Sun Network QDR InfiniBand Gateway Switch

Command Reference for Firmware Version 2.1

Adobe PostScript™

Contents

Using This Documentation

This command reference provides information about the commands used to administer the Sun Network QDR InfiniBand Gateway Switch from Oracle. This document is written for technicians, system administrators, and users who have advanced experience administrating InfiniBand fabric hardware.

- "Product Notes" on page xxi
- "Related Documentation" on page xxi
- "Feedback" on page xxii
- "Access to Oracle Support" on page xxii

Product Notes

For late-breaking information an known issues about this product, refer to the product notes at:

http://docs.oracle.com/cd/E36256_01

Related Documentation

Documentation	Links
Sun Network QDR InfiniBand Gateway Switch Firmware Version 2.1	http://docs.oracle.com/cd/E36256_01

Documentation	Links
Oracle Solaris 11 OS	http://www.oracle.com/goto/Solaris11/docs
Oracle Integrated Lights Out Manager (ILOM) 3.0	http://docs.oracle.com/cd/E19860-01
All Oracle products	http://docs.oracle.com

Feedback

Provide feedback on this documentation at:

http://www.oracle.com/goto/docfeedback

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs visit if you are hearing impaired.

Understanding Hardware Commands

The hardware commands act upon or monitor the gateway hardware. They are available through the Oracle ILOM CLI interface with the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` targets. See "Linux Shells for Hardware Commands" on page 4.

With the standard Linux shell CLI interface, only the `root` user of the management controller can run *all* of the hardware commands. The format of the hardware commands is as follows:

FabMan@*hostname*-> *command* [*arguments*][*arguments*]...

Command Syntax	Links
`addlagport` *lagname* *connector1* [*connector2* [*...connectorN*]]	"addlagport Command" on page 8
`allowhostconfig` [-h]	"allowhostconfig Command" on page 9
`autodisable` *subcommand* [*connector* *cause*][-h]	"autodisable Command" on page 10
`checkboot`	"checkboot Command" on page 12
`checkpower`	"checkpower Command" on page 13
`checktopomax` [-h][-l][-v]	"checktopomax Command" on page 14
`checkvoltages`	"checkvoltages Command" on page 15
`connector` *name* `present`|`portstate`|`info`|`dump` [-h]	"connector Command" on page 17
`create_ipoib` [-s][-n *if_name*][-m *mask*][-t *if_type*] *p_key* IP_address [-h]	"create_ipoib Command" on page 18
`createfabric` *subcommand* [-h]	"createfabric Command" on page 20
`createlag` *lagname* [-discoverpkey *p_key*][-datasl *sl*][-lacpmode active|passive|static][-distribution MAC-ONLY|IP-ONLY|TCP-IP|Default][-n][*connector1* [*connector2* [*...connectorN*]]][-vniclimit 1|4][-h]	"createlag Command" on page 23

Command Syntax	Links
createvlan *connector*|*lagname* -vlan *vlan_ID* -pkey *p_key* [-n][-h]	
createvnic *connector*|*lagname* <-guid *guid*>|<-host *hostname* -port *port*>|<-node *string* -port *port*> [-mac *mac*][-vlan *vlan_ID*] -pkey *p_key* [-n][-h]	
dcsport [-guid *guid*|-type DCS-gw -ibdev *ibdev*] -port *port*|-connector *connector*|-printconnectors|-printinternal	
delete_ipoib *if_name* | *pkey* [-s][-h]	
deletelag *lagname*	
deletevlan *lagname*|*connector* -vlan *vlan_ID* [-h]	
deletevnic *connector* *vnic_ID*	
dellagport *lagname* *connector1* [*connector2* [...*connectorN*]]	
disablecablelog	
disablegwport *connector*	
disablelagmode [-h]	
disablelinklog	
disablesm	
disableswitchport [--automatic|--reason=*reason*] *connector*|*ibdev* *port*	
disablevnic *connector* *vnic_ID* [-h]	
disallowhostconfig [-h]	
enablecablelog	
enablegwport *connector* [-discoverpkey *p_key*]	
enablelagmode [-h]	
enablelinklog	
enablesm	
enableswitchport [--automatic|--reason=*reason*] *connector*|*ibdev* *port*	
enablevnic *connector* *vnic_ID* [-h]	
env_test	
exit	
fdconfig *subcommand* [-h]	

Command Syntax	Links
`fwverify`	
`generatetopology` *topofile*`|[-h]`	
`getfanspeed`	
`getmaster [-l]`	
`getportcounters` *port*`|`*connector* `[-R]`	
`getportstatus` *connector* `|` *ibdev port*	
`help` *command* `|` *class*	
`listlinkup`	
`localmkeypersistence enable|disable|show|help`	
`matchtopology [-s` *systemname*`]` *topofile* `[-h]`	
`modifyvnic` *connector*`|`*lagname* *vnic_ID* `[-mac` *mac*`][-vlan` *vlan_ID*`][-n][-h]`	
`setcontrolledhandover` *state*`|list`	
`setdefaultgwdiscpkey` *p_key*	
`setgwethport` *connector* `[-mtu` *mtu*`][-txpause` *pause*`][-rxpause` *pause*`][-clear][-h]`	
`setgwinstance` *instance*`|list`	
`setgwsl eoib|ctrl` *level*	
`setgwsystemname` *systemname*`|--list|--clear [-h]`	
`sethostvniclimit` *connector*`|`*lagname* `<-guid` *guid*`>|<-host` *hostname* `-port` *port*`>|<-node` *string* `-port` *port*`> -limit` *number* `-vlanlist` *vlan_ID1* `...` *vlan_IDn*	
`setmsmlocationmonitor` *state*`|list [-h]`	
`setsmmkey` *m_key*`|none|list`	
`setsmpriority` *priority*`|list`	
`setsmrouting fattree|minhop|list`	
`setsubnetprefix` *prefix*`|list`	
`showdisk`	
`showfree [-m][-d][-k]`	
`showfruinfo`	
`showgwconfig`	

Command Syntax	Links
`showgwports [-v][-h]`	
`showioadapters [--withlimit][-h]`	
`showlag [lagname][-h]`	
`showpsufru slot`	
`showsmlog [-h]`	
`showtemps`	
`showtopology [-h]`	
`showunhealthy`	
`showvlan connector [-h]`	
`showvnics`	
`smconfigtest [-l][-h]`	
`smnodes add IP_address [IP_address ...]|delete IP_address [IP_address ...]|list`	
`smpartition subcommand [-h]`	
`smsubnetprotection subcommand [-h]`	
`version`	

Related Information

Linux Shells for Hardware Commands

The preferred method of accessing the Linux shell is through the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Using the `show` command on the `/SYS/Switch_Diag` target opens a restricted Linux shell that enables the `ilom-admin` user, `ilom-operator` user, and users with similar permissions to run diagnostic commands.

Using the `show` command on the `/SYS/Gateway_Mgmt` target opens a different restricted Linux shell that enables the `ilom-admin` user and users with similar permissions to run both diagnostic and gateway management commands.

Using the show command on the /SYS/Fabric_Mgmt target opens still an other restricted Linux shell that enables the ilom-admin user and users with similar permissions to run diagnostic, gateway, and fabric management commands.

Note – The ilom-operator user cannot access the Linux shell from either the /SYS/Gateway_Mgmt or /SYS/Fabric_Mgmt target.

This table lists the hardware commands and their availability from the respective Linux shell targets. Typing the help all command from within the restricted shells lists the commands available to that shell.

Note – Hardware commands that are not listed are unavailable from the /SYS/Switch_Diag, /SYS/Gateway_Mgmt, or /SYS/Fabric_Mgmt targets.

