint16[]	Enumerated value indicating an administrator's default or startup configuration for the enabled state of an element. By default, the element is 2 (Enabled). The following values apply: {2, 3, 5, 6, 7, 9, .., 32768..65535} Definitions for the above values are: {Enabled, Disabled, Not Applicable, Enabled but Offline, No Default, Quiesce, DMTF Reserved, Vendor Reserved}	Set to default value 2 (Enabled).

TABLE A-16 Properties for Oracle_Memory *(Continued)*

Property	Data Type	Description	ILOM Value
EnabledState	uint16[]	Integer enumeration that indicates the enabled and disabled states of an element. It can also indicate the transitions between these requested states. For example, Shutting Down (value=4) and Starting (value=10) are transient states between enabled and disabled. The following values apply: • 0 (Unknown) • 1 (Other) • 2 (Enabled) - The element is or could be executing commands, will process any queued commands, and queues new requests. • 3 (Disabled) - The element will not execute commands and will drop any new requests. • 4 (Shutting Down) - The element is in the process of going to a Disabled state. • 5 (Not Applicable) - The element does not support being enabled or disabled. • 6 (Enabled but Offline) - The element might be completing commands, and will drop any new requests. • 7 (Test) - The element is in a test state. • 8 (Deferred) - The element might be completing commands, but will queue any new requests. • 9 (Quiesce) - The element is enabled but in a restricted mode. • 10 (Starting) - The element is in the process of going to an Enabled state. New requests are queued. • 11..32767 (DMTF Reserved) • 32768..65539 (Vendor Reserved)	Appropriate value.

TABLE A-16 Properties for Oracle_Memory *(Continued)*

Property	Data Type	Description	ILOM Value
`HealthState`	uint16[]	Indicates the current health of the element. This attribute expresses the health of this element but not necessarily that of its subcomponents. The following values apply: • 0 (Unknown) - The implementation cannot report on `HealthState` at this time. • 5 (OK) - The element is fully functional and is operating within normal operational parameters and without error. • 10 (Degraded/Warning) - The element is in working order and all functionality is provided. However, the element is not working to the best of its abilities. For example, the element might not be operating at optimal performance or it might be reporting recoverable errors. • 15 (Minor Failure) - All functionality is available but some might be degraded. • 20 (Major Failure) - The element is failing. It is possible that some or all of the functionality of this component is degraded or not working. • 25 (Critical Failure) - The element is non-functional and recovery might not be possible. • 30 (Non-recoverable Error) - The element has completely failed, and recovery is not possible. All functionality provided by this element has been lost. DMTF has reserved the unused portion of the continuum for additional health states in the future.	Appropriate value.

TABLE A-16 Properties for Oracle_Memory *(Continued)*

Property	Data Type	Description	ILOM Value
NumberOfBlocks	uint16[]	Total number of logically contiguous blocks, of size BlockSize, which form this Extent. The total size of the Extent can be calculated by multiplying BlockSize by NumberOfBlocks. If the BlockSize is 1, this property is the total size of the Extent.	Will have appropriate value if memory size can be computed.
OperationalStatus	uint16[]	The OperationalStatus property indicates the current statuses of the element. Various operational statuses are defined. Many of the enumeration's values are self-explanatory. Enumeration values can include any of the following: {Unknown, Other, OK, Degraded, Stressed, Predictive Failure, Error, Non-Recoverable Error, Starting, Stopping, Stopped, In Service, No Contact, Lost Communication, Aborted, Dormant, Supporting Entity in Error, Completed, Power Mode, DMTF Reserved, Vendor Reserved} Possible values for the enumeration values include: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, .., 0x8000..}	Appropriate value.
RequestedState	uint16[]	The RequestedState property is an integer enumeration that indicates the last requested or desired state for the element, irrespective of the mechanism through which it was requested. The actual state of the element is represented by EnabledState. This property is provided to compare the last requested and current enabled or disabled states. Element definitions include any of the following: {Unknown, Enabled, Disabled, Shut Down, No Change, Offline, Test, Deferred, Quiesce, Reboot, Reset, Not Applicable, DMTF Reserved, Vendor Reserved} Values for the above definitions include: {0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, .., 32768..65535} **Note -** When EnabledState is set to 5 (Not Applicable), then this property has no meaning.	Set to 12 (Not Applicable).

Oracle_NumericSensor

Description:	A numeric sensor that returns numeric readings and optionally supports thresholds settings.
Inheritance:	`CIM_NumericSensor`
Properties:	For a description of the supported properties for the `Oracle_NumericSensor` class, see TABLE A-17. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-17), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Sensor

TABLE A-17 Properties for Oracle_NumericSensor

Property	Data Type	Description	ILOM Value
`CreationClassName`	string	The `CreationClassName` property is a mandatory *key* property. `CreationClassName` indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	Set to `Oracle_NumericSensor`.
`DeviceID`	string	The `DeviceID` property is a mandatory *key* property. An address or other identifying information used to uniquely name the `LogicalDevice`.	Set to the NAC name of the sensor.
`SystemCreationClassName`	string	The `SystemCreationClassName` property is a mandatory *key* property. Indicates the `CreationClassName` for the scoping system.	Will be set to `Oracle_ComputerSystem.Name` of the instance of `Oracle_ComputerSystem` which represents the controllee.

TABLE A-17 Properties for Oracle_NumericSensor *(Continued)*

Property	Data Type	Description	ILOM Value
SystemName	string	The SystemName property is a mandatory *key* property. Indicates the SystemName of the scoping system.	Set to Oracle_ComputerSystem.Name of the instance of Oracle_ComputerSystem that represents the controllee.
BaseUnits	uint16[]	The base unit of the values returned by this sensor. All the values returned by this sensor are represented in the units obtained by (BaseUnits * 10 raised to the power of the UnitModifier). For example, if BaseUnits is *Volts* and the UnitModifier is -6, then the units of the values returned are microvolts. However, if the RateUnits property is set to a value other than *None,* then the units are further qualified as rate units. In the above example, if RateUnits is set to *Per Second*, then the values returned by the sensor are in microvolts/second. The units apply to all numeric properties of the sensor, unless explicitly overridden by the units qualifier. The following values apply: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66} Definitions of above values are: {Unknown, Other, Degrees C, Degrees F, Degrees K, Volts, Amps, Watts, Joules, Coulombs, VA, Nits, Lumens, Lux, Candelas, kPa, PSI, Newtons, CFM, RPM, Hertz, Seconds, Minutes, Hours, Days, Weeks, Mils, Inches, Feet, Cubic Inches, Cubic Feet, Meters, Cubic Centimeters, Cubic Meters, Liters, Fluid Ounces, Radians, Steradians, Revolutions, Cycles, Gravities, Ounces, Pounds, Foot-Pounds, Ounce-Inches, Gauss, Gilberts, Henries, Farads, Ohms, Siemens, Moles, Becquerels, PPM (parts/million), Decibels, DbA, DbC, Grays, Sieverts, Color Temperature Degrees K, Bits, Bytes, Words (data), DoubleWords, QuadWords, Percentage, Pascals}	Appropriate value depending on sensor type.
CurrentReading	sint32	The current value indicated by the sensor.	Appropriate value.
CurrentState	string	The current state indicated by the sensor. This is always one of the *PossibleStates.*	Appropriate value representing current state of the sensor.

TABLE A-17 Properties for Oracle_NumericSensor *(Continued)*

Property	Data Type	Description	ILOM Value
ElementName	string	The ElementName property is a user-friendly name. This property allows each instance to define a user-friendly name in addition to its key properties, identity data, and description information. **Note -** The Name property of ManagedSystemElement is also defined as a user-friendly name. But, it is often subclassed to be a *Key*. It is not reasonable that the same property can convey both identity and a user-friendly name, without inconsistencies. Where Name exists and is *not a Key* (such as for instances of LogicalDevice), the same information can be present in both the Name and ElementName properties.	Set to the NAC name of the sensor.
EnabledDefault	uint16[]	An enumerated value indicating an administrator's default or startup configuration for the enabled state of an element. By default, the element is *Enabled* (value=2). The following values apply: {2, 3, 5, 6, 7, 9, .., 32768..65535} Definitions of the above values are: {Enabled, Disabled, Not Applicable, Enabled but Offline, No Default, Quiesce, DMTF Reserved, Vendor Reserved}	Set to default value 2 (Enabled).

TABLE A-17 Properties for Oracle_NumericSensor *(Continued)*

Property	Data Type	Description	ILOM Value
EnabledState	uint16[]	Integer enumeration that indicates the enabled and disabled states of an element. It can also indicate the transitions between these requested states. For example, *Shutting Down* (value=4) and *Starting* (value=10) are transient states between enabled and disabled. The following values apply: • 0 (Unknown) • 1 (Other) • 2 (Enabled) - The element is or could be executing commands, will process any queued commands, and queues new requests. • 3 (Disabled) - The element will not execute commands and will drop any new requests. • 4 (Shutting Down) - The element is in the process of going to a Disabled state. • 5 (Not Applicable) - The element does not support being enabled or disabled. • 6 (Enabled but Offline) - The element might be completing commands, and will drop any new requests. • 7 (Test) - The element is in a test state. • 8 (Deferred) - The element might be completing commands, but will queue any new requests. • 9 (Quiesce) - The element is enabled but in a restricted mode. • 10 (Starting) - The element is in the process of going to an Enabled state. New requests are queued. • 11..32767 (DMTF Reserved) • 32768..65539 (Vendor Reserved)	Will have appropriate value depending on whether the sensor is enabled, disabled or unknown.

TABLE A-17 Properties for Oracle_NumericSensor *(Continued)*

Property	Data Type	Description	ILOM Value
HealthState	uint16[]	Indicates the current health of the element. This attribute expresses the health of this element but not necessarily that of its subcomponents. The following values apply: • 0 (Unknown) - The implementation cannot report on HealthState at this time. • 5 (OK) - The element is fully functional and is operating within normal operational parameters and without error. • 10 (Degraded/Warning) - The element is in working order and all functionality is provided. However, the element is not working to the best of its abilities. For example, the element might not be operating at optimal performance or it might be reporting recoverable errors. • 15 (Minor Failure) - All functionality is available but some might be degraded. • 20 (Major Failure) - The element is failing. It is possible that some or all of the functionality of this component is degraded or not working. • 25 (Critical Failure) - The element is non-functional and recovery might not be possible. • 30 (Non-recoverable Error) - The element has completely failed, and recovery is not possible. All functionality provided by this element has been lost. DMTF has reserved the unused portion of the continuum for additional health states in the future.	Appropriate value.
LowerThresholdCritical	sint32	The sensor's threshold values specify the ranges (min and max values) for determining whether the sensor is operating under Normal, NonCritical, Critical or Fatal conditions. If the CurrentReading is between LowerThresholdCritical and Lower ThresholdFatal, then the CurrentState is Critical.	Will have appropriate value if sensor supports this threshold. If sensor does not support this threshold, this property will not be set.
LowerThresholdFatal	sint32	The sensor's threshold values specify the ranges (min and max values) for determining whether the sensor is operating under Normal, NonCritical, Critical or Fatal conditions. If the CurrentReading is below LowerThresholdFatal, then the current state is Fatal.	Will have appropriate value if sensor supports this threshold. If sensor does not support this threshold, this property will not be set.

TABLE A-17 Properties for Oracle_NumericSensor *(Continued)*

Property	Data Type	Description	ILOM Value
OperationalStatus	uint16[]	The OperationalStatus property indicates the current statuses of the element. Various operational statuses are defined. Many of the enumeration's values are self-explanatory. Enumeration definitions can include any of the following: {Unknown, Other, OK, Degraded, Stressed, Predictive Failure, Error, Non-Recoverable Error, Starting, Stopping, Stopped, In Service, No Contact, Lost Communication, Aborted, Dormant, Supporting Entity in Error, Completed, Power Mode, DMTF Reserved, Vendor Reserved} Values for the enumeration definitions are as follows: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, .., 0x8000..}	Will have appropriate value.
PossibleStates	string	PossibleStates enumerates the string outputs of the sensor. For example, a switch sensor may output the states *On*, or *Off* Another implementation of the switch might output the states *Open* and *Close*. Another example is a NumericSensor supporting thresholds. This sensor can report the states like *Normal*, *Upper Fatal*, *Lower Non-Critical*, etc. A NumericSensor that does not publish readings and thresholds, but can store the data internally and still report its states.	Will have appropriate values depending on the type of the sensor.
RateUnits	uint16[]	Specifies if the units returned by this sensor are rate units. All the values returned by this Sensor are represented in the units obtained by (BaseUnits * 10 raised to the power of the UnitModifier). This is true unless this property (RateUnits) has a value different than "None". For example, if BaseUnits is Volts and the UnitModifier is -6, then the units of the values returned are microvolts. But, if the RateUnits property is set to a value other than "None", then the units are further qualified as rate units. In the above example, if RateUnits is set to "Per Second", then the values returned by the Sensor are in microvolts/second. The units apply to all numeric properties of the Sensor, unless explicitly overridden by the Units qualifier. Any implementation of CurrentReading should be qualified with either a Counter or a Gauge qualifier, depending on the characteristics of the sensor being modeled.	Will be set to 0.