Command	/SYS/Switch_Diag	/SYS/Gateway_Mgmt	/SYS/Fabric_Mgmt
addlagport		Available	Available
allowhostconfig		Available	Available
autodisable			Available
checkboot	Available	Available	Available
checkpower	Available	Available	Available
checktopomax			Available
checkvoltages	Available	Available	Available
connector	Available	Available	Available
create_ipoib			Available
createfabric			Available
createlag		Available	Available
createvlan		Available	Available
createvnic		Available	Available
dcsport	Available	Available	Available
delete_ipoib			Available
deletelag		Available	Available
deletevlan		Available	Available
deletevnic		Available	Available
dellagport		Available	Available

Command	/SYS/Switch_Diag	/SYS/Gateway_Mgmt	/SYS/Fabric_Mgmt
disablecablelog			Available
disablegwport		Available	Available
disablelagmode		Available	Available
disablelinklog			Available
disablesm			Available
disableswitchport			Available
disablevnic		Available	Available
disallowhostconfig		Available	Available
enablecablelog			Available
enablegwport		Available	Available
enablelagmode		Available	Available
enablelinklog			Available
enablesm			Available
enableswitchport			Available
enablevnic		Available	Available
env_test	Available	Available	Available
exit	Available	Available	Available
fdconfig			Available
fwverify	Available	Available	Available
generatetopology			Available
getfanspeed	Available	Available	Available
getmaster	Available	Available	Available
getportcounters	Available	Available	Available
getportstatus	Available	Available	Available
help	Available	Available	Available
listlinkup	Available	Available	Available
localmkeypersistence			Available
matchtopology			Available
modifyvnic		Available	Available
setcontrolledhandover			Available
setdefaultgwdiscpkey		Available	Available

Command	/SYS/Switch_Diag	/SYS/Gateway_Mgmt	/SYS/Fabric_Mgmt
setgwethport		Available	Available
setgwinstance		Available	Available
setgwsl		Available	Available
setgwsystemname		Available	Available
sethostvniclimit		Available	Available
setmsmlocationmonitor			Available
setsmmkey			Available
setsmpriority			Available
setsmrouting			Available
setsubnetprefix			Available
showdisk	Available	Available	Available
showfree	Available	Available	Available
showfruinfo	Available	Available	Available
showgwconfig		Available	Available
showgwports		Available	Available
showioadapters		Available	Available
showlag		Available	Available
showpsufru	Available	Available	Available
showsmlog	Available	Available	Available
showtemps	Available	Available	Available
showtopology	Available	Available	Available
showunhealthy	Available	Available	Available
showvlan		Available	Available
showvnics		Available	Available
smconfigtest			Available
smnodes			Available
smpartition			Available
smsubnetprotection			Available
version	Available	Available	Available

`addlagport` Command

Adds a connector to a LAG.

Syntax

`addlagport` *lagname* *connector1* [*connector2* [*...connectorN*]]

where:

■ *lagname* is the identifier of the LAG

■ *connector1* is the name of the first connector to be added to the LAG (`0A-ETH-1` to `0A-ETH-4` and `1A-ETH-1` to `1A-ETH-4`).

■ *connectorN* is the name of the last connector to be added to the LAG (`0A-ETH-1` to `0A-ETH-4` and `1A-ETH-1` to `1A-ETH-4`).

Description

This hardware command adds an external gateway Ethernet port to an existing LAG. Once assigned to one LAG, a connector cannot also be assigned to an additional LAG.

Note – If a VNIC has been assigned to a connector, that connector cannot be added to a LAG until the respective VNIC has been deleted.

The `addlagport` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to add connector `0A-ETH-3` to the LAG with name `test` using the `addlagport` command.

```
FabMan@gateway_name->addlagport test 0A-ETH-3
Lag updated
FabMan@gateway_name->
```

Related Information

- "dellagport Command" on page 36
- "createlag Command" on page 23
- "enablelagmode Command" on page 48
- "deletevnic Command" on page 35
- "Linux Shells for Hardware Commands" on page 4

`allowhostconfig` Command

Permits host configuration of VNICs.

Syntax

```
allowhostconfig [-h]
```

Description

This hardware command permits unregulated configuration of VNICs on the individual hosts of the InfiniBand fabric, rather than the gateway itself. This methodology is called Host Manual Mode. The [-h] option provides help.

The `allowhostconfig` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to permit unregulated host configuration of VNICs for all hosts in the fabric, with the `allowhostconfig` command.

```
FabMan@gateway_name->allowhostconfig
Stopping Bridge Manager..                               [  OK  ]
Starting Bridge Manager.                                [  OK  ]
FabMan@gateway_name->
```

Related Information

■ "disallowhostconfig Command" on page 45
■ "Linux Shells for Hardware Commands" on page 4

autodisable Command

Automatically disables problematic links.

Syntax

autodisable *subcommand* [*connector cause*][-h]

This hardware command has subcommands that determine its functionality. This table describes the *subcommand*s and provides their syntax.

Subcommand Syntax	Description
add *connector cause*	Adds a *connector* to the autodisable list and disables that connector when the *cause* is fulfilled.
all	Adds all connectors to the autodisable list and disables the connectors when either cause is fulfilled.
del *connector cause*	Removes a *connector* from the autodisable list. The connector isn't disabled regardless of *cause*.
list	Displays the list of autodisabled connectors for each cause.
none	Removes all connectors from the autodisable list. None of the connectors are disabled regardless of the cause.

where:

- *connector* is the name of the connector (`0A` to `15A` and `0B` to `15B`).
- *cause* is the reason to disable the connector:
  - `H` – The error rate is higher than what the InfiniBand specification permits.
  - `L` – The link speed or width is less than optimal (less than 4x QDR).

Description

This hardware command creates an internal list of connectors that are to be automatically disabled, should the links through the connectors exhibit sub-standard performance. The autodisable list has two parts; one for connectors that experience high error rates on their links, the other for connectors that experience suboptimal link speed and width. A connector can be assigned to both parts of the list.

The H cause is configured for an exceptionally high error rate as seen in:

- SNMP traps
- Oracle ILOM event log
- Syslog
- Email alerts

The L cause is configured for a sub-optimal link width or speed. By default, the optimal link speed and width is 4x QDR. Any of these combinations of link speed and width are reason to disable the connector:

- 1x SDR
- 1x DDR
- 1x QDR
- 4x SDR
- 4x DDR

If a connector (or port) is intentionally configured for lesser performance, the connector should be removed from the link speed and width part of the autodisable list, otherwise it would be automatically disabled.

When a connector has been autodisabled, the `listlinkup` command, the Fabric Monitor, and other screen output identify the autodisabled connector with the string `AutomaticHighErrorRate`, `AutomaticBadSpeedOrWidth`, or both.

After autodisabling, the connector is re-enabled with the `enableswitchport --automatic` *port* command.

The `autodisable` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to use the autodisable command to preemptively autodisable connector 2A when there are too many errors present in the link.

```
FabMan@gateway_name->autodisable add 2A H
FabMan@gateway_name->
```

Related Information

- *Gateway Administration*, autodisabling port and connectors
- "enableswitchport Command" on page 51
- "Linux Shells for Hardware Commands" on page 4

checkboot Command

Displays switch chip and BridgeX boot status.

Syntax

```
checkboot
```

Description

This hardware command checks the boot status of the switch chip and of the BridgeX chips. Output is a simplified OK for each.

The checkboot command is available from the /SYS/Switch_Diag, /SYS/Gateway_Mgmt, and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to check the boot status of the switch chip and BridgeX chips with the `checkboot` command.

```
FabMan@gateway_name->checkboot
Switch OK
Bridge-0 OK
Bridge-1 OK
All Internal ibdevices OK
FabMan@gateway_name->
```

Related Information

- "env_test Command" on page 54
- "Linux Shells for Hardware Commands" on page 4

`checkpower` Command

Displays power supply status.

Syntax

```
checkpower
```

Description

This hardware command checks the status of the power supplies. Output is a simplified `OK`.

The `checkpower` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to check the power supply status with the `checkpower` command.

```
FabMan@gateway_name->checkpower
PSU 0 present status: OK
PSU 1 present status: OK
All PSUs OK
FabMan@gateway_name->
```

Related Information

- "checkvoltages Command" on page 15
- "env_test Command" on page 54
- "Linux Shells for Hardware Commands" on page 4

`checktopomax` Command

Displays size of topology.

Syntax

```
checktopomax [-h][-l][-v]
```

Description

This hardware command is a script that verifies if the number of switches and HCAs found in the InfiniBand fabric is less than the maximum values of 48 switch chips and 1500 CAs.

The `checktopomax` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Options

This table describes the options to the checktopomax command and their purposes.

Option	Purpose
-h	Provides help.
-l	Disables writing output to syslog.
-v	Verbose mode.

Example

This example shows how to verify the number of switches and HCAs within your InfiniBand fabric with the checktopomax command.

Note – The output for your InfiniBand fabric will differ from that in the example.

```
FabMan@gateway_name->checktopomax -v
checktopomax: INFO: You have allowed max number of Switch elements to be 48
checktopomax: INFO: You have allowed max number of CA    elements to be 1500
checktopomax: INFO: Number of Switch elements empiric found by application
/usr/sbin/ibnodes: 2
checktopomax: INFO: Number of Ca    elements empiric found by application
/usr/sbin/ibnodes: 7
FabMan@gateway_name->
```

Related Information

checkvoltages Command

Verifies voltages.

Syntax

```
checkvoltages
```

Description

This hardware command displays the internal voltages for the main board. On the left side of the equals sign is the expected voltage. On the right side of the equals sign is the measured voltage. If the difference between the expected voltage and the measured voltage is more than 10%, the cause should be investigated. The command also provides a summary of the voltage conditions.

The `checkvoltages` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to check that voltages are at nominal values with the `checkvoltages` command.

```
FabMan@gateway_name->checkvoltages
Voltage ECB OK
Measured 3.3V Main = 3.30 V
Measured 3.3V Standby = 3.42 V
Measured 12V = 12.06 V
Measured 5V = 5.03 V
Measured VBAT = 3.17 V
Measured 1.0V = 1.01 V
Measured I4 1.2V = 1.22 V
Measured 2.5V = 2.51 V
Measured V1P2 DIG = 1.18 V
Measured V1P2 ANG = 1.18 V
Measured 1.2V BridgeX = 1.22 V
Measured 1.8V = 1.80 V
Measured 1.2V Standby = 1.20 V
All voltages OK
FabMan@gateway_name->
```

Related Information

- "checkpower Command" on page 13
- "env_test Command" on page 54

- "Linux Shells for Hardware Commands" on page 4

`connector` Command

Reads QSFP cable information.

Syntax

`connector` *name* `present|portstate|info|dump [-h]`

where *name* is the name of the connector (`0A` to `15A`, `0A-ETH`, `1A-ETH`, `0B` to `15B`).

Description

This hardware command performs a pass-fail test to verify that an InfiniBand cable is connected to a particular connector and to the switch chip port that the link routes. The command can also read the data registers of the cable and report FRU ID information.

The `connector` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the `connector` command and their purposes.

Option	Purpose
present	Checks for the presence of connector *number*.
portstate	Returns a mapping of the connector to the respective port of the switch chip.
info	Displays FRU ID information.
dump	Displays a raw hexadecimal dump of the FRU ID information.
-h	Provides help.

Example

This example shows how to display the FRU ID information for connector 1A with the connector command.

```
FabMan@gateway_name->connector 1A info
Cable connector 1A present
Indentifier: QSFP
Connector type: Copper pigtail
Vendor: Amphenol
Vendor OUI: 415048
Partnumber: 568400005
Revision: C
Serialnumber: APF08510050019
Date: 081219
FabMan@gateway_name->
```

Related Information

- "env_test Command" on page 54
- "listlinkup Command" on page 71
- "Linux Shells for Hardware Commands" on page 4

create_ipoib Command

Creates an IP over InfiniBand interface.

Syntax

create_ipoib [-s][-n *if_name*][-m *mask*][-t *if_type*] *p_key* IP_address [-h]

where:

- *if_name* is the identifier of the IP interface.
- *mask* is the subnet mask.
- *if_type* is the type of interface (external, internal, or both).
- *p_key* is the partition key (1 to 7fff or default).
- *IP_address* is the IP address of the interface.

Description

This hardware command creates an IP interface, assigns it an IP address, and attaches it to a P_Key value for use over the InfiniBand fabric.

Note – IP interfaces created with the `create_ipoib` command are independent of those implemented through the `fdconfig` command.

This table describes the output of the `create_ipoib -s` command when IP over InfiniBand interfaces have been configured.

Column Heading	Description
Name	Name of the IP interface.
PKey	Partition key of the IP interface.
IP-address	IP address of the interface.
Subnetmask	Subnet mask of the IP interface.
Interface	Name of the IP interface, recognized by the system. Typically displayed as ib*x*, where *x* is instance of the interface, starting with 0.
Interface-type	Type of interface, either external, internal or both.

The `create_ipoib` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Options

This table describes the options to the `create_ipoib` command and their purposes.

Option	Purpose
-s	Displays information about the IP interfaces.
-n	Specifies the interface name.
-pkey	Specifies the partition key.

Option	Purpose
-t	Specifies the interface type:
	• external – Interface is available for external access.
	• internal – Interface is available for internal communication between switch nodes.
	• both – Interface is both external and internal.
-m	Specifies the subnet mask.
-h	Provides help.

Example

This example shows how to create an IP over InfiniBand interface with P_Key 0x0001 and IP address 123.45.67.89 with the create_ipoib command.

```
FabMan@gateway_name->create_ipoib -n newipoib -m 255.255.254.0 -t both 0x0001
123.45.67.89
Create interface:
Name      PKey    IP-address       Subnetmask      Interface  Interface-type
-------------------------------------------------------------------------
newipoib  0x0001 123.45.67.89     255.255.254.0   ib0         both
FabMan@gateway_name->
```

Related Information

- *Gateway Administration*, creating the IPoIB interface
- "delete_ipoib Command" on page 31
- "Linux Shells for Hardware Commands" on page 4

createfabric Command

Correlates fabric elements.

Syntax

createfabric *subcommand* [-h]

This hardware command has subcommands that determine its functionality. This table describes the *subcommand*s and provides their syntax.

Subcommand Syntax	Description
start from-current|empty	Initiates a configuration based upon a currently used configuration or an entirely new one.
add-element|force-add-element -name *hostname* -ilomIp *IP_address* [-redundantIP *IP_address*]	Adds a fabric element to the edited configuration. The force prefix adds the element without validation. Use this prefix when you are configuring an element without verification of bindings.
modify-element|force-modify-element -name *hostname* [-ilomIp *IP_address*][-redundantIP *IP_address*]	The force prefix modifies the element without validation. Use this prefix when you are modifying an element without verification of bindings.
delete-element -name *hostname*	Removes a fabric element from the edited configuration.
complete	Completes the configuration process and commits the edited configuration to become the active configuration.
list active|modified	Displays the active or modified configuration.
abort	Abruptly ends the configuration session. All edited configuration information is lost, and the active configuration remains unchanged.

where:

- *hostname* is the host name of the element.
- *IP_address* is the IP address for the element recognized by Oracle ILOM or IPoIB.

Description

This hardware command correlates the fully qualified domain names, the Oracle ILOM IP address, and redundant IP address of the fabric elements in the InfiniBand fabric into a fabric configuration file. Fabric elements are typically management controllers hosting Subnet Manager instances.

There are two configuration files, the active configuration and the modified configuration. When creating a configuration file, you must initiate the configuration session with the createfabric start command. During the session, you can

create a modified copy of the active configuration or an entirely new configuration. To end the session, you must use the `createfabric complete` command to make the modified configuration the active configuration. Once committed, the active configuration is distributed to all Subnet Managers in the InfiniBand fabric where the management controller's IP addresses are listed in the Subnet Manager nodes (`smnodes`) list.

Like the `smnodes` list, the fabric configuration file must exist in every management controller filesystem. The file contains a list of Oracle ILOM IP addresses and redundant IP addresses of all active management controllers hosting a Subnet Manager in your fabric. The file should have an entry for every Sun Datacenter InfiniBand Switch 36 and Sun Network QDR InfiniBand Gateway Switch that runs a Subnet Manager in your InfiniBand fabric.

Note – If the Subnet Manager nodes of your InfiniBand fabric ever change (disabled, added, and so on), you must update all copies of the fabric configuration file.

This table describes each of the columns of the output of the `createfabric list active` and `createfabric list modified` commands.

Column Heading	Description
Hostname	Name of the element. The host name of the management controller hosting a Subnet Manager.
ILOM IP Addr	IP address of the element, recognized by Oracle ILOM.
Redundant IP Addr	IP address of the element, used by IPoIB.

The `createfabric` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Options

This table describes the options to the `createfabric` command and their purposes.

Option	Purpose
-name	Specifies the element host name.
-ilomIp	Specifies the IP address of the element recognized by Oracle ILOM.
-redundantIp	Specifies a second IP address of the element:
-h	Provides help.

Example

This example shows how to display the current fabric configuration file with the `createfabric` command.

```
FabMan@gateway_name->createfabric list active
Hostname                            ILOM IP Addr          Redundant IP Addr
-------------------------------     -------------------   -----------------
mnm12-gw-1                              123.45.67.89          -
FabMan@gateway_name->
```

Related Information

- *Gateway Administration*, correlating fabric elements
- "Linux Shells for Hardware Commands" on page 4

`createlag` Command

Creates a LAG.

Syntax

```
createlag lagname [-discoverpkey p_key][-datasl sl][-lacpmode
active|passive|static][-distribution
MAC-ONLY|IP-ONLY|TCP-IP|Default][-n][connector1  [connector2
[...connectorN]]][-vniclimit 1|4][-h]
```

where:

- *lagname* is the identifier of the LAG.
- *p_key* is the discovery partition key (1 to 7fff or default).
- *sl* is the data service level.

Note – Use service level 1 (`sl 1`) only for low-latency, high-priority, small-message, low-bandwidth traffic. Use other service levels for regular, high-bandwidth traffic.