TABLE A-17 Properties for Oracle_NumericSensor *(Continued)*

Property	Data Type	Description	ILOM Value
RequestedState	uint16[]	The RequestedState property is an integer enumeration that indicates the last requested or desired state for the element, irrespective of the mechanism through which it was requested. The actual state of the element is represented by EnabledState. This property is provided to compare the last requested and current enabled or disabled states. Element definitions include any of the following: {Unknown, Enabled, Disabled, Shut Down, No Change, Offline, Test, Deferred, Quiesce, Reboot, Reset, Not Applicable, DMTF Reserved, Vendor Reserved} Values for the above definitions are as follows {0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, .., 32768..65535} **Note -** When EnabledState is set to 5 (Not Applicable), then this property has no meaning.	Set to 12 (Not Applicable).
SensorType	uint16[]	Identifies the type of the sensor, for example, voltage sensor or temperature sensor. If the type is set to *Other*, then the OtherSensorType Description can be used to further identify the type, or if the sensor has numeric readings, then the type of the sensor can be implicitly determined by the Units. A description of the different Sensor types is as follows: • A temperature sensor measures the environmental temperature. • Voltage and current sensors measure electrical voltage and current readings. • A tachometer measures speed/revolutions of a device. For example, a fan device can have an associated tachometer which measures its speed. • A counter is a general purpose sensor that measures some numerical property of a device. • A counter value can be cleared, but it never decreases. • A switch sensor has states like Open or Close, On or Off, or Upor Down. • A Lock has states of Lockedor Unlocked. Humidity, smoke detection and air flow sensors measure the equivalent environmental characteristics. • A presence sensor detects the presence of a PhysicalElement. • A power consumption sensor measures the instantaneous power consumed by a managed element. • A power production sensor measures the instantaneous power produced by a managed element such as a power supply or a voltage regulator. • A pressure sensor is used to report pressure.	Appropriate value.

TABLE A-17 Properties for Oracle_NumericSensor *(Continued)*

Property	Data Type	Description	ILOM Value
		The following values apply: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, .., 32768..65535} Definitions of above values are: {Unknown, Other, Temperature, Voltage, Current, Tachometer, Counter, Switch, Lock, Humidity, Smoke Detection, Presence, Air Flow, Power Consumption, Power Production, Pressure, DMTF Reserved, Vendor Reserved}	
`SupportedThresholds`	uint16[]	An array representing the thresholds supported by this sensor. The following values apply: {0, 1, 2, 3, 4, 5} Definitions of above values are: {LowerThresholdNonCritical, UpperThresholdNonCritical, LowerThresholdCritical, UpperThresholdCritical, LowerThresholdFatal, UpperThresholdFatal}	String values of supported thresholds.
`UpperThresholdCritical`	sint32	The sensor's threshold values specify the ranges (min and max values) for determining whether the sensor is operating under Normal, NonCritical, Critical or Fatal conditions. If the `CurrentReading` is above `UpperThresholdFatal`, then the current state is Fatal.	Will have appropriate value if sensor supports this threshold. If sensor does not support this threshold, then this property will not be set.
`UpperThresholdNonCritical`	sint32	The sensor's threshold values specify the ranges (min and max values) for determining whether the sensor is operating under Normal, NonCritical, Critical or Fatal conditions. If the `CurrentReading` is between `LowerThresholdNonCritical` and `UpperThresholdNonCritical`, then the sensor is reporting a normal value. If the `CurrentReading` is between `UpperThreshold NonCritical` and `UpperThresholdCritical`, then the `current state` is `NonCritical`	Will have appropriate value if sensor supports this threshold. If sensor does not support this threshold, then this property will not be set.

Oracle_PhysicalAssetCapabilities

Description:	Provides the capabilities for representing FRU related information for an associated instance of the `CIM_PhysicalElement` subclass.
Inheritance:	`CIM_PhysicalAssetCapabilities`
Properties:	For a description of the supported properties for the `Oracle_PhysicalAssetCapabilities` class, see TABLE A-18. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-18), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Physical asset

TABLE A-18 Properties for Oracle_PhysicalAssetCapabilities

Property	Data Type	Description	ILOM Value
InstanceID	string	The InstanceID property is a mandatory *key* property. Within the scope of the instantiating NameSpace, the InstanceID property uniquely identifies an instance of this class. The value of InstanceID should be constructed using the following preferred algorithm: <OrgID>:<LocalID> Where: • <OrgID> and <LocalID> are separated by a colon ':' • <OrgID> must include a copyrighted, trademarked or otherwise unique name that is owned by the business entity creating or defining the InstanceID, or is a registered ID that is assigned to the business entity by a recognized global authority (This is similar to the <Schema Name>_<Class Name> structure of schema class names) • <OrgID> must not contain a colon (':'). The first colon to appear in InstanceID must appear between <OrgID> and <LocalID> • <LocalID> is chosen by the business entity and should not be re-used to identify different underlying (real-world) elements • For DMTF defined instances, the <OrgID> must be set to *CIM*. If the above preferred algorithm is not used, the defining entity must ensure that the resultant InstanceID is not re-used across any instance IDs produced by this or other providers for this instance's NameSpace.	Implementation dependent value representing unique ID of the PhysicalAssetCapabilities.
FRUInfoSupported	boolean	boolean indicating whether the *PartNumber*, *Serial Number*, *Model*, *Manufacturer*, and *SKU* properties of PhysicalElement are non-null, non-blank values, and the availability of the complete FRU information.	Will be set to TRUE or FALSE depending on whether the associated instance of CIM_PhysicalElement is considered to be a FRU by the platform.

Oracle_PhysicalComponent

Description:	The PhysicalComponent class represents any low-level or basic component within a package. A component object either can not or does not need to be decomposed into its constituent parts.
Inheritance:	`CIM_PhysicalComponent`
Properties:	For a description of the supported properties for the `Oracle_PhysicalComponent` class, see TABLE A-19. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-19), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Physical Asset

TABLE A-19 Properties for Oracle_PhysicalComponent

Property	Data Type	Description	ILOM Value
`CreationClassName`	string	The `CreationClassName` property is a mandatory *key* property. `CreationClassName` indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	Set to `Oracle_PhysicalComponent`

TABLE A-19 Properties for Oracle_PhysicalComponent *(Continued)*

Property	Data Type	Description	ILOM Value
`Tag`	string	The `Tag` property is a mandatory *key* property. An arbitrary string that uniquely identifies the physical element and serves as the key of the element. The `Tag` property can contain information such as asset tag or serial number data. The key for physical element is placed very high in the object hierarchy in order to independently identify the hardware or entity, regardless of physical placement in or on cabinets, adapters, and so on. For example, a hot-swappable or removable component can be taken from its containing (scoping) package and be temporarily unused. The object still continues to exist and can even be inserted into a different scoping container. Therefore, the key for physical element is an arbitrary string and is defined independently of any placement or location-oriented hierarchy.	Set to component NAC name.
`CanBeFRUed`	boolean	A boolean that indicates whether this physical element can be a FRU (TRUE) or not (FALSE).	Will be set to TRUE or FALSE depending on whether the component is considered to be a FRU by the platform.
`Description`	string	The `Description` property provides a textual description of the object	Will have appropriate description.

TABLE A-19 Properties for Oracle_PhysicalComponent *(Continued)*

Property	Data Type	Description	ILOM Value
`ElementName`	string	User-friendly name. This property allows each instance to define a user-friendly name in addition to its key properties, identity data, and description information. **Note -** The `Name` property of `ManagedSystemElement` is also defined as a user-friendly name. But, it is often subclassed to be a *Key*. It is not reasonable that the same property can convey both identity and a user-friendly name, without inconsistencies. Where `Name` exists and is not a *Key* (such as for instances of `LogicalDevice`), the same information can be present in both the `Name` and `ElementName` properties.	Set to component NAC name

TABLE A-19 Properties for Oracle_PhysicalComponent *(Continued)*

Property	Data Type	Description	ILOM Value
HealthState	uint16	Indicates the current health of the element. This attribute expresses the health of this element but not necessarily that of its subcomponents. The possible values are 0 to 30, where 5 means the element is entirely healthy and 30 means the element is completely non-functional. The following continuum is defined: • "Non-recoverable Error" (30) - The element has completely failed, and recovery is not possible. All functionality provided by this element has been lost. • "Critical Failure" (25) - The element is nonfunctional and recovery might not be possible. • "Major Failure" (20) - The element is failing. It is possible that some or all of the functionality of this component is degraded or not working. • "Minor Failure" (15) - All functionality is available but some might be degraded. • "Degraded/Warning" (10) - The element is in working order and all functionality is provided. However, the element is not working to the best of its abilities. For example, the element might not be operating at optimal performance or it might be reporting recoverable errors. • "OK" (5) - The element is fully functional and is operating within normal operational parameters and without error.	Will have the appropriate value depending on whether the component is in error state or not.

TABLE A-19 Properties for Oracle_PhysicalComponent *(Continued)*

Property	Data Type	Description	ILOM Value
		• "Unknown" (0) - The implementation cannot report on HealthState at this time. DMTF has reserved the unused portion of the continuum for additional HealthStates in the future. Possible values are: {0, 5, 10, 15, 20, 25, 30, ..} Definitions for the above values are: {Unknown, OK, Degraded/Warning, Minor failure, Major failure, Critical failure, Non-recoverable error, DMTF Reserved}	
`Manufacturer`	string	The name of the organization responsible for producing the `PhysicalElement`. This organization might be the entity from whom the element is purchased, but this is not necessarily true. The latter information is contained in the `Vendor` property of `CIM_Product`.	Will have the appropriate value if the component is considered as a FRU by the platform.
`Model`	string	The name by which the `PhysicalElement` is generally known.	Will have the appropriate value if the component is considered as a FRU by the platform.

TABLE A-19 Properties for Oracle_PhysicalComponent *(Continued)*

Property	Data Type	Description	ILOM Value
`OperationalStatus`	Uint16 []	Indicates the current statuses of the element. Various operational statuses are defined. Many of the enumeration's values are self-explanatory. However, a few are not and are described here in more detail. • "Stressed" indicates that the element is functioning, but needs attention. • Examples of "Stressed" states are "overload, overheated, and so on. " • "Predictive Failure" indicates that an element is functioning nominally but predicting a failure in the near future." • "In Service" describes an element being configured, maintained, cleaned, or otherwise administered." • "No Contact" indicates that the monitoring system has knowledge of this element, but has never been able to establish communications with it." • "Lost Communication" indicates that the `ManagedSystemElement` is known to exist and has been contacted successfully in the past, but is currently unreachable." • "Stopped" and "`Aborted`" are similar, although the former implies a clean and orderly stop, while the latter implies an abrupt stop where the state and configuration of the element might need to be updated. • "Dormant" indicates that the element is inactive or quiesced.	OperationalStatus[0] will have appropriate value depending on whether the component is in error state or not.

TABLE A-19 Properties for Oracle_PhysicalComponent *(Continued)*

Property	Data Type	Description	ILOM Value
		• "Supporting Entity in Error" indicates that this element might be "OK" but that another element, on which it is dependent, is in error. An example is a network service or endpoint that cannot function due to lower-layer networking problems. • "Completed" indicates that the element has completed its operation. This value should be combined with either OK, Error, or Degraded so that a client can tell if the complete operation Completed with OK (passed), Completed with Error (failed), or Completed with Degraded (the operation finished, but it did not complete OK or did not report an error). • "Power Mode" indicates that the element has additional power model information contained in the Associated `PowerManagementService` association. `OperationalStatus` replaces the `Status` property on `ManagedSystemElement` to provide a consistent approach to enumerations, to address implementation needs for an array property, and to provide a migration path from today's environment to the future. This change was not made earlier because it required the deprecated qualifier. Due to the widespread use of the existing `Status` property in management applications, it is strongly recommended that providers or instrumentation provide both the	

TABLE A-19 Properties for Oracle_PhysicalComponent *(Continued)*

Property	Data Type	Description	ILOM Value
		`Status` and `OperationalStatus` properties. Further, the first value of `OperationalStatus` should contain the primary status for the element. When instrumented, `Status` (because it is single-valued) should also provide the primary status of the element. Possible values are: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, .., 0x8000..} Definitions of the above values are: { Unknown, Other, OK, Degraded, Stressed, Predictive Failure, Error, Non- Recoverable Error, Starting, Stopping, Stopped, In Service, No Contact, Lost Communication, Aborted, Dormant, Supporting Entity in Error, Completed, Power Mode, DMTF Reserved, VendorReserved}	
`PartNumber`	string	The part number assigned by the organization that is responsible for producing or manufacturing the `PhysicalElement`.	Will have the appropriate value if the component is considered a FRU by the platform.
`SKU`	string	The stock-keeping unit number for this `PhysicalElement`.	Will have the appropriate value if the component is considered a FRU by the platform.
`SerialNumber`	string	A manufacturer-allocated number used to identify the `Physical Element`.	Will have the appropriate value if the component is considered a FRU by the platform.