- *connector1* is the name of the first connector to be added to the LAG (`0A-ETH-1` to `0A-ETH-4` and `1A-ETH-1` to `1A-ETH-4`).

- *connectorN* is the name of the last connector to be added to the LAG (`0A-ETH-1` to `0A-ETH-4` and `1A-ETH-1` to `1A-ETH-4`).

Description

This hardware command creates a LAG for maximizing Ethernet bandwidth. You specify which connectors are to belong to the LAG.

Note – If a VNIC has been assigned to a connector, that connector cannot be added to a LAG until the respective VNIC has been deleted.

The `createlag` command supports the Link Aggregate Control Protocol (LACP) and permits network devices to auto-negotiate link bundling by sending LACP packets to peers which also implement LACP.

The LAG and associated VNICs are discovered by host drivers in the default or other partition that has been configured with the default discovery P_Key. By defining a LAG-specific discovery P_Key, the visibility is restricted to hosts configured with the same discovery P_Key. The gateway must be a full member of the discovery partition, and the hosts must be at least limited members. The discovery P_Key does not need to be the same as the partition used for data traffic, and different VLANs can still be associated with different partitions for data traffic.

You can create a Global Gateway by not specifying any connectors for the LAG.

Note – You can create a maximum of 16 Global Gateways per gateway.

The `createlag` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the `createlag` command and their purposes.

Option	Purpose
-discoverypkey	Specifies a discovery P_Key, other than default, used by the BridgeX manager. Discovery P_Keys limit the visibility of the LAGs.

Option	Purpose
-datasl	Specifies the data service level for a Global Gateway when no connectors are specified. **Note -** Use service level 1 (sl 1) only for low-latency, high-priority, small-message, low-bandwidth traffic. Use other service levels for regular, high-bandwidth traffic.
-distribution	Specifies how the packet flows are distributed over the connectors in the LAG: • MAC-ONLY – Packet flow distribution is based on the MAC. • IP-ONLY – Packet flow distribution is based on IP. • TCP-IP – Packet flow distribution is based on TCP-IP. • Default – Default packet flow distribution. **Note -** For firmware version 2.1, distribution options MAC-ONLY, IP-ONLY, and TCP-IP map to the Default option.
-lacpmode	Specifies the LACP mode of operation: • active – The ports always send out frames to the peers. • passive – The ports send out frames only in response to received frames from peers. • static – LACP is not used. LAGs are statically configured. This is the default.
-n	Specifies the LAG to be non-persistent. The LAG does not survive a reboot.
-vniclimit	Specifies the maximum number of VNICs for the LAG: • 1 – The default and represents 1024 VNICs. • 4 – Represents 4096 VNICs. **Note -** When set to 4, only one *connector* is specified, either 0A-ETH or 1A-ETH.
-h	Provides help.

Example

This example shows how to create a LAG with name test using the createlag command.

```
FabMan@gateway_name->createlag test 0A-ETH-1 0A-ETH-2
Lag created
FabMan@gateway_name->
```

Related Information

■ "deletelag Command" on page 33

createvlan Command

Creates a VLAN.

Syntax

createvlan *connector*|*lagname* -vlan *vlan_ID* -pkey *p_key* [-n][-h]

where:

- *connector* is the name of the connector (0A-ETH-1 to 0A-ETH-4 and 1A-ETH-1 to 1A-ETH-4).
- *lagname* is the identifier of the LAG.
- *vlan_ID* is the VLAN identifier (NO or 2 to 4094).

Note – Do not use VLAN identifiers 0, 1, or 4095. According to the IEEE 802.1Q specification, VLAN ID 0 is used for priority tag, and VLAN ID 1 is usually reserved for a switch or bridge management VLAN.

- *p_key* is the partition key (1 to 7fff or default).

Description

This hardware command creates a VLAN association for a connector.

Due to hardware limitations for MultiCast groups, there is a maximum of 1000 VLANs. Additionally, do not use VLAN identifiers 0, 1, and 4095. The IEEE 802.1Q specification designates that VLAN ID 0 is used for priority tag, and VLAN ID 1 is reserved for the switch and bridge management VLAN.

The createvlan command is available from the /SYS/Gateway_Mgmt and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the `createvlan` command and their purposes.

Option	Purpose
-vlan	Specifies the VLAN identifier.
-pkey	Specifies the partition key value.
-n	Use for nonpersistent VLANs.
-h	Provides help.

Example

This example shows how to create a VLAN association to connector 1A-ETH-3 with VLAN id 3 and pkey 5 using the `createvlan` command.

```
FabMan@gateway_name->createvlan 1A-ETH-3 -VLAN 3 -PKEY 5
FabMan@gateway_name->
```

Related Information

- "deletevlan Command" on page 34
- "showvlan Command" on page 118
- "Linux Shells for Hardware Commands" on page 4

`createvnic` Command

Creates a VNIC.

Syntax

```
createvnic connector|lagname <-guid guid>|<-host hostname -port
port>|<-node string -port port> [-mac mac][-vlan vlan_ID] -pkey p_key
[-n][-h]
```

where:

- *connector* is the name of the connector (0A-ETH-1 to 0A-ETH-4 and 1A-ETH-1 to 1A-ETH-4).
- *lagname* is the identifier of the LAG
- *guid* is the global unique identifier of the target port on the host associated with the VNIC.
- *hostname* is the host name of the management controller or system name recognized by the BridgeX manager.
- *port* is the port of the host.
- *string* is the character string in the node description field.
- *mac* is the MAC address.

Note – Only even numbers are supported for the most significant byte of the MAC address (unicast).

Note – You must assign a MAC when you create a VNIC. If you have forgotten, you can assign the MAC with the modifyvnic command.

- *vlan_ID* is the VLAN identifier (NO or 2 to 4094).

Note – Do not use VLAN identifiers 0, 1, or 4095. According to the IEEE 802.1Q specification, VLAN ID 0 is used for priority tag, and VLAN ID 1 is usually reserved for a switch or bridge management VLAN.

- *p_key* is the partition key (1 to 7fff or default).

Description

This hardware command creates a VNIC for a specific connector. When a VNIC is created, it is automatically assigned an ID number. You can see this ID number with the showvnics command.

Note – For VNICs created by the createvnic command, the Network Interface names at the host which is set by the DEVICE parameter in the ifcfg-eth*X* (where *X* is identifier), the name should be different from eth0 to eth7, since these names are reserved for devices created and controlled from the host. A suggested name scheme would be eth*gateway_instance_VNIC_ID*, but all names apart for eth0 to eth7 would be valid.

The `createvnic` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the `createvnic` command and their purposes.

Option	Purpose
-guid	Specifies the GUID of the target port on the host associated with the VNIC.
-host	Specifies the host name or system name.
-port	Specifies the port of the host.
-node	Specifies the node description to identify the respective host.
-mac	Specifies the MAC address.
-vlan	Specifies the VLAN identifier.
-pkey	Specifies the partition key value.
-n	Use for nonpersistent VNICs.
-h	Provides help.

Example

This example shows how to create a VNIC with the `default` partition key using the `createvnic` command.

```
FabMan@gateway_name->createvnic 1A-ETH-2 -guid 00:21:28:00:01:3E:93:f7 -mac
02:02:03:00:01:57 -pkey default
VNIC created
FabMan@gateway_name->
```

Related Information

- "deletevnic Command" on page 35
- "disablevnic Command" on page 44
- "enablevnic Command" on page 53
- "showvnics Command" on page 119
- "Linux Shells for Hardware Commands" on page 4

`dcsport` Command

Maps between switch chip ports and QSFP connectors.

Syntax

```
dcsport [-guid guid|-type DCS-gw -ibdev ibdev] -port
port|-connector connector|-printconnectors|-printinternal
```

where:

- *guid* is the global unique identifier.
- *ibdev* is the InfiniBand device name (`Switch`, `Bridge-0-0`, `Bridge-0-1`, `Bridge-1-0`, `Bridge-1-1`)
- *port* is the number of the port (`1` to `36`).
- *connector* is the name of the connector (`0A` to `15A`, `0A-ETH`, `1A-ETH`, `0B` to `15B`).

Note – The subordinate names of the 10GbE connectors, `0A-ETH-1` to `0A-ETH-4` and `1A-ETH-1` to `1A-ETH-4` are also valid *connector* names.

Description

This hardware command displays the mapping between I4 switch chip ports, BridgeX chip ports, and QSFP connectors. You can specify either a *port* or a *connector*.

The `dcsport` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the `dcsport` command and their purposes.

Option	Purpose
-guid	Identifies the GUID of the IB device for mapping.

Option	Purpose
-ibdev	Identifies the name of the IB device for mapping.
-port	Identifies the *port* to provide the connector mapping.
-connector	Identifies the *connector* to provide the port mapping.
-printconnectors	Displays mapping for all connectors.
-printinternal	Displays I4 switch chip to BridgeX chip internal mapping.

Note – If no *guid* or *ibdev* are specified, the command defaults to the local I4 switch chip or BridgeX chips, inferred by the *port* number or *connector* name.

Example

This example shows how to display the mapping for connector 0A-ETH with the dcsport command.

```
FabMan@gateway_name->dcsport -connector 0A-ETH
Connector 0A-ETH maps to:
  0A-ETH-1 Bridge-0-1 port 0A-ETH-1
  0A-ETH-2 Bridge-0-1 port 0A-ETH-2
  0A-ETH-3 Bridge-0-0 port 0A-ETH-3
  0A-ETH-4 Bridge-0-0 port 0A-ETH-4
FabMan@gateway_name->
```

Related Information

- "enablesm Command" on page 50
- "setsmpriority Command" on page 91
- "Linux Shells for Hardware Commands" on page 4

delete_ipoib Command

Deletes an IP over InfiniBand interface.

Syntax

```
delete_ipoib if_name | pkey [-s][-h]
```

where:

- *if_name* is the identifier of the IP interface.
- *p_key* is the partition key (1 to 7fff or default).

Description

This hardware command deletes an IP over InfiniBand interface created with the create_ipoib command.

This table describes the output of the delete_ipoib -s command when IP over InfiniBand interfaces have been configured.

Column Heading	Description
Name	Name of the IP interface.
PKey	Partition key of the IP interface.
IP-address	IP address of the interface.
Subnetmask	Subnet mask of the IP interface.
Interface	Name of the IP interface, recognized by the system. Typically displayed as ib*x*, where *x* is the instance of the interface, starting with 0.
Interface-type	Type of interface, either external, internal, or both.

The delete_ipoib command is available from the /SYS/Fabric_Mgmt Linux shell target of the Oracle ILOM CLI interface.

Options

This table describes the options to the delete_ipoib command and their purposes.

Option	Purpose
-s	Displays information about the IP interfaces.
-h	Provides help.

Example

This example shows how to delete an IP over InfiniBand interface with the name `newipoib` using the `delete_ipoib` command.

```
FabMan@gateway_name->delete_ipoib newipoib
Delete interface:
Name      PKey   IP-address      Subnetmask      Interface Interface-type
--------------------------------------------------------------------------
newipoib  0x0001 123.45.67.89    255.255.254.0   ib0       both
FabMan@gateway_name->
```

Related Information

■ *Gateway Administration*, deleting the IPoIB interface

■ "create_ipoib Command" on page 18

■ "Linux Shells for Hardware Commands" on page 4

deletelag Command

Deletes a LAG.

Syntax

`deletelag` *lagname*

where *lagname* is the identifier of the LAG

Description

This hardware command deletes a LAG and disassociates the connector previously assigned to that LAG.

The `deletelag` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to delete the LAG named `test` with the `deletelag` command.

```
FabMan@gateway_name->deletelag test
LAG test deleted
FabMan@gateway_name->
```

Related Information

- "createlag Command" on page 23
- "disablelagmode Command" on page 40
- "Linux Shells for Hardware Commands" on page 4

`deletevlan` Command

Deletes a VLAN.

Syntax

`deletevlan` *lagname|connector* `-vlan` *vlan_ID* `[-h]`

where:

- *lagname* is the identifier of the LAG
- *connector* is the name of the connector (`0A-ETH-1` to `0A-ETH-4` and `1A-ETH-1` to `1A-ETH-4`).
- *vlan_ID* is the VLAN identifier (`NO` or `2` to `4094`).

Note – Do not use VLAN identifiers `0`, `1`, or `4095`. According to the IEEE 802.1Q specification, VLAN ID 0 is used for priority tag, and VLAN ID 1 is usually reserved for a switch or bridge management VLAN.

Description

This hardware command removes the association of a connector to a VLAN.

The `deletevlan` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the `deletevlan` command and their purposes.

Option	Purpose
`-vlan`	Specifies the VLAN identifier.
`-h`	Provides help.

Example

This example shows how to delete a VLAN id `3` from connector `1A-ETH-3` with the `deletevlan` command.

```
FabMan@gateway_name->deletevlan 1A-ETH-3 -vlan 3
FabMan@gateway_name->
```

Related Information

- "createvlan Command" on page 26
- "showvlan Command" on page 118
- "Linux Shells for Hardware Commands" on page 4

`deletevnic` Command

Deletes a VNIC.

Syntax

`deletevnic` *connector*|*lagname* *vnic_ID*

where:

- *connector* is the name of the connector (`0A-ETH-1` to `0A-ETH-4` and `1A-ETH-1` to `1A-ETH-4`).
- *lagname* is the identifier of the LAG
- *vnic_ID* is the VNIC identifier.

Description

This hardware command deletes a VNIC from a connector.

The `deletevnic` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to delete VNIC `1` on connector 0A-ETH-1 with the `deletevnic` command.

```
FabMan@gateway_name->deletevnic 0A-ETH-1 1
FabMan@gateway_name->
```

Related Information

- "createvnic Command" on page 27
- "disablevnic Command" on page 44
- "enablevnic Command" on page 53
- "showvnics Command" on page 119
- "Linux Shells for Hardware Commands" on page 4

`dellagport` Command

Deletes a connector from a LAG.

Syntax

```
dellagport lagname connector1 [connector2 [...connectorN]]
```

where:

- *lagname* is the identifier of the LAG
- *connector1* is the name of the first connector to be removed from the LAG (`0A-ETH-1` to `0A-ETH-4` and `1A-ETH-1` to `1A-ETH-4`).
- *connectorN* is the name of the last connector to be removed from the LAG (`0A-ETH-1` to `0A-ETH-4` and `1A-ETH-1` to `1A-ETH-4`).

Description

This hardware command removes an external gateway Ethernet port from an existing LAG.

The `dellagport` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to remove connector `0A-ETH-3` from the LAG named `test` with the `dellagport` command.

Note – The output for your InfiniBand fabric will differ from that in the example.

```
FabMan@gateway_name->dellagport test 0A-ETH-3
Lag updated
FabMan@gateway_name->
```

Related Information

- "addlagport Command" on page 8
- "disablelagmode Command" on page 40
- "Linux Shells for Hardware Commands" on page 4

`disablecablelog` Command

Disables logging of cable events.

Syntax

```
disablecablelog
```

Description

This hardware command disables logging of cable events to the Syslog.

The `disablecablelog` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to disable logging of cable events to the Syslog with the `disablecablelog` command.

```
FabMan@gateway_name->disablecablelog
Stopping Environment daemon.                                    [  OK  ]
Starting Environment daemon.                                    [  OK  ]
FabMan@gateway_name->
```

Related Information

- "enablecablelog Command" on page 46
- "disablelinklog Command" on page 41
- "Linux Shells for Hardware Commands" on page 4

`disablegwport` Command

Disables a gateway port.

Syntax

`disablegwport` *connector*

where *connector* is the name of the connector hosting the port (`0A-ETH-1` to `0A-ETH-4`, `0A-ETH`, `1A-ETH-1` to `1A-ETH-4`, and `1A-ETH`).

Description

This hardware command disables a connector and its associated port on the gateway.

The `disablegwport` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to disable the gateway port at connector `0A-ETH-3` with the `disablegwport` command.

```
FabMan@gateway_name->disablegwport 0A-ETH-3
FabMan@gateway_name->
```

Related Information

- "enablegwport Command" on page 47
- "showgwports Command" on page 102
- "disablevnic Command" on page 44
- "Linux Shells for Hardware Commands" on page 4

`disablelagmode` Command

Disables LAG mode.

Syntax

```
disablelagmode [-h]
```

Description

This hardware command disables LAG mode, though the LAGs themselves are not deleted. If LAG mode is re-enabled using the `enablelagmode` command, the LAGs are re-activated. The [-h] option provides help.

The `disablelagmode` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to disable LAG mode with the `disablelagmode` command.

```
FabMan@gateway_name->disablelagmode
Stopping Bridge Manager..                                    [  OK  ]
Starting Bridge Manager.                                     [  OK  ]
FabMan@gateway_name->
```

Related Information

- "enablelagmode Command" on page 48
- "deletelag Command" on page 33
- "Linux Shells for Hardware Commands" on page 4

disablelinklog Command

Disables logging of link events.

Syntax

```
disablelinklog
```

Description

This hardware command disables logging of link events to the Syslog.