TABLE A-19 Properties for Oracle_PhysicalComponent *(Continued)*

Property	Data Type	Description	ILOM Value
StatusDescriptions	String[]	Strings describing the various OperationalStatus array values. For example, if "Stopping" is the value assigned to OperationalStatus, then this property may contain an explanation as to why an object is being stopped. **Note -** Note that entries in this array are correlated with those at the same array index in OperationalStatus.	StatusDescriptions[0] will have appropriate description on the reason for the value of OperationalStatus[0].

Oracle_PhysicalElementCapabilities

Description:	`Oracle_PhysicalElementCapabilities` is used to associate an instance of `CIM_PhysicalElement` to its capabilities, `Oracle_PhysicalAssetCapabilities`.
Inheritance:	`CIM_ElementCapabilities`
Properties:	For a description of the supported properties for the `Oracle_PhysicalElementCapabilities` class, see TABLE A-20. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-20), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Physical Asset

TABLE A-20 Properties for Oracle_PhysicalElementCapabilities

Property	Data Type	Description	ILOM Value
`Capabilities`	`Oracle_PhysicalAssetCapabilities` REF	The `Capabilities` property is a mandatory *key* property. The Capabilities object associated with the element.	Object path to an instance of `Oracle_PhysicalAssetCapabilities`.
`ManagedElement`	`CIM_PhysicalElement` REF	The `ManagedElement` property is a mandatory *key* property. Identifies the managed element.	Object path to an instance of `Oracle_PhysicalElement`.

Oracle_PhysicalMemory

Description:	The `Oracle_PhysicalMemory` is used to represent low-level memory devices such as SIMMs, DIMMs, raw memory chips, and so forth.
Inheritance:	`CIM_PhysicalMemory`
Properties:	For a description of the supported properties for the `Oracle_PhysicalMemory` class, see TABLE A-21. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-21), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Physical asset

TABLE A-21 Properties for Oracle_PhysicalMemory

Property	Data Type	Description	ILOM Value
`CreationClassName`	string	The `CreationClassName` property is a mandatory *key* property `CreationClassName` indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	Set to `Oracle_PhysicalMemory`.
`Tag`	string	The `Tag` property is a mandatory *key* property The `Tag` property is an arbitrary string that uniquely identifies the physical element and serves as the key of the element. The `Tag` property can contain information such as asset tag or serial number data. The key for `PhysicalElement` is placed very high in the object hierarchy in order to independently identify the hardware or entity, regardless of physical placement in or on cabinets, adapters, and so on. For example, a hot-swappable or removable component can be taken from its containing (scoping) Package and be temporarily unused. The object still continues to exist and can even be inserted into a different scoping container. Therefore, the key for `PhysicalElement` is an arbitrary string and is defined independently of any placement or location-oriented hierarchy.	Set to component NAC name.

TABLE A-21 Properties for Oracle_PhysicalMemory *(Continued)*

Property	Data Type	Description	ILOM Value
`CanBeFRUed`	boolean	The boolean that indicates whether this `PhysicalElement` is a FRU (TRUE) or not (FALSE).	Set to TRUE or FALSE depending on whether the component is considered to be a FRU by the platform.
`Description`	string	Textual description of the object.	Appropriate description.
`FormFactor`	uint16[]	The implementation form factor for the Chip. For example, values such as SIMM (7), TSOP (9) or PGA (10) can be specified. The following values apply: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23} Definitions for the above values are: {Unknown, Other, SIP, DIP, ZIP, SOJ, Proprietary, SIMM, DIMM, TSOP, PGA, RIMM, SODIMM, SRIMM, SMD, SSMP, QFP, TQFP, SOIC, LCC, PLCC, BGA, FPBGA, LGA}	Set to value 8 (DIMM).

TABLE A-21 Properties for Oracle_PhysicalMemory *(Continued)*

Property	Data Type	Description	ILOM Value
HealthState	uint16[]	Indicates the current health of the element. This attribute expresses the health of this element but not necessarily that of its subcomponents. The following values apply: • 0 (Unknown) - The implementation cannot report on HealthState at this time. • 5 (OK) - The element is fully functional and is operating within normal operational parameters and without error. • 10 (Degraded/Warning) - The element is in working order and all functionality is provided. However, the element is not working to the best of its abilities. For example, the element might not be operating at optimal performance or it might be reporting recoverable errors. • 15 (Minor Failure) - All functionality is available but some might be degraded. • 20 (Major Failure) - The element is failing. It is possible that some or all of the functionality of this component is degraded or not working. • 25 (Critical Failure) - The element is non-functional and recovery might not be possible. • 30 (Non-recoverable Error) - The element has completely failed, and recovery is not possible. All functionality provided by this element has been lost. DMTF has reserved the unused portion of the continuum for additional health states in the future.	Will have appropriate value depending on whether the component is in error state or not.
MemoryType	uint16[]	The type of PhysicalMemory. Synchronous DRAM is also known as SDRAM. Cache DRAM is also known as CDRAM. CDRAM is also known as Cache DRAM. SDRAM is also known as Synchronous DRAM. BRAM is also known as Block RAM. The following values apply: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26..32567, 32568..65535} Definitions for the above values are: {"Unknown", "Other", "DRAM", "Synchronous DRAM", "Cache DRAM", "EDO", "EDRAM", "VRAM", "SRAM", "RAM", "ROM", "Flash", "EEPROM", "FEPROM", "EPROM", "CDRAM", "3DRAM", "SDRAM", "SGRAM", "RDRAM", "DDR", "DDR-2", "BRAM", "FB-DIMM", "DDR3", "FBD2", "DMTF Reserved","Vendor Reserved" }	Appropriate value.

TABLE A-21 Properties for Oracle_PhysicalMemory *(Continued)*

Property	Data Type	Description	ILOM Value
Manufacturer	string	The name of the organization responsible for producing the PhysicalElement. This organization might be the entity from whom the Element is purchased, but this is not necessarily true. The latter information is contained in the Vendor property of CIM_Product.	Will have appropriate value if the processor chip is considered a FRU by the platform.
Model	string	The name by which the PhysicalElement is generally known.	Will have appropriate value if the processor chip is considered a FRU by the platform.
OperationalStatus	uint16[]	The OperationalStatus property indicates the current statuses of the element. Various operational statuses are defined. Many of the enumeration's values are self-explanatory. Enumeration values can include any of the following: {Unknown, Other, OK, Degraded, Stressed, Predictive Failure, Error, Non-Recoverable Error, Starting, Stopping, Stopped, In Service, No Contact, Lost Communication, Aborted, Dormant, Supporting Entity in Error, Completed, Power Mode, DMTF Reserved, Vendor Reserved} Possible values for the enumeration values include: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, .., 0x8000..}	OperationalStatus[0] will have appropriate value depending on whether the component is in error state or not.
PartNumber	string	Part number assigned by the organization that is responsible for producing or manufacturing the physical element.	Will have appropriate value if the processor chip is considered a FRU by the platform.
SKU	string	The stock-keeping unit number for this physical element.	Will have appropriate value if the processor chip is considered a FRU by the platform.
SerialNumber	string	A manufacturer-allocated number used to identify the physical element.	Will have appropriate value if the processor chip is considered a FRU by the platform.
StatusDescriptions	string[]	Strings describing the various OperationalStatus array values. For example, if *Stopping* is the value assigned to OperationalStatus, then this property might contain an explanation as to why an object is being stopped. Note that entries in this array are correlated with those at the same array index in OperationalStatus.	StatusDescriptions [0] will have appropriate description on the reason for the value of OperationalStatus[0].

Oracle_PhysicalPackage

Description:	The `Oracle_PhysicalPackage` class represents physical elements that contain or host other components.
Inheritance:	`CIM_PhysicalPackage`
Properties:	For a description of the supported properties for the `Oracle_PhysicalPackage` class, see TABLE A-22. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-22), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Physical asset

TABLE A-22 Properties for Oracle_PhysicalPackage

Property	Data Type	Description	ILOM Value
`CreationClassName`	string	The `CreationClassName` property is a mandatory *key* property `CreationClassName` indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	Set to `Oracle_PhysicalPackage`.
`Tag`	string	The `Tag` property is a mandatory *key* property The `Tag` property is an arbitrary string that uniquely identifies the physical element and serves as the key of the element. The `Tag` property can contain information such as asset tag or serial number data. The key for `PhysicalElement` is placed very high in the object hierarchy in order to independently identify the hardware or entity, regardless of physical placement in or on Cabinets, Adapters, and so on. For example, a hot-swappable or removable component can be taken from its containing (scoping) package and be temporarily unused. The object still continues to exist and can even be inserted into a different scoping container. Therefore, the key for `PhysicalElement` is an arbitrary string and is defined independently of any placement or location-oriented hierarchy.	Set to component NAC name.

TABLE A-22 Properties for Oracle_PhysicalPackage *(Continued)*

Property	Data Type	Description	ILOM Value
CanBeFRUed	boolean	A boolean that indicates whether this PhysicalElement is a FRU (TRUE) or not (FALSE).	Will be set to TRUE or FALSE depending on whether the component is considered to be a FRU by the platform.
Description	string	Textual description of the object.	Appropriate description.
ElementName	string	The ElementName property is a user-friendly name. This property allows each instance to define a user-friendly name in addition to its key properties, identity data, and description information. Note that the Name property of ManagedSystemElement is also defined as a user-friendly name. But, it is often subclassed to be a Key. It is not reasonable that the same property can convey both identity and a user-friendly name, without inconsistencies. Where Name exists and is not a Key (such as for instances of LogicalDevice), the same information can be present in both the Name and ElementName properties.	Set to component NAC name.

TABLE A-22 Properties for Oracle_PhysicalPackage *(Continued)*

Property	Data Type	Description	ILOM Value
HealthState	uint16[]	Indicates the current health of the element. This attribute expresses the health of this element but not necessarily that of its subcomponents. The following values apply • 0 (Unknown) - The implementation cannot report on HealthState at this time. • 5 (OK) - The element is fully functional and is operating within normal operational parameters and without error. • 10 (Degraded/Warning) - The element is in working order and all functionality is provided. However, the element is not working to the best of its abilities. For example, the element might not be operating at optimal performance or it might be reporting recoverable errors. • 15 (Minor Failure) - All functionality is available but some might be degraded. • 20 (Major Failure) - The element is failing. It is possible that some or all of the functionality of this component is degraded or not working. • 25 (Critical Failure) - The element is non-functional and recovery might not be possible. • 30 (Non-recoverable Error) - The element has completely failed, and recovery is not possible. All functionality provided by this element has been lost. DMTF has reserved the unused portion of the continuum for additional health states in the future.	Will have appropriate value depending on whether the component is in error state or not.
Manufacturer	string	The name of the organization responsible for producing the PhysicalElement. This organization might be the entity from whom the Element is purchased, but this is not necessarily true. The latter information is contained in the Vendor property of CIM_Product.	Will have appropriate value if the processor chip is considered as a FRU by the platform.
Model	string	The name by which the PhysicalElement is generally known.	Will have appropriate value if the component is considered as a FRU by the platform.

TABLE A-22 Properties for Oracle_PhysicalPackage *(Continued)*

Property	Data Type	Description	ILOM Value
`OperationalStatus`	Uint16 []	Indicates the current statuses of the element. Various operational statuses are defined. Many of the enumeration's values are self-explanatory. However, a few are not and are described here in more detail. • "Stressed" indicates that the element is functioning, but needs attention. Examples of "Stressed" states are "overload, overheated, and so on. " • "Predictive Failure" indicates that an element is functioning nominally but predicting a failure in the near future." • "In Service" describes an element being configured, maintained, cleaned, or otherwise administered." • "No Contact" indicates that the monitoring system has knowledge of this element, but has never been able to establish communications with it." • "Lost Communication" indicates that the `ManagedSystemElement` is known to exist and has been contacted successfully in the past, but is currently unreachable." "Stopped" and "Aborted" are similar, although the former implies a clean and orderly stop, while the latter implies an abrupt stop where the state and configuration of the element might need to be updated. • "Dormant" indicates that the element is inactive or quiesced. • "Supporting Entity in Error" indicates that this element might be "OK" but that another element, on which it is dependent, is in error. An example is a network service or endpoint that cannot function due to lower-layer networking problems. • "Completed" indicates that the element has completed its operation. This value should be combined with either OK, Error, or Degraded so that a client can tell if the complete operation Completed with OK (passed), Completed with Error (failed), or Completed with Degraded (the operation finished, but it did not complete OK or did not report an error). • "Power Mode" indicates that the element has additional power model information contained in the Associated `PowerManagementService` association."	`OperationalStatus[0]` will have appropriate value depending on whether the component is in error state or not.