The `disablelinklog` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to disable logging of link events to the Syslog with the `disablecablelog` command.

```
FabMan@gateway_name->disablelinklog
Stopping Environment daemon.                                    [  OK  ]
Starting Environment daemon.                                    [  OK  ]
FabMan@gateway_name->
```

Related Information

- "enablelinklog Command" on page 49
- "disablecablelog Command" on page 38
- "Linux Shells for Hardware Commands" on page 4

disablesm Command

Disables the Subnet Manager within the management controller.

Syntax

```
disablesm
```

Description

This hardware command disables the Subnet Manager within the management controller of the gateway. You use this command in the event that a Subnet Manager external to the gateway is preferred.

The `disablesm` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to disable the Subnet Manager with the `disablesm` command.

```
FabMan@gateway_name->disablesm
Stopping partitiond daemon.                                    [  OK  ]
Stopping IB Subnet Manager..                                   [  OK  ]
FabMan@gateway_name->
```

Related Information

- "enablesm Command" on page 50
- "setsmpriority Command" on page 91
- "Linux Shells for Hardware Commands" on page 4

`disableswitchport` Command

Disables a connector or switch chip port.

Syntax

`disableswitchport [--reason=`*reason*`]` *connector*|*ibdev* *port*

where:

- *reason* is the reason for disabling the port, `Blacklist` or `Partition`.
- *connector* is the number of the QSFP connector (`0A` to `15B`).
- *ibdev* is the InfiniBand device name (`Switch`, `Bridge-0-0`, `Bridge-0-1`, `Bridge-1-0`, `Bridge-1-1`)
- *port* is the number of the port (`1` to `36`).

Description

This hardware command disables a QSFP connector and port on the switch chip or a port on the BridgeX chips. The command addresses either the connector or the port on the switch chip or the BridgeX port. The `disableswitchport` command is persistent and survives reboots.

The `--reason` option enables you to use a passphrase to lock the state of the port:

- `Blacklist` – A connector and port pair are identified as being inaccessible because of unreliable operation.
- `Partition` – A connector and port pair are identified as being isolated from the InfiniBand fabric.

Both the `Blacklist` and `Partition` passphrases survive reboot. You unlock these passphrases using the `enableswitchport` command with the `--reason` option.

The `disableswitchport` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to disable and blacklist connector 14A with the
disableswitchport command.

```
FabMan@gateway_name->disableswitchport --reason=Blacklist 14A
Disable Switch port 7 reason: Blacklist
Initial PortInfo:
# Port info: DR path slid 65535; dlid 65535; 0 port 7
LinkState:.......................Down
PhysLinkState:...................Polling
LinkWidthSupported:..............1X or 4X
LinkWidthEnabled:................1X or 4X
LinkWidthActive:.................4X
LinkSpeedSupported:..............2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkSpeedEnabled:................2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkSpeedActive:.................2.5 Gbps
After PortInfo set:
# Port info: DR path slid 65535; dlid 65535; 0 port 7
LinkState:.......................Down
PhysLinkState:...................Disabled
FabMan@gateway_name->
```

Related Information

- "enableswitchport Command" on page 51
- "ibportstate Command" on page 153
- "getportstatus Command" on page 68
- "Linux Shells for Hardware Commands" on page 4

disablevnic Command

Disables a VNIC.

Syntax

disablevnic *connector*|*lagname* *vnic_ID* [-h]

where:

- *connector* is the name of the connector (`0A-ETH-1` to `0A-ETH-4`, `0A-ETH`, `1A-ETH-1` to `1A-ETH-4`, and `1A-ETH`).
- *lagname* is the identifier of the LAG
- *vnic_ID* is the VNIC identifier.

Description

This hardware command disables a VNIC. The `-h` option provides help.

The `disablevnic` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to disable VNIC `2` on connector `0A-ETH-2` with the `disablevnic` command.

```
FabMan@gateway_name->disablevnic 0A-ETH-2 2
FabMan@gateway_name->
```

Related Information

`disallowhostconfig` Command

Restricts host configuration of VNICs.

Syntax

```
disallowhostconfig [-h]
```

Description

This hardware command restricts the configuration of VNICs from the individual hosts of the InfiniBand fabric, effectively disabling Host Manual Mode. Consequently, Gateway Manual Mode becomes the default means of configuring VNICs. The -h option provides help.

The disallowhostconfig command is available from the /SYS/Gateway_Mgmt and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to restrict host configuration of VNICs with the disallowhostconfig command.

```
FabMan@gateway_name->disallowhostconfig
Stopping Bridge Manager..                                      [  OK  ]
Starting Bridge Manager.                                       [  OK  ]
FabMan@gateway_name->
```

Related Information

- "allowhostconfig Command" on page 9
- "Linux Shells for Hardware Commands" on page 4

enablecablelog Command

Enables logging of cable events.

Syntax

```
enablecablelog
```

Description

This hardware command enables logging of cable events to the Syslog.

The `enablecablelog` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to enable logging of cable events to the Syslog with the `enablecablelog` command.

```
FabMan@gateway_name->enablecablelog
Stopping Environment daemon.                              [  OK  ]
Starting Environment daemon.                              [  OK  ]
FabMan@gateway_name->
```

Related Information

- "enablecablelog Command" on page 46
- "enablelinklog Command" on page 49

`enablegwport` Command

Enables a gateway port.

Syntax

`enablegwport` *connector* [`-discoverpkey` *p_key*]

where:

- *connector* is the name of the connector hosting the port (`0A-ETH-1` to `0A-ETH-4`, `0A-ETH`, `1A-ETH-1` to `1A-ETH-4`, and `1A-ETH`).
- *p_key* is the partition key (`1` to `7fff` or `default`).

Description

This hardware command enables a previously disabled connector and its associated port on the gateway. The `-discoverpkey` option specifies a discovery P_Key, other than default, used by the BridgeX manager.

The `enablegwport` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to enable the gateway port at connector `0A-ETH-3` with the `enablegwport` command.

```
FabMan@gateway_name->enablegwport 0A-ETH-3
FabMan@gateway_name->
```

Related Information

■ "disablegwport Command" on page 39
■ "showgwports Command" on page 102
■ "enablevnic Command" on page 53
■ "Linux Shells for Hardware Commands" on page 4

`enablelagmode` Command

Enables LAG mode.

Syntax

```
enablelagmode [-h]
```

Description

This hardware command enables LAG mode. By default, LAG mode is disabled. LAGs are a means of effectively increasing the bandwidth of an Ethernet network by grouping ports. To successfully run the `enablelagmode` command, the BXOFED software used by the gateway and hosts must be version 1.3.9 or later. The [-h] option to the `enablelagmode` command provides help.

Note – LAG mode must be enabled on all InfiniBand gateways in the fabric before LAGs are created.

The enablelagmode command is available from the /SYS/Gateway_Mgmt and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to enable LAG mode with the enablelagmode command.

```
FabMan@gateway_name->enablelagmode
VNICs for host nodes with EoIB driver version that do not support LAG will go
down if LAG mode is enabled.
Do you still want to enable LAG(y/n)?y
Stopping Bridge Manager..                              [  OK  ]
Starting Bridge Manager.                               [  OK  ]
FabMan@gateway_name->
```

Related Information

- "disablelagmode Command" on page 40
- "createlag Command" on page 23
- "Linux Shells for Hardware Commands" on page 4

enablelinklog Command

Enables logging of link events.

Syntax

enablelinklog

Description

This hardware command enables logging of link events to the Syslog.

The `enablelinklog` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to enable logging of link events to the Syslog with the `enablelinklog` command.

```
FabMan@gateway_name->enablelinklog
Stopping Environment daemon.                              [  OK  ]
Starting Environment daemon.                              [  OK  ]
FabMan@gateway_name->
```

Related Information

- "disablelinklog Command" on page 41
- "enablecablelog Command" on page 46
- "Linux Shells for Hardware Commands" on page 4

enablesm Command

Enables the Subnet Manager within the management controller.

Syntax

```
enablesm
```

Description

This hardware command enables the Subnet Manager on the management controller.

The `enablesm` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to enable the Subnet Manager with the `enablesm` command.

```
FabMan@gateway_name->enablesm
Starting IB Subnet Manager.                                     [  OK  ]
Starting partitiond daemon.                                     [  OK  ]
FabMan@gateway_name->
```

Related Information

- "disablesm Command" on page 42
- "setsmpriority Command" on page 91
- "Linux Shells for Hardware Commands" on page 4

`enableswitchport` Command

Enables a connector or switch chip port.