TABLE A-22 Properties for Oracle_PhysicalPackage *(Continued)*

Property	Data Type	Description	ILOM Value
		• `OperationalStatus` replaces the `Status` property on `ManagedSystemElement` to provide a consistent approach to enumerations, to address implementation needs for an array property, and to provide a migration path from today's environment to the future. This change was not made earlier because it required the deprecated qualifier. Due to the widespread use of the existing `Status` property in management applications, it is strongly recommended that providers or instrumentation provide both the `Status` and `OperationalStatus` properties. Further, the first value of `OperationalStatus` should contain the primary status for the element. When instrumented, **Status** (because it is single-valued) should also provide the primary status of the element. Possible values are: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, .., 0x8000..} Definitions of the above values are: { Unknown, Other, OK, Degraded, Stressed, Predictive Failure, Error, Non- Recoverable Error, Starting, Stopping, Stopped, In Service, No Contact, Lost Communication, Aborted, Dormant, Supporting Entity in Error, Completed, Power Mode, DMTF Reserved, Vendor Reserved}	

TABLE A-22 Properties for Oracle_PhysicalPackage *(Continued)*

Property	Data Type	Description	ILOM Value
PackageType	uint16[]	Enumeration defining the type of the PhysicalPackage. Note that this enumeration expands on the list in the Entity MIB (the attribute, entPhysicalClass). The numeric values are consistent with the CIM enumeration numbering guidelines, but are slightly different than the MIB values. Unknown - indicates that the package type is not known. Other - The package type does not correspond to an existing enumerated value. The value is specified using the OtherPackageType property. The values Rack through Port or Connector are defined per the Entity-MIB (where the semantics of rack are equivalent to the MIB's stack value). The other values (for battery, processor, memory, power source or generator and storage media package) are self-explanatory. A value of the blade server should be used when the PhysicalPackage contains the operational hardware aspects of a ComputerSystem, without the supporting mechanicals such as power and cooling. For example, a blade server (server module) includes processor(s) and memory, and relies on the containing chassis to supply power and cooling. In many respects, a blade can be considered a module or card. However, it is tracked differently by inventory systems and differs in terms of service philosophy. For example, a blade server is intended to be hot-plugged into a hosting enclosure without requiring additional cabling, and does not require a cover to be removed from the enclosure for installation. Similarly, a blade expansion module has characteristics of a blade server and a module or card. However, it is distinct from both due to inventory tracking and service philosophy, and because of its hardware dependence on a blade. A blade expansion module (or card) must be attached to a blade prior to inserting the resultant assembly into an enclosure. The following values apply: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17} Definitions for the above values are: {Unknown, Other, Rack, Chassis/Frame, Cross Connect/Backplane, Container/Frame Slot, Power Supply, Fan, Sensor, Module/Card, Port/Connector, Battery, Processor, Memory, Power Source/Generator, Storage Media Package (e.g., Disk or Tape Drive), Blade, Blade Expansion}	Appropriate value.

TABLE A-22 Properties for Oracle_PhysicalPackage *(Continued)*

Property	Data Type	Description	ILOM Value
PartNumber	string	Part number assigned by the organization that is responsible for producing or manufacturing the PhysicalElement.	Will have appropriate value if the processor chip is considered a FRU by the platform.
SKU	string	The SKU property is a manufacturer-allocated number used to identify the PhysicalElement.	Will have appropriate value if the processor chip is considered a FRU by the platform.
SerialNumber	string	A manufacturer-allocated number used to identify the PhysicalElement.	Will have appropriate value if the processor chip is considered a FRU by the platform.
StatusDescriptions	string[]	Strings describing the various OperationalStatus array values. For example, if "Stopping" is the value assigned to OperationalStatus, then this property may contain an explanation as to why an object is being stopped. Note that entries in this array are correlated with those at the same array index in OperationalStatus.	StatusDescriptions[0] will have appropriate description on the reason for the value of OperationalStatus[0].

Oracle_Processor

Description:	Identifies capabilities and management of the processor logical device.
Inheritance:	`CIM_Processor`
Properties:	For a description of the supported properties for the `Oracle_Processor` class, see TABLE A-23. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-23), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	CPU

TABLE A-23 Properties for Oracle_Processor

Property	Data Type	Description	ILOM Value
`CreationClassName`	string	The `CreationClassName` property is a mandatory *key* property. `CreationClassName` indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	Set to `Oracle_Processor`.
`DeviceID`	string	The `DeviceID` property is a mandatory *key* property. The Device ID indicates an address or other identifying information used to uniquely name the `LogicalDevice`.	Will be set to the NAC name of the sensor.
`SystemCreationClassName`	string	The `SystemCreationName` property is a mandatory *key* property. Indicates the `CreationClassName` of the scoping system.	Set to `Oracle_ComputerSystem`.
`SystemName`	string	The SystemName property is a mandatory *key* property. Indicates the `SystemName` of the scoping system.	Set to `Oracle_ComputerSystem.Name` of the instance of `Oracle_ComputerSystem` that represents the controllee.

TABLE A-23 Properties for Oracle_Processor *(Continued)*

Property	Data Type	Description	ILOM Value
CPUStatus	uint16[]	Indicates the current status of the Processor. For example, the processor might be disabled by the user (value=2), or disabled due to a POST error (value=3). Information in this property can be obtained from SMBIOS, the type 4 structure, and the status attribute. The following values are apply: {0, 1, 2, 3, 4, 7} Definitions of above values are: {Unknown, CPU Enabled, CPU Disabled by User, CPU Disabled By BIOS (POST Error), CPU Is Idle, Other}	Appropriate value.
ElementName	string	The ElementName property is a user-friendly name. This property allows each instance to define a user-friendly name in addition to its key properties, identity data, and description information. Note that the Name property of ManagedSystemElement is also defined as a user-friendly name. But, it is often subclassed to be a Key. It is not reasonable that the same property can convey both identity and a user-friendly name, without inconsistencies. Where Name exists and is not a Key (such as for instances of LogicalDevice), the same information can be present in both the Name and ElementName properties.	Will be set to the NAC name of the sensor.
EnabledDefault	uint16[]	Eumerated value indicating an administrator's default or startup configuration for the Enabled State of an element. By default, the element is "Enabled" (value=2). The following values apply: {2, 3, 5, 6, 7, 9, .., 32768..65535} Definitions for the above values are: {Enabled, Disabled, Not Applicable, Enabled but Offline, No Default, Quiesce, DMTF Reserved, Vendor Reserved}	Set to default value 2 (Enabled).

TABLE A-23 Properties for Oracle_Processor *(Continued)*

Property	Data Type	Description	ILOM Value
EnabledState	uint16[]	Integer enumeration that indicates the enabled and disabled states of an element. It can also indicate the transitions between these requested states. For example, shutting down (value=4) and starting (value=10) are transient states between enabled and disabled. The following values apply: • 0 (Unknown) • 1 (Other) • 2 (Enabled) - The element is or could be executing commands, will process any queued commands, and queues new requests. • 3 (Disabled) - The element will not execute commands and will drop any new requests. • 4 (Shutting Down) - The element is in the process of going to a Disabled state. • 5 (Not Applicable) - The element does not support being enabled or disabled. • 6 (Enabled but Offline) - The element might be completing commands, and will drop any new requests. • 7 (Test) - The element is in a test state. • 8 (Deferred) - The element might be completing commands, but will queue any new requests. • 9 (Quiesce) - The element is enabled but in a restricted mode. • 10 (Starting) - The element is in the process of going to an Enabled state. New requests are queued. • 11..32767 (DMTF Reserved) • 32768..65535 (Vendor Reserved)	Appropriate value.

TABLE A-23 Properties for Oracle_Processor *(Continued)*

Property	Data Type	Description	ILOM Value
Family	uint16[]	The Processor family type. For example, values include P*entium(R) processor with MMX(TM) technology* (value=14) and *68040* (value=96). The following values apply: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 48, 49, 50, 51, 52, 53, 54, 55, 64, 65, 66, 67, 68, 69, 80, 81, 82, 83, 84, 85, 86, 87, 88, 96, 97, 98, 99, 100, 101, 112, 120, 121, 122, 128, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 160, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 200, 201, 202, 203, 204, 210, 211, 212, 213, 230, 250, 251, 254, 255, 260, 261, 280, 281, 300, 301, 302, 320, 350, 500, 65534, 65535}	Appropriate value.

TABLE A-23 Properties for Oracle_Processor *(Continued)*

Property	Data Type	Description	ILOM Value
`HealthState`	uint16[]	Indicates the current health of the element. This attribute expresses the health of this element but not necessarily that of its subcomponents. The following values apply: • 0 (Unknown) - The implementation cannot report on `HealthState` at this time. • 5 (OK) - The element is fully functional and is operating within normal operational parameters and without error. • 10 (Degraded/Warning) - The element is in working order and all functionality is provided. However, the element is not working to the best of its abilities. For example, the element might not be operating at optimal performance or it might be reporting recoverable errors. • 15 (Minor Failure) - All functionality is available but some might be degraded. • 20 (Major Failure) - The element is failing. It is possible that some or all of the functionality of this component is degraded or not working. • 25 (Critical Failure) - The element is non-functional and recovery might not be possible. • 30 (Non-recoverable Error) - The element has completely failed, and recovery is not possible. All functionality provided by this element has been lost. DMTF has reserved the unused portion of the continuum for additional health states in the future.	Appropriate value.
`OperationalStatus`	uint16[]	The `OperationalStatus` property indicates the current statuses of the element. Various operational statuses are defined. Many of the enumeration's values are self-explanatory. Enumeration definitions can include any of the following: {Unknown, Other, OK, Degraded, Stressed, Predictive Failure, Error, Non-Recoverable Error, Starting, Stopping, Stopped, In Service, No Contact, Lost Communication, Aborted, Dormant, Supporting Entity in Error, Completed, Power Mode, DMTF Reserved, Vendor Reserved} Values for the enumeration definition are as follows: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, .., 0x8000..}	Appropriate value.

TABLE A-23 Properties for Oracle_Processor *(Continued)*

Property	Data Type	Description	ILOM Value
`RequestedState`	uint16[]	The `RequestedState` property is an integer enumeration that indicates the last requested or desired state for the element, irrespective of the mechanism through which it was requested. The actual state of the element is represented by `EnabledState`. This property is provided to compare the last requested and current enabled or disabled states. Element definitions include any of the following: {Unknown, Enabled, Disabled, Shut Down, No Change, Offline, Test, Deferred, Quiesce, Reboot, Reset, Not Applicable, DMTF Reserved, Vendor Reserved} Values for the above definitions are: {0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, .., 32768..65535} **Note -** When `EnabledState` is set to 5 ("Not Applicable"), then this property has no meaning.	Set to 12 (Not Applicable).

Oracle_ProcessorChip

Description:	Identifies the integrated circuit hardware for the processor.
Inheritance:	`CIM_Chip`
Properties:	For a description of the supported properties for the `Oracle_ProcessorChip` class, see TABLE A-24. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-24), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Physical asset

TABLE A-24 Properties for Oracle_ProcessorChip

Property	Data Type	Description	ILOM Value
`CreationClassName`	string	The `CreationClassName` property is a mandatory *key* property. `CreationClassName` indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	Set to `Oracle_ProcessorChip`.
`Tag`	string	The `Tag` property is a mandatory *key* property. The `Tag` property is an arbitrary string that uniquely identifies the physical element and serves as the key of the element. The `Tag` property can contain information such as asset tag or serial number data. The key for `PhysicalElement` is placed very high in the object hierarchy in order to independently identify the hardware or entity, regardless of physical placement in or on cabinets, adapters, and so on. For example, a hot-swappable or removable component can be taken from its containing (scoping) Package and be temporarily unused. The object still continues to exist and can even be inserted into a different scoping container. Therefore, the key for `PhysicalElement` is an arbitrary string and is defined independently of any placement or location-oriented hierarchy.	Set to component NAC name.

TABLE A-24 Properties for Oracle_ProcessorChip *(Continued)*

Property	Data Type	Description	ILOM Value
CanBeFRUed	boolean	The boolean indicates whether this PhysicalElement can be a FRU (TRUE) or not (FALSE).	Set to TRUE or FALSE depending on whether the component is considered to be a FRU by the platform.
Description	string	Textual description of the object.	Appropriate description.
ElementName	string	The ElementName property is a user-friendly name. This property allows each instance to define a user-friendly name in addition to its key properties, identity data, and description information. Note that the Name property of ManagedSystemElement is also defined as a user-friendly name. But, it is often subclassed to be a Key. It is not reasonable that the same property can convey both identity and a user-friendly name, without inconsistencies. Where Name exists and is not a Key (such as for instances of LogicalDevice), the same information can be present in both the Name and ElementName properties.	Set to component NAC name.

TABLE A-24 Properties for Oracle_ProcessorChip *(Continued)*

Property	Data Type	Description	ILOM Value
HealthState	uint16[]	Indicates the current health of the element. This attribute expresses the health of this element but not necessarily that of its subcomponents. The following values apply: • 0 (Unknown) - The implementation cannot report on HealthState at this time. • 5 (OK) - The element is fully functional and is operating within normal operational parameters and without error. • 10 (Degraded/Warning) - The element is in working order and all functionality is provided. However, the element is not working to the best of its abilities. For example, the element might not be operating at optimal performance or it might be reporting recoverable errors. • 15 (Minor Failure) - All functionality is available but some might be degraded. • 20 (Major Failure) - The element is failing. It is possible that some or all of the functionality of this component is degraded or not working. • 25 (Critical Failure) - The element is non-functional and recovery might not be possible. • 30 (Non-recoverable Error) - The element has completely failed, and recovery is not possible. All functionality provided by this element has been lost. DMTF has reserved the unused portion of the continuum for additional health states in the future.	Will have appropriate value depending on whether the component is in error state or not.
Manufacturer	string	The name of the organization responsible for producing the physical element. This organization might be the entity from whom the element is purchased, but this is not necessarily true. The latter information is contained in the Vendor property of CIM_Product.	Will have appropriate value if the processor chip is considered a FRU by the platform.
Model	string	The name by which the physical element is generally known.	Will have appropriate value if the processor chip is considered a FRU by the platform.

TABLE A-24 Properties for Oracle_ProcessorChip *(Continued)*

Property	Data Type	Description	ILOM Value
OperationalStatus	uint16[]	The OperationalStatus property indicates the current statuses of the element. Various operational statuses are defined. Many of the enumeration's values are self-explanatory. Enumeration definitions can include any of the following: {Unknown, Other, OK, Degraded, Stressed, Predictive Failure, Error, Non-Recoverable Error, Starting, Stopping, Stopped, In Service, No Contact, Lost Communication, Aborted, Dormant, Supporting Entity in Error, Completed, Power Mode, DMTF Reserved, Vendor Reserved} Values for the above definitions: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, .., 0x8000..}	OperationalStatus[0] will have appropriate value depending on whether the component is in error state or not.
PartNumber	string	Part number assigned by the organization that is responsible for producing or manufacturing the PhysicalElement.	Will have appropriate value if the processor chip is considered a FRU by the platform.
SKU	string	The stock-keeping unit number for this PhysicalElement.	Will have appropriate value if the processor chip is considered a FRU by the platform.
SerialNumber	string	A manufacturer-allocated number used to identify the physical element.	Will have appropriate value if the processor chip is considered a FRU by the platform.
StatusDescriptions	string[]	Strings describing the various OperationalStatus array values. For example, if *Stopping* is the value assigned to OperationalStatus, then this property may contain an explanation as to why an object is being stopped. Note that entries in this array are correlated with those at the same array index in OperationalStatus.	StatusDescriptions[0] will have appropriate description on the reason for the value of OperationalStatus[0].

Oracle_Realizes

Description:	`Oracle_Realizes` is the association that defines the mapping between `LogicalDevices` and the `PhysicalElements` that implement them.
Inheritance:	`CIM_Realizes`
Properties:	For a description of the supported properties for the `Oracle_Realizes` class, see TABLE A-25. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-25), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	• Physical asset • CPU • System Memory

TABLE A-25 Properties for Oracle_Realizes

Property	Data Type	Description	ILOM Value
`Antecedent`	`CIM_PhysicalElement` REF	The `Antecedent` property is a mandatory *key* property. The physical component that implements the device.	Object path to an instance of `CIM_PhysicalElement`.
`Dependent`	`CIM_LogicalDevice` REF	The `Dependent` property is a mandatory *key* property. The `LogicalDevice`.	Object path to an instance of `CIM_LogicalDevice`.

Oracle_RegisteredProfile

Description:	Provides implementation conformance to a CIM profile.
Inheritance:	`CIM_RegisteredProfile`
Properties:	For a description of the supported properties for the `Oracle_RegisteredProfile` class, see TABLE A-26. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-26), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Profile Registration

TABLE A-26 Properties for Oracle_RegisteredProfile

Property	Data Type	Description	ILOM Value
InstanceID	string	The InstanceID property is a *key* mandatory property. Within the scope of the instantiating NameSpace, the InstanceID property uniquely identifies an instance of this class. The value of InstanceID should be constructed using the following preferred algorithm: <OrgID>:<LocalID> Where: • <OrgID> and <LocalID> are separated by a colon ':' • <OrgID> must include a copyrighted, trademarked or otherwise unique name that is owned by the business entity creating or defining the InstanceID, or is a registered ID that is assigned to the business entity by a recognized global authority (This is similar to the <Schema Name>_<Class Name> structure of Schema class names). • <OrgID> must not contain a colon (':'). The first colon to appear in InstanceID must appear between <OrgID> and <LocalID> . • <LocalID> is chosen by the business entity and should not be re-used to identify different underlying (real-world) elements • For DMTF defined instances, the <OrgID> must be set to 'CIM'. If the above preferred algorithm is not used, the defining entity must ensure that the resultant InstanceID is not re-used across any instance IDs produced by this or other providers for this instance's NameSpace.	Implementation dependent value representing unique ID.
AdvertiseTypes	uint16[]	Signifies the advertisement for the profile information. It is used by the advertising services of the WBEM infrastructure to determine what should be advertised, using what mechanisms. The property is an array so that the profile might be advertised using several mechanisms. **Note -** If this property is null/uninitialized, this is equivalent to specifying the value 2 (Not Advertised).	Will have the value 2 (Not Advertised).

TABLE A-26 Properties for Oracle_RegisteredProfile *(Continued)*

Property	Data Type	Description	ILOM Value
RegisteredName	string	The name of this registered profile. Since multiple versions can exist for the same RegisteredName, the combination of RegisteredName, RegisteredOrganization, and RegisteredVersion must uniquely identify the registered profile within the scope of the organization.	Value of supported profile name.
RegisteredOrganization	uint16[]	The organization that defines this profile. The values for this property include: {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ..} Definitions for the above values are as follows: {Other, DMTF, CompTIA, Consortium for Service Innovation, FAST, GGF, INTAP, itSMF, NAC, Northwest Energy Efficiency Alliance, SNIA, TM Forum, The Open Group, ANSI, IEEE, IETF, INCITS, ISO, W3C, OGF, DMTF Reserved}	Will have the value 2 (DMTF).
RegisteredVersion	string	The version of this profile. The string representing the version must be in the form: M + "." + N + "." + U Where: • M - The major version (in numeric form) describing the profile's creation or last modification. • N - The minor version (in numeric form) describing the profile's creation or last modification. • U - The update (for example, errata, patch, and so forth, in numeric form) describing the profile's creation or last modification.	Will have the value "1.0.0".

Oracle_RecordLog

Description:	`Oracle_RecordLog` serves as an aggregation point for log entry objects. It is used to represent the IPMI SEL log. Properties of `Oracle_RecordLog` follow guidelines in IPMI CIM Mapping Guideline.
Inheritance:	`CIM_RecordLog`
Properties:	For a description of the supported properties for the `Oracle_RecordLog` class, see TABLE A-27. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-27), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Record Log

TABLE A-27 Properties for Oracle_RecordLog

Property	Data Type	Description	ILOM Value
InstanceID	string	The InstanceID property is a *key* manadatory property. Within the scope of the instantiating NameSpace, the InstanceID property uniquely identifies an instance of this class. The value of InstanceID should be constructed using the following preferred algorithm: <OrgID>:<LocalID> Where: • <OrgID> and <LocalID> are separated by a colon ':' • <OrgID> must include a copyrighted, trademarked or otherwise unique name that is owned by the business entity creating or defining the InstanceID, or is a registered ID that is assigned to the business entity by a recognized global authority (This is similar to the <Schema Name>_<Class Name> structure of Schema class names). • <OrgID> must not contain a colon (':'). The first colon to appear in InstanceID must appear between <OrgID> and <LocalID> • <LocalID> is chosen by the business entity and should not be re-used to identify different underlying (real-world) elements. • For DMTF defined instances, the <OrgID> must be set to 'CIM'. If the above preferred algorithm is not used, the defining entity must ensure that the resultant InstanceID is not re-used across any instance IDs produced by this or other providers for this instance's NameSpace.	Implementation dependent value representing unique ID.
CurrentNumberOfRecords	UInt64	Current number of records in the Log.	Appropriate value.
ElementName	string	The ElementName property is a user-friendly name. This property allows each instance to define a user-friendly name in addition to its key properties, identity data, and description information. **Note -** The Name property of ManagedSystemElement is also defined as a user-friendly name. But, it is often subclassed to be a *Key*. It is not reasonable that the same property can convey both identity and a user-friendly name, without inconsistencies. Where Name exists and is not a *Key* (such as for instances of LogicalDevice), the same information can be present in both the Name and ElementName properties.	Will hve the value *SEL Log*.

TABLE A-27 Properties for Oracle_RecordLog *(Continued)*

Property	Data Type	Description	ILOM Value
EnabledDefault	uint16[]	An enumerated value indicating an administrator's default or startup configuration for the EnabledState of an element. By default, the element is "Enabled" (value=2). The following values apply: {2, 3, 5, 6, 7, 9, .., 32768..65535} Definitions of the above values are: {Enabled, Disabled, Not Applicable, Enabled but Offline, No Default, Quiesce, DMTF Reserved, Vendor Reserved}	Will be set to default value 2 (Enabled).
EnabledState	uint16[]	Integer enumeration that indicates the enabled and disabled states of an element. It can also indicate the transitions between these requested states. For example, shutting down (value=4) and starting (value=10) are transient states between enabled and disabled. The following values apply: • 0 (Unknown) • 1 (Other) • 2 (Enabled) - The element is or could be executing commands, will process any queued commands, and queues new requests. • 3 (Disabled) - The element will not execute commands and will drop any new requests. • 4 (Shutting Down) - The element is in the process of going to a Disabled state. • 5 (Not Applicable) - The element does not support being enabled or disabled. • 6 (Enabled but Offline) - The element might be completing commands, and will drop any new requests • 7 (Test) - The element is in a test state. • 8 (Deferred) - The element might be completing commands, but will queue any new requests. • 9 (Quiesce) - The element is enabled but in a restricted mode. • 10 (Starting) - The element is in the process of going to an Enabled state. New requests are queued. • 11..32767 (DMTF Reserved) • 32768..65539 (Vendor Reserved)	Appropriate value.

TABLE A-27 Properties for Oracle_RecordLog *(Continued)*

Property	Data Type	Description	ILOM Value
HealthState	uint16[]	Indicates the current health of the element. This attribute expresses the health of this element but not necessarily that of its subcomponents. The following values apply: • 0 (Unknown) - The implementation cannot report on HealthState at this time. • 5 (OK) - The element is fully functional and is operating within normal operational parameters and without error. • 10 (Degraded/Warning) - The element is in working order and all functionality is provided. However, the element is not working to the best of its abilities. For example, the element might not be operating at optimal performance or it might be reporting recoverable errors. • 15 (Minor Failure) - All functionality is available but some might be degraded. • 20 (Major Failure) - The element is failing. It is possible that some or all of the functionality of this component is degraded or not working. • 25 (Critical Failure) - The element is non-functional and recovery might not be possible. • 30 (Non-recoverable Error) - The element has completely failed, and recovery is not possible. All functionality provided by this element has been lost. DMTF has reserved the unused portion of the continuum for additional health states in the future.	Appropriate value.
LogState	uint16[]	LogState is an integer enumeration that indicates the current state of a log represented by CIM_Log subclasses. LogState is to be used in conjunction with the EnabledState property to fully describe the current state of the log. The following text briefly summarizes the various log states: Unknown (0) indicates the state of the log is unknown. Normal (2) indicates that the log is or could be executing logging commands, will process any queued log entries, and will queue new logging requests. Erasing (3) indicates that the log is being erased. Not Applicable (4) indicates the log does not support representing a log state. The following values apply: {0, 2, 3, 4, .., 32768..65535} Definitions for the above values are: {Unknown, Normal, Erasing, Not Applicable, DMTF Reserved, Vendor Reserved}	Appropriate value.
MaxNumberOfRecords	UInt64	Maximum number of records that can be captured in the log. If undefined, a value of zero should be specified.	Appropriate value.