Syntax

enableswitchport [--automatic|--reason=*reason*] *connector*|*ibdev port*

where:

- *reason* is the reason for disabling the port, `Blacklist` or `Partition`.
- *connector* is the number of the QSFP connector (`0A` to `15B`).
- *ibdev* is the InfiniBand device name (`Switch`, `Bridge-0-0`, `Bridge-0-1`, `Bridge-1-0`, `Bridge-1-1`)
- *port* is the number of the port (`1` to `36`).

Description

This hardware command enables a QSFP connector and port on the switch chip or a port on the BridgeX chips. The command addresses either the connector or the port on the switch chip or the BridgeX port.

The `--reason` option enables you to use the `Blacklist` or `Partition` passphrases to unlock the state of the port as locked using the `disableswitchport` command.

The `enableswitchport` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Options

This table describes the options to the `enableswitchport` command and their purposes.

Option	Purpose
`--automatic`	Specifies an autodisabled port for re-enabling.
`--reason`	Specifies the `Blacklist` or `Partition` passphrases to unlock the state of the port as locked using the `disableswitchport` command.

Example

This example shows how to enable and de-blacklist connector 14A with the `enableswitchport` command.

```
FabMan@gateway_name->enableswitchport --reason=Blacklist 14A
Enable Switch port 7
Initial PortInfo:
# Port info: DR path slid 65535; dlid 65535; 0 port 7
LinkState:......................Down
PhysLinkState:...................Disabled
LinkWidthSupported:..............1X or 4X
LinkWidthEnabled:................1X or 4X
LinkWidthActive:.................4X
LinkSpeedSupported:..............2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkSpeedEnabled:................2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkSpeedActive:.................2.5 Gbps
After PortInfo set:
# Port info: DR path slid 65535; dlid 65535; 0 port 7
```

```
LinkState:........................Down
PhysLinkState:....................Polling
FabMan@gateway_name->
```

Related Information

- "disableswitchport Command" on page 43
- "ibportstate Command" on page 153
- "getportstatus Command" on page 68
- "Linux Shells for Hardware Commands" on page 4

enablevnic Command

Enables a VNIC.

Syntax

enablevnic *connector|lagname vnic_ID* [-h]

where:

- *connector* is the name of the connector (0A-ETH-1 to 0A-ETH-4, 0A-ETH, 1A-ETH-1 to 1A-ETH-4, and 1A-ETH).
- *lagname* is the identifier of the LAG
- *vnic_ID* is the VNIC identifier.

Description

This hardware command enables a previously disabled VNIC. The -h option provides help.

The enablevnic command is available from the /SYS/Gateway_Mgmt and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to enable VNIC 2 on connector 0A-ETH-2 with the enablevnic command.

```
FabMan@gateway_name->enablevnic 0A-ETH-2 2
FabMan@gateway_name->
```

Related Information

- "createvnic Command" on page 27
- "deletevnic Command" on page 35
- "disablevnic Command" on page 44
- "showvnics Command" on page 119
- "Linux Shells for Hardware Commands" on page 4

env_test Command

Displays environmental status.

Syntax

```
env_test
```

Description

This hardware command performs a series of hardware and environmental tests of the gateway. This command is an amalgamation of these commands:

- checkpower
- checkvoltages
- showtemps
- getfanspeed
- connector
- checkboot

The command output provides voltage and temperature values, pass-fail results, and error messages.

The env_test command is available from the /SYS/Switch_Diag, /SYS/Gateway_Mgmt, and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display the hardware and environmental status of the gateway with the env_test command.

```
FabMan@gateway_name->env_test
Environment test started:
Starting Environment Daemon test:
Environment daemon running
Environment Daemon test returned OK
Starting Voltage test:
Voltage ECB OK
Measured 3.3V Main = 3.28 V
Measured 3.3V Standby = 3.37 V
Measured 12V = 12.06 V
Measured 5V = 5.03 V
Measured VBAT = 3.25 V
Measured 1.0V = 1.01 V
Measured I4 1.2V = 1.22 V
Measured 2.5V = 2.51 V
Measured V1P2 DIG = 1.18 V
Measured V1P2 ANG = 1.18 V
Measured 1.2V BridgeX = 1.22 V
Measured 1.8V = 1.80 V
Measured 1.2V Standby = 1.20 V
Voltage test returned OK
Starting PSU test:
PSU 0 present
PSU 1 present
PSU test returned OK
Starting Temperature test:
Back temperature 30.5
Front temperature 30.6
SP temperature 34.1
Switch temperature 48, maxtemperature 49
Bridge-0 temperature 40, maxtemperature 41
Bridge-1 temperature 45, maxtemperature 45
Temperature test returned OK
Starting FAN test:
```

```
Fan 0 not present
Fan 1 running at rpm 12075
Fan 2 running at rpm 11960
Fan 3 running at rpm 12075
Fan 4 not present
FAN test returned OK
Starting Connector test:
Connector test returned OK
Starting Onboard ibdevice test:
Switch OK
Bridge-0 OK
Bridge-1 OK
All Internal ibdevices OK
Onboard ibdevice test returned OK
Starting SSD test:
SSD test returned OK
Environment test PASSED
FabMan@gateway_name->
```

Related Information

- "checkboot Command" on page 12
- "checkpower Command" on page 13
- "checkvoltages Command" on page 15
- "connector Command" on page 17
- "getfanspeed Command" on page 64
- "showtemps Command" on page 113
- "Linux Shells for Hardware Commands" on page 4

exit Command (Hardware)

Terminates the session.

Syntax

```
exit
```

Description

This hardware command immediately terminates the session with the management controller.

The `exit` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to terminate the session with the `exit` command.

```
FabMan@gateway_name->exit
Connection to 123.45.67.89 closed.
FabMan@gateway_name->
```

Related Information

- *Gateway Remote Management*, `exit` command
- "Linux Shells for Hardware Commands" on page 4

`fdconfig` Command

Defines fabric elements monitored through the FabricMIB.

Syntax

```
fdconfig subcommand [-h]
```

This hardware command has subcommands that determine its functionality. This table describes the *subcommand*s and provides their syntax.

Subcommand Syntax	Description
`start-fabric-config from-current`	Initiates a new configuration based upon a currently used configuration.
`start-fabric-config empty`	Initiates a new original configuration.
`define-fabric-name` *name*	Defines a fabric (or configuration) name.
`define-element -name` *name* `-ip` *IP_address* `-type` *type* `-role` *role1* [`-role` *role2*]	Defines an element of the fabric.
`redefine-element -name` *name* [`-ip` *IP_address*|`-type` *type*|`-role` *role1* [`-role` *role2*]]	Modifies an element of the fabric.
`remove-element -name` *name*	Removes an element from the fabric.
`complete-fabric-config`	Finalizes the configuration.
`list-current-fabric-config`	Displays the current configuration.
`list-in-progress-fabric-config`	Displays the configuration being modified.
`abort`	Abruptly ends the configuration session. All modified configuration information is lost, and the active configuration remains unchanged.

where:

- *name* is the host name of the element as configured by Oracle ILOM.
- *IP_address* is the IP address of an element.
- *type* is the identifier of the type of element:
    - `sw36` – Sun Datacenter InfiniBand Switch 36.
    - `sw36gw` – Sun Network QDR InfiniBand Gateway Switch.
- *role* is the purpose of the element:
    - `gateway` – The element acts as a gateway.
    - `spineSwitch` – The element acts as a spine switch.
    - `leafSwitch` – The element acts as a leaf switch.
    - `unknown` – The role of the element is unknown.

Description

This hardware command configures a list of InfiniBand devices expected to be in the fabric. The Fabric Director node list file is used by the Fabric Directors to compile fabric MIBs that describe the InfiniBand fabric. For the file, InfiniBand devices are referred to as elements, and each internal element hosts a Fabric Director. Elements are switch chips with these roles:

- gateway
- spine switch
- leaf switch

By default, Fabric Directors run in nonmaster mode. Only when a fabric element's Subnet Manager becomes the master Subnet Manager, does the fabric element's corresponding Fabric Director become the master Fabric Director.

This table describes each of the columns of the output.

Column Heading	Description
Name	Host name of the fabric element as configured by Oracle ILOM.
IP Addr	The IP address of the fabric element.
Type	The platform of the fabric element: • sw36 – Sun Datacenter InfiniBand Switch 36 • sw36gw – Sun Network QDR InfiniBand Gateway Switch
Role(s)	The role of the fabric element: • gateway • spine switch • leaf switch • unknown
isMaster	Identifies elements which are not the master Fabric Director.
Fabricname	Name of the fabric or configuration.

The fdconfig command is available from the /SYS/Fabric_Mgmt Linux shell target of the Oracle ILOM CLI interface.

Options

This table describes the options to the fdconfig command and their purposes.