TABLE A-27 Properties for Oracle_RecordLog *(Continued)*

Property	Data Type	Description	ILOM Value
OperationalStatus	uint16[]	The OperationalStatus property indicates the current statuses of the element. Various operational statuses are defined. Many of the enumeration's values are self-explanatory. Enumeration definitions can include any of the following: {Unknown, Other, OK, Degraded, Stressed, Predictive Failure, Error, Non-Recoverable Error, Starting, Stopping, Stopped, In Service, No Contact, Lost Communication, Aborted, Dormant, Supporting Entity in Error, Completed, Power Mode, DMTF Reserved, Vendor Reserved} Values for the enumeration definitions include: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, .., 0x8000..}	Appropriate value.
OverwritePolicy	uint16[]	Integer enumeration that indicates whether the log, represented by the CIM_Log subclasses, can overwrite its entries.Unknown (0) indicates the log's overwrite policy is unknown. Wraps when Full (2) indicates that the log overwrites its entries with new entries when the log has reached its maximum capacity. Never Overwrites (7) indicates that the log never overwrites its entries by the new entries. The following values apply: {0, 2, 7, .., 32768..65535} Definitions for the above values are: {Unknown, Wraps When Full, Never Overwrites, DMTF Reserved, Vendor Reserved}	Will have value 2 (Wraps When Full).
RequestedState	uint16[]	The RequestedState proprety is an integer enumeration that indicates the last requested or desired state for the element, irrespective of the mechanism through which it was requested. The actual state of the element is represented by EnabledState. This property is provided to compare the last requested and current enabled or disabled states. Element definitions include any of the following: {Unknown, Enabled, Disabled, Shut Down, No Change, Offline, Test, Deferred, Quiesce, Reboot, Reset, Not Applicable, DMTF Reserved, Vendor Reserved} Values for the above definitions are as follows: {0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, .., 32768..65535} **Note -** When EnabledState is set to 5 (Not Applicable), then this property has no meaning.	Will bet set to 12 (Not Applicable).

Oracle_ReferencedProfile

Description:	`Oracle_ReferencedProfile` is used to associate an instance of `Oracle_RegisteredProfile` to the instance of `Oracle_RegisteredProfile` representing the Base Server profile. ILOM uses Scoping Class advertisement methodology. See Profile Registration profile for details
Inheritance:	`CIM_ReferencedProfile`
Properties:	For a description of the supported properties for the `Oracle_ReferencedProfile` class, see TABLE A-28. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-28), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Profile Registration

TABLE A-28 Properties for Oracle_ReferencedProfile

Property	Data Type	Description	ILOM Value
`Antecedent`	`Oracle_RegisteredProfile` REF	The `Antecedent` property is a mandatory *key* property. Instance of `Oracle_RegisteredProfile`.	Object path to an instance of `Oracle_RegisteredProfile` .
`Dependent`	`Oracle_RegisteredProfile` REF	The `Dependent` property is a mandatory *key* property. Indicates the `Oracle_RegisteredProfile`.	Object path to the instance of `Oracle_RegisteredProfile` representing the Base Server profile.

Oracle_Sensor

Description:	Represents a hardware component capable of measuring the characteristics of a physical property (for example, the temperature or voltage characteristics of a computer system).
Inheritance:	`CIM_Sensor`
Properties:	For a description of the supported properties for the `Oracle_Sensor` class, see TABLE A-29. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-29), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Sensor

TABLE A-29 Properties for Oracle_Sensor

Property	Data Type	Description	ILOM Value
`CreationClassName`	string	The `CreationClassName` property is a mandatory *key* property. `CreationClassName` indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.	Set to `Oracle_Sensor`.
`DeviceID`	string	The `DeviceID` property is a mandatory *key* property. The `DeviceID` property indicates an address or other identifying information used to uniquely name the `LogicalDevice`.	Set to the NAC name of the sensor.
`SystemCreationClassName`	string	The `SystemCreationClassName` property is a mandatory *key* property. Indicates the `SystemCreationClassName` for the scoping system.	Set to `Oracle_ComputerSystem`.
`SystemName`	string	The `SystemName` property is a mandatory *key* property. Indicates the `SystemName` of the scoping system.	Set to `Oracle_ComputerSystem.Name` of the instance of `Oracle_ComputerSystem` that represents the controllee.

TABLE A-29 Properties for Oracle_Sensor *(Continued)*

Property	Data Type	Description	ILOM Value
`CurrentState`	string	The current state indicated by the sensor. This is always one of the `PossibleStates`.	Value representing current state of the sensor.
`ElementName`	string	The `ElementName` property is a user-friendly name. This property allows each instance to define a user-friendly name in addition to its key properties, identity data, and description information. Note that the `Name` property of `ManagedSystemElement` is also defined as a user-friendly name. But, it is often subclassed to be a Key. It is not reasonable that the same property can convey both identity and a user-friendly name, without inconsistencies. Where `Name` exists and is not a Key (such as for instances of `LogicalDevice`), the same information can be present in both the `Name` and `ElementName` properties.	Will be set to the NAC name of the sensor.
`EnabledDefault`	uint16[]	An enumerated value indicating an administrator's default or startup configuration for the enabled state of an element. By default, the element is *Enabled* (value=2). The following values apply: {2, 3, 5, 6, 7, 9, .., 32768..65535} Definitions for the above values are: {Enabled, Disabled, Not Applicable, Enabled but Offline, No Default, Quiesce, DMTF Reserved, Vendor Reserved}.	Set to default value `2` (Enabled).

TABLE A-29 Properties for Oracle_Sensor *(Continued)*

Property	Data Type	Description	ILOM Value
EnabledState	uint16[]	Integer enumeration that indicates the enabled and disabled states of an element. It can also indicate the transitions between these requested states. For example, shutting down (value=4) and starting (value=10) are transient states between enabled and disabled. The following values apply: • 0 (Unknown) • 1 (Other) • 2 (Enabled) - The element is or could be executing commands, will process any queued commands, and queues new requests. • 3 (Disabled) - The element will not execute commands and will drop any new requests. • 4 (Shutting Down) - The element is in the process of going to a Disabled state. • 5 (Not Applicable) - The element does not support being enabled or disabled. • 6 (Enabled but Offline) - The element might be completing commands, and will drop any new requests. • 7 (Test) - The element is in a test state. • 8 (Deferred) - The element might be completing commands, but will queue any new requests. • 9 (Quiesce) - The element is enabled but in a restricted mode. • 10 (Starting) - The element is in the process of going to an Enabled state. New requests are queued. • 11..32767 (DMTF Reserved) • 32768..65535 (Vendor Reserved)	Will have appropriate value depending on whether the sensor is enabled, disabled or unknown.

TABLE A-29 Properties for Oracle_Sensor *(Continued)*

Property	Data Type	Description	ILOM Value
`HealthState`	uint16[]	Indicates the current health of the element. This attribute expresses the health of this element but not necessarily that of its subcomponents. The following values apply: • 0 (Unknown) - The implementation cannot report on `HealthState` at this time. • 5 (OK) - The element is fully functional and is operating within normal operational parameters and without error. • 10 (Degraded/Warning) - The element is in working order and all functionality is provided. However, the element is not working to the best of its abilities. For example, the element might not be operating at optimal performance or it might be reporting recoverable errors. • 15 (Minor Failure) - All functionality is available but some might be degraded. • 20 (Major Failure) - The element is failing. It is possible that some or all of the functionality of this component is degraded or not working. • 25 (Critical Failure) - The element is non-functional and recovery might not be possible. • 30 (Non-recoverable Error) - The element has completely failed, and recovery is not possible. All functionality provided by this element has been lost. DMTF has reserved the unused portion of the continuum for additional health states in the future.	Appropriate value.
`OperationalStatus`	uint16[]	The `OperationalStatus` property indicates the current statuses of the element. Various operational statuses are defined. Many of the enumeration's values are self-explanatory. Enumeration values can include any of the following: {Unknown, Other, OK, Degraded, Stressed, Predictive Failure, Error, Non-Recoverable Error, Starting, Stopping, Stopped, In Service, No Contact, Lost Communication, Aborted, Dormant, Supporting Entity in Error, Completed, Power Mode, DMTF Reserved, Vendor Reserved} Possible values for the enumeration values include: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, .., 0x8000..}	Appropriate value.

TABLE A-29 Properties for Oracle_Sensor *(Continued)*

Property	Data Type	Description	ILOM Value
PossibleStates	string	Enumerates the string outputs of the sensor. For example, a switch sensor may output the states *On* or *Off*. Another implementation of the switch may output the states *Open*, and *Close*. Another example is a NumericSensor supporting thresholds. This sensor can report the states like *Normal, Upper Fatal*, *Lower Non-Critical,* and so forth. A NumericSensor that does not publish readings and thresholds, but can store this data internally and still report its states.	Appropriate values depending on the type of the sensor.
RequestedState	uint16[]	The RequestedState property is an integer enumeration that indicates the last requested or desired state for the element, irrespective of the mechanism through which it was requested. The actual state of the element is represented by EnabledState. This property is provided to compare the last requested and current enabled or disabled states. Element definitions include any of the following: {Unknown, Enabled, Disabled, Shut Down, No Change, Offline, Test, Deferred, Quiesce, Reboot, Reset, Not Applicable, DMTF Reserved, Vendor Reserved} Values for the above definitions include: {0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, .., 32768..65535} **Note -** When EnabledState is set to 5 ("Not Applicable"), then this property has no meaning. Refer to the DMTF CIM EnabledState property description for explanations of the values in the RequestedState enumeration.	Set to 12 (Not Applicable).

TABLE A-29 Properties for Oracle_Sensor *(Continued)*

Property	Data Type	Description	ILOM Value
SensorType	uint16[]	The type of the sensor, for example, voltage or temperature sensor. If the type is set to *Other*, then the OtherSensorType description can be used to further identify the type, or if the sensor has numeric readings, then the type of the sensor can be implicitly determined by the units. A description of the different sensor types is as follows: • A temperature sensor measures the environmental temperature. • Voltage and current sensors measure electrical voltage and current readings. • A tachometer measures speed/revolutions of a device. For example, a fan device can have an associated tachometer which measures its speed. • A counter is a general purpose sensor that measures some numerical property of a Device. A counter value can be cleared, but it never decreases. • A switch sensor has states like Open orClose, Onor Off, or, Upor Down. • A lock has states of Lockedor Unlocked. • Humidity, smoke detection and air flow sensors measure the equivalent environmental characteristics. • A presence sensor detects the presence of a PhysicalElement. • A power consumption sensor measures the instantaneous power consumed by a managed element. • A power production sensor measures the instantaneous power produced by a managed element such as a power supply or a voltage regulator. • A pressure sensor is used to report pressure. The following values apply: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, .., 32768..65535} Definitions of above values are: {Unknown, Other, Temperature, Voltage, Current, Tachometer, Counter, Switch, Lock, Humidity, Smoke Detection, Presence, Air Flow, Power Consumption, Power Production, Pressure, DMTF Reserved, Vendor Reserved}	Will have appropriate value.

Oracle_SpSystemComponent

Description:	`Oracle_SpSystemComponent` is used to associate the instance of `Oracle_ComputerSystem` representing the controllee and the instance of `Oracle_ComputerSystem` representing the controller.
Inheritance:	`CIM_SystemComponent`
Properties:	For a description of the supported properties for the `Oracle_SpSystemComponent` class, see TABLE A-30. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-30), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Service Processor

TABLE A-30 Properties for Oracle_SpSystemComponent

Property	Data Type	Description	ILOM Value
`GroupComponent`	`CIM_ComputerSystem` REF	The `GroupComponent` property is a *key* mandatory property. Indicates the parent system in the association.	Object path to the instance of `Oracle_ComputerSystem` representing the controllee.
`PartComponent`	`CIM_ComputerSystem` REF	The `PartComponent` property is a *key* mandatory property. Indicates the child element of a system component.	Object path to the instance of `Oracle_ComputerSystem` representing the controller.

Oracle_SystemDevice

Description:	Association that represents an explicit relationship in which logical devices are aggregated by a `ComputerSystem`.
Inheritance:	`CIM_SystemDevice`
Properties:	For a description of the supported properties for the `Oracle_SystemDevice` class, see TABLE A-31. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-31), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	• Base Server • System Memory • Sensors • CPU • Indicator LED

TABLE A-31 Properties for Oracle_SystemDevices

Property	Data Type	Description	ILOM Value
`GroupComponent`	`Oracle_ComputerSystem` REF	The `GroupComponent` property is a *key* mandatory property. Indicates the `Oracle_ComputerSystem`.	Object path to an instance of `Oracle_ComputerSystem` representing the controllee.
`PartComponent`	`CIM_LogicalDevice` REF	The `PartComponent` property is a *key* mandatory property. The `PartComponent` is the `LogicalDevice` that is a component of a system.	Object path to an instance of `CIM_LogicalDevice`.

Oracle_ThresholdIndication

Description:	When the client creates an indication subscription in which the filter indicates one of the following: • `CIM_AlertIndication` and `CIM_AlertIndication.ProbalbleCause` is 52 ("Threshold Crossed") • `CIM_ThresholdIndication` The ILOM CIM sub-system will generate an instance of the `Oracle_ThresholdIndication` class when it notices a sensor crossing a threshold.
Inheritance:	`CIM_ThresholdIndication`
Properties:	For a description of the supported properties for the `Oracle_ThresholdIndication` class, see TABLE A-32. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-32), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	None

TABLE A-32 Properties for Oracle_ThresholdIndication

Property	Data Type	Description	ILOM Value
`AlertingElementFormat`	uint16[]	The format of the `AlertingManagedElement` property is interpretable based upon the value of this property. Values are defined as: • `0` - Unknown. The format is unknown or not meaningfully interpretable by a CIM client application. • `1` - Other. The format is defined by the value of the OtherAlertingElementFormat property. • 2 - CIMObjectPath. The format is a CIMObjectPath, with format <NameSpacePath>:<ClassName>.<Prop1>= "<*Value1*>", <Prop2>="<*Value2*>", and so forth specifying an instance in the CIM Schema. The following values apply: {0, 1, 2} Definitions for the above values are: {Unknown, Other, CIMObjectPath}	Will have the value `2` (CIMObjectPath).

TABLE A-32 Properties for Oracle_ThresholdIndication *(Continued)*

Property	Data Type	Description	ILOM Value
AlertingManagedElement	string	The identifying information of the entity (that is the instance) for which this indication is generated. The property contains the path of an instance, encoded as a string parameter - if the instance is modeled in the CIM Schema. If not a CIM instance, the property contains some identifying string that names the entity for which the Alert is generated. The path or identifying string is formatted per the AlertingElementFormat property.	Will have the string representation of the object path of the sensor that crosses the threshold.
AlertType	uint16[]	Primary classification of the Indication. The following values are defined: • 1 (Other) - Current Indication does not fit into the categories described by this enumeration. • 2 (Communications Alert) - Associated with the procedures and/or processes required to convey information from one point to another. • 3 (Quality of Service Alert) - A degradation or errors in the performance or function of an entity have occurred. • 4 (Processing Error) - A software or processing fault has occurred. • 5 (Device Alert) - An equipment or hardware fault has occurred. • 6 (Environmental Alert) - Refers to an enclosure in which the hardware resides, or other environmental considerations. • 7 (Model Change) - Addresses changes in the information model. For example, it might embed a lifecycle indication to convey the specific model change being alerted. • 8 (Security Alert) - Security violations, detection of viruses, or similar issues have occurred.	Will have the value 6 (Environmental Alert).
Descriptions	string	Short description for the instance.	Appropriate value describing why the indication is generated.
HwComponentObjectPath (Sun-specific)	string	Object path of the associated hardware component.	The object path of an instance of CIM_PhysicalElement.
ObservedValue	string	A string holding the current reading value that exceeds the threshold. This is modeled as a string for universal mapping, similar to the CIM_Sensor properties in the device model.	Appropriate value.

TABLE A-32 Properties for Oracle_ThresholdIndication *(Continued)*

Property	Data Type	Description	ILOM Value
ProbableCause	uint16[]	Enumerated value that describes the probable cause of the situation which resulted in the AlertIndication. The following values apply: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130} Definitions for the above values are: {Unknown, Other, Adapter/Card Error, Application Subsystem Failure, Bandwidth Reduced, Connection Establishment Error,Communications Protocol Error, Communications Subsystem Failure, Configuration/Customization Error, Congestion, Corrupt Data, CPU Cycles Limit Exceeded, Dataset/Modem Error, Degraded Signal, DTE-DCE Interface Error, Enclosure Door Open, Equipment Malfunction, Excessive Vibration, File Format Error, Fire Detected, Flood Detected, Framing Error, HVAC Problem, Humidity Unacceptable, I/O Device Error, Input Device Error, LAN Error, Non-Toxic Leak Detected, Local Node Transmission Error, Loss of Frame, Loss of Signal, Material Supply Exhausted, Multiplexer Problem, Out of Memory, Output Device Error, Performance Degraded, Power Problem, Pressure Unacceptable, Processor Problem (Internal Machine Error), Pump Failure, Queue SizeExceeded, Receive Failure, Receiver Failure, Remote NodeTransmission Error, Resource ator Nearing Capacity, ResponseTime Excessive, RetransmissionRate Excessive, Software Error, Software Program AbnormallyTerminated, Software Program Error (Incorrect Results), Storage Capacity Problem, Temperature Unacceptable, Threshold Crossed, Timing Problem, Toxic Leak Detected, Transmit Failure, Transmitter Failure, Underlying Resource Unavailable, Version MisMatch, Previous AlertCleared, Login Attempts Failed, Software Virus Detected,Hardware Security Breached, Denial of Service Detected, Security Credential MisMatch,	Set to 52 (Threshold Crossed).

TABLE A-32 Properties for Oracle_ThresholdIndication *(Continued)*

Property	Data Type	Description	ILOM Value
		Unauthorized Access, Alarm Received, Loss of Pointer,Payload Mismatch, Transmission Error, Excessive Error Rate, Trace Problem, Element Unavailable,Element Missing, Loss of MultiFrame, Broadcast Channel Failure, Invalid Message Received,Routing Failure, Backplane Failure, Identifier Duplication,Protection Path Failure, Sync Lossor Mismatch, Terminal Problem, Real Time Clock Failure, Antenna Failure, Battery Charging Failure,Disk Failure, Frequency Hopping Failure, Loss of Redundancy, Power Supply Failure, Signal Quality Problem, Battery Discharging, Battery Failure,Commercial Power Problem, FanFailure, Engine Failure, SensorFailure, Fuse Failure, GeneratorFailure, Low Battery, Low Fuel,Low Water, Explosive Gas, High Winds, Ice Buildup, Smoke,Memory Mismatch, Out of CPU Cycles, Software Environment Problem, Software Download Failure, Element Reinitialized, Timeout, Logging Problems, Leak Detected, Protection Mechanism Failure, Protecting Resource Failure, Database Inconsistency,Authentication Failure, Breach of Confidentiality, Cable Tamper,Delayed Information, Duplicate Information, Information Missing, Information Modification,Information Out of Sequence, Key Expired, Non-Repudiation Failure, Out of Hours Activity, Out of Service, Procedural Error,Unexpected Information}	
`ProviderName`	string	The name of the provider generating this indication.	Appropriate value.
`SystemCreationClassName`	string	The `SystemCreationClassName` the scoping system (provider generating this indication).	Will have the value `Oracle_ComputerSystem`.
`SystemName`	string	Indicates the `SystemName` for the scoping system (name for the provider generating this indication).	Will have the value `Oracle_ComputerSystem.Name` of the instance of `Oracle_ComputerSystem` representing the controllee.
`ThresholdIdentifier`	string	Describes the threshold or names the property that represents the threshold, if modeled in the CIM hierarchy. In the latter case, the value should be written as: `<schema name>_ <class name>.<property name>`.	Appropriate value.
`ThresholdValue`	string	Current value of the threshold. This is modeled as a string for universal mapping, similar to the `CIM_Sensor` properties in the device model.	Appropriate value.

Oracle_UseOfLog

Description:	The `Oracle_UseOfLog` is used to associate an instance of a `Oracle_RecordLog` to an instance of the `Oracle_ComputerSystem`, which represents the controllee.
Inheritance:	`CIM_UseOfLog`
Properties:	For a description of the supported properties for the `Oracle_UseOfLog` class, see TABLE A-33. **Note -** For more details about Oracle's Sun-supported properties (described in TABLE A-33), see the DMTF CIM schema, version 2.18.1, at: `http://www.dmtf.org/standards/cim/cim_schema_v2181/`
Profile:	Record Log Base Server

TABLE A-33 Properties for Oracle_UseOfLog

Property	Data Type	Description	ILOM Value
`Antecedent`	`Oracle_RecordLog` REF	The `Antecedent` property is a mandatory *key* property. Instance of `Oracle_RecordLog`	Object path to the instance of `Oracle_RecordLog` representing the IPMI SEL log.
`Dependent`	`Oracle_ComputerSystem` REF	The `Dependent` property is a mandatory *key* property. The `Oracle_ComputerSystem`.	Object path to the instance of `Oracle_ComputerSystem` representing the controllee.

APPENDIX **B**

SNMP Command Examples

This section provides examples of how to use the following SNMP commands:

Topics

Description	Links
Command examples	• "snmpget Command" on page 296 • "snmpwalk Command" on page 296 • "snmpbulkwalk Command" on page 297 • "snmptable Command" on page 298 • "snmpset Command" on page 301 • "snmptrapd Command" on page 302

Note – All command examples given in this section are executed on the SNMP management station, unless instructions indicate otherwise.

Note – The example SNMP commands presented in this appendix are based on the Net-SNMP sample applications and, therefore, will only work as presented if you have Net-SNMP and the Net-SNMP sample applications installed.

Before using SNMP, be sure to install the ILOM MIBs files in the directory where net-snmp tools load MIBs or where your SNMP tool of choice loads MIBs. See the following URL for additional information on net-snmp:

`http://net-snmp.sourceforge.net/wiki/index.php/TUT:Using_and_loading_MIBS`

snmpget Command

snmpget -mALL -v1 -cpublic *snmp_agent_Ip_address* **sysName.0**

As stated in the description of the sysName.0 MIB object in the SNMPv2-MIB, this command returns an administratively assigned name for this managed node. By convention, this is the node's fully-qualified domain name. If the name is unknown, the value returned is the zero-length string.

For example:

```
% snmpget -v2c -cprivate -mALL snmp_agent_Ip_address sysName.0 sysObjectID.0
ilomCtrlDateAndTime.0
SNMPv2-MIB::sysName.0 = STRING: SUNSPHOSTNAME
SNMPv2-MIB::sysObjectID.0 = OID: SUN-ILOM-SMI-MIB::sunILOMSystems
SUN-ILOM-CONTROL-MIB::ilomCtrlDateAndTime.0 = STRING: 2007-12-10,20:33:32.0
```

In addition to the sysName.0 object, this command displays the content of the sysObjectID.0 and the ilomCtrlDateAndTime.0 MIB objects. Notice that the MIB file name is given for each MIB object as part of the reply.

The following descriptions of the MIB objects are taken from the MIB files.

- **sysName** – An administratively assigned name for this managed node. By convention, this is the node's fully-qualified domain name. If the name is unknown, the value is the zero-length string.
- **sysObjectID** – The vendor's authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMI enterprises sub-tree (1.3.6.1.4.1) and provides an easy and unambiguous means for determining 'what kind of box' is being managed.
- **ilomCtrlDataAndTime** – The date and time of the device.

snmpwalk Command

The `snmpwalk` command performs a sequence of chained `GETNEXT` requests automatically. It is a work saving command. Rather than having to issue a series of snmpgetnext requests, one for each object ID, or node, in a sub-tree, you can simply issue one `snmpwalk` request on the root node of the sub-tree and the command gets the value of every node in the sub-tree.

For example:

```
% snmpwalk -mALL -v1 -cpublic snmp_agent_Ip_address system
SNMPv2-MIB::sysDescr.0 = STRING: ILOM machine custom description
SNMPv2-MIB::sysObjectID.0 = OID: SUN-ILOM-SMI-MIB::sunILOMSystems
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (16439826) 1 day, 21:39:58.26
```

```
SNMPv2-MIB::sysContact.0 = STRING: set via snmp test
SNMPv2-MIB::sysName.0 = STRING: SUNSPHOSTNAME
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 72
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (14) 0:00:00.14
SNMPv2-MIB::sysORID.1 = OID: IF-MIB::ifMIB
SNMPv2-MIB::sysORID.2 = OID: SNMPv2-MIB::snmpMIB
SNMPv2-MIB::sysORID.3 = OID: TCP-MIB::tcpMIB
SNMPv2-MIB::sysORID.4 = OID: RFC1213-MIB::ip
SNMPv2-MIB::sysORID.5 = OID: UDP-MIB::udpMIB
SNMPv2-MIB::sysORID.6 = OID: SNMP-VIEW-BASED-ACM-MIB::vacmBasicGroup
SNMPv2-MIB::sysORID.7 = OID: SNMP-FRAMEWORK-MIB::snmpFrameworkMIBCompliance
SNMPv2-MIB::sysORID.8 = OID: SNMP-MPD-MIB::snmpMPDCompliance
SNMPv2-MIB::sysORID.9 = OID: SNMP-USER-BASED-SM-MIB::usmMIBCompliance
SNMPv2-MIB::sysORDescr.1 = STRING: The MIB module to describe generic objects
for network interface sub-layers
SNMPv2-MIB::sysORDescr.2 = STRING: The MIB module for SNMPv2 entities
SNMPv2-MIB::sysORDescr.3 = STRING: The MIB module for managing TCP
implementations
SNMPv2-MIB::sysORDescr.4 = STRING: The MIB module for managing IP and ICMP
implementations
SNMPv2-MIB::sysORDescr.5 = STRING: The MIB module for managing UDP
implementations
SNMPv2-MIB::sysORDescr.6 = STRING: View-based Access Control Model for SNMP.
SNMPv2-MIB::sysORDescr.7 = STRING: The SNMP Management Architecture MIB.
SNMPv2-MIB::sysORDescr.8 = STRING: The MIB for Message Processing and
Dispatching.
SNMPv2-MIB::sysORDescr.9 = STRING: The management information definitions for
the SNMP User-based Security Model.
SNMPv2-MIB::sysORUpTime.1 = Timeticks: (1) 0:00:00.01
SNMPv2-MIB::sysORUpTime.2 = Timeticks: (2) 0:00:00.02
SNMPv2-MIB::sysORUpTime.3 = Timeticks: (2) 0:00:00.02
SNMPv2-MIB::sysORUpTime.4 = Timeticks: (2) 0:00:00.02
SNMPv2-MIB::sysORUpTime.5 = Timeticks: (2) 0:00:00.02
SNMPv2-MIB::sysORUpTime.6 = Timeticks: (2) 0:00:00.02
SNMPv2-MIB::sysORUpTime.7 = Timeticks: (14) 0:00:00.14
SNMPv2-MIB::sysORUpTime.8 = Timeticks: (14) 0:00:00.14
SNMPv2-MIB::sysORUpTime.9 = Timeticks: (14) 0:00:00.14
```

snmpbulkwalk Command

The `snmpbulkwalk` command uses the GETBULK SNMP protocol feature to query for an entire tree of information about a network entity. This command can pack more objects into the packets by specifying "repeaters." As a result, the `snmpbulkwalk` command is faster than the `snmpwalk` command.

Here is example of an `snmpwalk` command with approximate start and end time stamps.

```
% date
Fri Dec 14 12:21:44 EST 2007
% snmpwalk -mALL -v2c -cprivate snmp_agent_Ip_address entPhysicalTable>time3
% date
Fri Dec 14 12:21:53 EST 2007
```

Here is example of an `snmpbulkwalk` command performing the same operation. Notice that the `snmpbulkwalk` command is faster than the `snmpwalk` command.

```
% date
Fri Dec 14 12:40:57 EST 2007
% snmpbulkwalk -mALL -v2c -cprivate snmp_agent_Ip_address entPhysicalTable>time7
% date
Fri Dec 14 12:41:03 EST 2007
```

snmptable Command

The `snmptable` command retrieves the contents of an SNMP table and displays the contents in a tabular format, that is, one table row at a time, such that the resulting output resembles the table being retrieved. This is contrasted with the `snmpwalk` command, which displays the contents of the table one column at a time.

Here is an example of the `snmptable` command:

```
% snmptable -mALL -v2c -cprivate snmp_agent_Ip_address sysORTable
SNMP table: SNMPv2-MIB::sysORTable
sysORID                          sysORDescr                    sysORUpTime
IF-MIB::ifMIB                    The MIB module to describe    0:0:00:00.01
                                 generic objects for network
                                 interface sub-layers.
SNMPv2-MIB::snmpMIB              The MIB module for SN MPv2    0:0:00:00.02
                                 entities.
TCP-MIB::tcpMIB                  The MIB module for managing   0:0:00:00.02
                                 TCP implementations.
RFC1213-MIB::ip                  The MIB module for managing   0:0:00:00.02
                                 IP and ICMP implementations.
UDP-MIB::udpMIB                   The MIB module for managing  0:0:00:00.02
                                 UDP implementations.
SNMP-VIEW-BASED-ACM-             View-based Access Control     0:0:00:00.02
MIB::vacmBasicGroup              Model for SNMP.
```

```
SNMP-FRAMEWORK-                    The SNMP Management          0:0:00:00.14
MIB::snmpFrameworkMIBCompliance    Architecture MIB.
SNMP-MPD-MIB::snmpMPDCompliance    The MIB for Message          0:0:00:00.14
                                   Processing and Dispatching.
SNMP-USER-BASED-SM-                The management information   0:0:00:00.14
MIB::usmMIBCompliance              definitions for the SNMP
                                   User-based Security Model.
```

Note – While the `snmpget`, `snmpgetnext`, and `snmpwalk` command can be used on any type of MIB object, the `snmptable` command can only be used on MIB table objects. If this command is given any other type of object ID, it will be rejected. This restriction applies to a tableEntry object, a table column object, and any object that represents information within a table. Only a MIB table object ID can be used with the `snmptable` command.

In the examples of the `snmptable` command, the `-Ci` and `-Cb` options are used. For example, here is an `snmptable` command with the `-Ci` option:

```
% snmptable -Ci -mALL -v2c -cprivate snmp_agent_IP_address sunPlatFanTable
SNMP table: SUN-PLATFORM-MIB::sunPlatFanTable
index sunPlatFanClass
10              fan
11              fan
17              fan
23              fan
29              fan
30              fan
36              fan
42              fan
```

Here is an example of an `snmptable` command without the `-Ci` option. Notice that the index column is not displayed**:**

```
% snmptable -mALL -v2c -cprivate snmp_agent_Ip_address sunPlatFanTable
SNMP table: SUN-PLATFORM-MIB::sunPlatFanTable
   sunPlatFanClass
               fan
               fan
               fan
               fan
               fan
               fan
               fan
               fan
```

Here is an example of an `snmptable` command with the `-Ci` and `-Cb` options. The output is abbreviated.

```
% snmptable -Ci -Cb -mALL -v2c -cprivate snmp_agent_IP_address entPhysicalTable
SNMP table: ENTITY-MIB::entPhysicalTable
index       Descr                           VendorType      ContainedIn
 1          ? SNMPv2-SMI::zeroDotZero       0               chassis
```

Here is an example of the same `snmptable` command with the `-Ci` option but without the `-Cb` option. Again the output is abbreviated. Notice that the name of the MIB object is repeated on each heading.

```
% snmptable -Ci -mALL -v2c -cprivate snmp_agent_IP_address entPhysicalTable
SNMP table: ENTITY-MIB::entPhysicalTable
index    entPhysicalDescr            entPhysicalVendor   entPhysicalContained
                                     Type                In
 1       ?SNMPv2-SMI::zeroDotZero    0                   chassis
```

Here is another example of an `snmptable` command with both the `-Ci` and `-Cb` options. Notice that the MIB object is not repeated on each heading.

```
% snmptable -Cb -Ci -mALL -v2c -cprivate snmp_agent_IP_address ilomCtrlAlertsTable
SNMP table: SUN-ILOM-CONTROL-MIB::ilomCtrlAlertsTable
 in-   Sever-   Type      Dest-     Dest-     SNMPV   SNMP-     EmailE    Email
dex    ity               ina-      ina-      er-     Communi   ventCl    Event
                         tion-IP   tion-     sion    tyOrUse   assFil    Type
                                   Email             rname     -ter      Filter
1      crit-   email     ?         0.0.0.    v1      public     none      none
       ical                      0
2       dis-   ipmipet   0.0.0.0   ?         v1      public     ?         ?
       able
3       dis-   ipmipet   0.0.0.0   ?         v1      public     ?         ?
       able
4       dis-   ipmipet   0.0.0.0   ?         v1      public     ?         ?
       able
.
.
.
15      dis-   ipmipet   0.0.0.0   ?         v1      public     ?         ?
       able
```

Thus, when you used the `-Cb` option with the `snmptable` command, the table output is easier to read.

Here is an example of an `snmptable` command using version 3 of the SNMP protocol:

```
% snmptable -Cb -Ci -mALL -v3 -aMD5 -utestuser -Apassword -lauthNoPriv
snmp_agent_Ip_address sunPlatPowerSupplyTable
SNMP table: SUN-PLATFORM-MIB::sunPlatPowerSupplyTable
index sunPlatPowerSupplyClass
90              powerSupply
92              powerSupply
96              powerSupply
```

The following `snmptable` command returns an empty table.

```
% snmptable -Cb -Ci -mALL -v2c -cprivate snmp_agent_Ip_address sunPlatBatteryTable
SUN-PLATFORM-MIB::sunPlatBatteryTable: No entries
```

snmpset Command

While the syntax of the `snmpset` command is similar to that of the `snmpget` command, the commands are quite different. The `snmpget` command merely reads the value of the specified object ID, while the `snmpset` command writes the value specified to the object ID. Further, along with the value to be written to the object ID, you must also specify the data type of the object ID in the `snmpset` command because SNMP objects support more than one data type.

The following example shows how use of the `snmpget` and `snmpset` commands together. The sequence of steps is as follows:

1. **Use the `snmpget` command to check to current value of the MIB object.**
2. **Use the `snmpset` command to change the value of the MIB object.**
3. **Use the `snmpget` command to verify that the MIB object was in fact changed to the requested value.**

```
% snmpget -mALL -v2c -cprivate snmp_agent_Ip_address ilomCtrlHttpEnabled.0
SUN-ILOM-CONTROL-MIB::ilomCtrlHttpEnabled.0 = INTEGER: false(2)
% snmpset -mALL -v2c -cprivate snmp_agent_Ip_address ilomCtrlHttpEnabled.0 i 1
SUN-ILOM-CONTROL-MIB::ilomCtrlHttpEnabled.0 = INTEGER: true(1)
% snmpget -mALL -v2c -cprivate snmp_agent_Ip_address ilomCtrlHttpEnabled.0
SUN-ILOM-CONTROL-MIB::ilomCtrlHttpEnabled.0 = INTEGER: true(1)
```

Note that if you try to execute this `snmpset` command using a public community, instead of private, it will not work. This is because the private community has write permission, but the public community does not. The Reason code returned by the command does not make this clear because it simply states that the object is not writable.

Here is an example:

```
% snmpset -mALL -v2c -cpublic snmp_agent_Ip_address ilomCtrlHttpEnabled.0 i 1
Error in packet.
Reason: notWritable (That object does not support modification)
```

snmptrapd Command

snmptrapd is an SNMP application that receives and logs SNMP Trap and Inform messages. Before your system can receive such messages, you must configure the trap daemon to listen for these messages.

To configure a trap daemon, perform the following steps:

1. **Configure an SNMP trap destination.**

   For an example, see "Configuring an snmptrapd Daemon" on page 302.

2. **Start the trap receiver application, snmptrapd.**

   For an example, see "Starting the Trap Daemon" on page 303.

3. **Generate a test trap to verify that traps are being sent by the agent and received by the trap receiver.**

   For an example, see "Testing the Trap Daemon" on page 303.

▼ Configuring an snmptrapd Daemon

The following example shows how to use the `snmpset` command to configure an `snmptrapd` daemon:

```
% snmpset -mALL -v2c -cprivate snmp_agent_Ip_address ilomCtrlAlertSeverity.1 i 2
ilomCtrlAlertType.1 i 2 ilomCtrlAlertDestinationIP.1 a dest_Ip_address
SUN-ILOM-CONTROL-MIB::ilomCtrlAlertSeverity.1 = INTEGER: critical(2)
SUN-ILOM-CONTROL-MIB::ilomCtrlAlertType.1 = INTEGER: snmptrap(2)
SUN-ILOM-CONTROL-MIB::ilomCtrlAlertDestinationIP.1 = IpAddress: dest_Ip_address
```

▼ Starting the Trap Daemon

The following example shows how to use the `snmptrapd` command start a trap daemon:

```
% snmptrapd -mALL -Lo -f -t -OvQ -e -F "%H.%J.%K:%W:%w %q from %A:%V,% %v\n"
2007-11-29 13:21:07 NET-SNMP version 5.2.3 Started.
```

▼ Testing the Trap Daemon

While the daemon is running, log in to the CLI on the host that is running the SNMP agent and type the following command:

```
-> set /SP/alertmgmt/rules testalert=true
```

Note – It is important to test the trap daemon to make sure it is configured properly.

The following screen shows a sample output when a `testalert` trap is received:

```
SUN-ILOM-CONTROL-MIB::ilom.103.2.1.20.0 = STRING: "This is a test trap"
```

Index

L

M

N

P

R

S

T

U