Option	Purpose
-name	Specifies the name of the fabric or element.
-ip	Specifies the IP address of the fabric element.
-type	Specifies the type of element.
-role	Specifies the role of the element.
-h	Provides help.

Example

This example shows how to list the current fabric configuration with the fdconfig command.

Note – The output for your InfiniBand fabric will differ from that in the example.

```
FabMan@gateway_name->fdconfig list-current-fabric-config
Name            IP Addr          Type    Role(s)          isMaster  Fabricname
--------------  ---------------- ------  ---------------  --------  ----------
primary         123.45.67.89     sw36gw  gateway                    test
secondary       123.45.67.90     sw36    leafSwitch       no        test
FabMan@gateway_name->
```

Related Information

■ "Linux Shells for Hardware Commands" on page 4

fwverify Command

Checks firmware integrity.

Syntax

fwverify

Description

This hardware command checks if the firmware installed is corrupted or has been tampered with. The command first makes a comparison of the installed RPM packages to a predefined list of what RPM packages should be installed for the given firmware version. This list was generated at the time of the firmware image build. The `fwverify` command then performs a verification for each installed package using the `rpm --verify` command.

Should the command find a missing, corrupt, or additional package or file, the command will display the package or file name, and its location in the filesystem. The conditions of the error are displayed as an 8-character string of flags, **SM5DLUGT**, where:

- **S** – File size differs.
- **M** – Mode (including permissions and file type) differs.
- **5** – MD5 sum differs.
- **D** – Device major/minor number mis-match.
- **L** – readlink(2) path mis-match.
- **U** – User ownership differs.
- **G** – Group ownership differs.
- **T** – mtime differs.
- **.** – Flag was not set (no error).

Note – The `fwverify` command might indicate that files in the `/config/fwutils` directory have changed. The error can safely be ignored.

The `fwverify` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to verify the firmware integrity with the `fwverify` command.

```
FabMan@gateway_name->fwverify
Checking all present packages:
...................................................................... OK
Checking if any packages are missing:
...................................................................... OK
```

```
Verifying installed files:
.................................................................... FAILED
* Package nm2gw-phs-2.0.5-1.i386:
S.5....T   /etc/init.d/dcs
.
.
.
FabMan@gateway_name->
```

In this example, within the nm2gw-phs-2.0.5-1.i386 RPM package, the
/etc/init.d/dcs file size differs, the MD5 sum differs, and the time differs

This example shows a successful verification.

```
FabMan@gateway_name->fwverify
Checking all present packages:
........................................................................ OK
Checking if any packages are missing:
........................................................................ OK
Verifying installed files:
........................................................................ OK
Checking FW Coreswitch:
 FW Version: 7.4.2200 OK
 PSID: SUNX2826_I40_002 OK
 Verifying image integrity OK

Checking FW Bridge-0:
 FW Version: 8.6.2010 OK
 PSID: SUNX2826_BX0_006 OK
 Verifying image integrity OK

Checking FW Bridge-1:
 FW Version: 8.6.2010 OK
 PSID: SUNX2826_BX1_006 OK
 Verifying image integrity OK
FabMan@gateway_name->
```

Related Information

- "version Command (Hardware)" on page 134
- "Linux Shells for Hardware Commands" on page 4

`generatetopology` Command

Creates topology file.

Syntax

`generatetopology` *topofile*|`[-h]`

where *topofile* is the filename and path to the topology file.

Description

This hardware command creates a topology file of the InfiniBand fabric. This file is not compatible with the topology file created by the `ibnetdiscover` command. The `generatetopology` command is used with the `matchtopology` and `showtopology` commands to determine changes in the InfiniBand fabric. The `-h` option provides help.

The `generatetopology` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to create a topology file called `Topo.conf` with the `generatetopology` command.

```
FabMan@gateway_name->generatetopology AUG22.topo
It will take some time to generate a topology file. Please wait!
topo.conf file is created. Will now start generating the topo file
Wrote Topology file: AUG22.topo
FabMan@gateway_name->
```

This is a portion of a topology file created by the `generatetopology` command.

```
SUNIBQDRGW mnm34-98 Bridge 0
  BX1/P1 -4x-10G-> I4_GENERIC mnm34-98 P4
      -4x-10G-> I4_GENERIC mnm34-98 P3
  BX3/P1 -4x-10G-> I4_GENERIC mnm34-98 P2
      -4x-10G-> I4_GENERIC mnm34-98 P1

DEV26428_02P mnm34-55_HCA-1
  P1 -4x-10G-> SUNIBQDRGW mnm34-97 C-7A
  P2 -4x-10G-> I4_GENERIC mnm34-98 P33

SUNIBQDRGW mnm34-97
  A-SW/P2 -4x-10G-> SUNIBQDRGW mnm34-97 BX3/P1
  A-SW/P4 -4x-10G-> SUNIBQDRGW mnm34-97 BX1/P1
  C-0B -4x-10G-> I4_GENERIC mnm34-98 P21
  C-0A -4x-10G-> I4_GENERIC mnm34-98 P20
  C-1B -4x-10G-> I4_GENERIC mnm34-98 P19
  C-1A -4x-10G-> I4_GENERIC mnm34-98 P22
  C-2B -4x-10G-> I4_GENERIC mnm34-98 P24
  C-2A -4x-10G-> I4_GENERIC mnm34-98 P23
.
.
.
DEV26428_02P mnm34-60_HCA-1
  P1 -4x-10G-> SUNIBQDRGW mnm34-97 C-8A
  P2 -4x-10G-> I4_GENERIC mnm34-98 P31

# Created from mnm34-98 at Thu Aug 19 13:32:25 PDT 2010
```

Related Information

- "matchtopology Command" on page 74
- "showtopology Command" on page 114
- "ibnetdiscover Command" on page 146
- "Linux Shells for Hardware Commands" on page 4

getfanspeed Command

Displays fan speed.

Syntax

```
getfanspeed
```

Description

This hardware command displays the speed of the fans. The command also indicates if the fan is not present or has stopped.

The `getfanspeed` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display fan speeds with the `getfanspeed` command.

```
FabMan@gateway_name->getfanspeed
Fan 0 not present
Fan 1 running at rpm 12075
Fan 2 running at rpm 11960
Fan 3 running at rpm 12075
Fan 4 not present
FabMan@gateway_name->
```

Related Information

- "env_test Command" on page 54
- "Linux Shells for Hardware Commands" on page 4

`getmaster` Command

Displays primary (or Master) Subnet Manager node information.

Syntax

```
getmaster [-l]
```

Description

This hardware command returns information about the node that hosts the primary (or master) Subnet Manager of the InfiniBand fabric. The -l option provides a short historical list of Subnet Manager activity.

The `getmaster` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display information about the node that hosts the master Subnet Manager with the `getmaster` command.

```
FabMan@gateway_name->getmaster -l
Local SM enabled and running, state MASTER
SM monitoring history:
20121212 15:05:05 whereismaster stopped
20121212 15:09:03 whereismaster started
20121212 15:09:03 Master SubnetManager on sm lid 1 sm guid 0x3baabba015c0a0 :
SUN IB QDR GW switch o4nm2-gw-4 10.172.144.72
20121212 15:11:31 No Master SubnetManager seen in the system
.
.
.
FabMan@gateway_name->
```

Related Information

■ "Linux Shells for Hardware Commands" on page 4

getportcounters Command

Displays port counters.

Syntax

`getportcounters` *port*|*connector* `[-R]`

where:

- *port* is the number of the port (`1` to `36`).
- *connector* is the name of the connector (`0A` to `15A`, `0A-ETH`, `1A-ETH`, `0B` to `15B`).

Note – The subordinate names of the 10 GbE connectors, `0A-ETH-1` to `0A-ETH-4` and `1A-ETH-1` to `1A-ETH-4` are also valid *connector* names.

Description

This hardware command returns the error and data throughput counters of the specified *port* of the switch chip or *connector* of the gateway. If a *port* is specified, then the output is for InfiniBand counters. Similarly, if the *connector* specified is `0A` to `15B`, the output is also for InfiniBand counters. If the *connector* specified is `0A-ETH`, `1A-ETH`, or a subordinate connector, the output is for 10GbE counters.

The `-R` option clears the counters for the specified *port* or *connector*.

The `getportcounters` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display the status of connector `1A-ETH-1` with the `getportcounters` command.

```
FabMan@gateway_name->getportcounters 1A-ETH-1
Port counters for connector 1A-ETH-1 Bridge-1 port Bridge-1-2
RX bytes.........................8832
RX packets.......................69
RX Jumbo packets.................0
RX unicast packets...............0
RX multicast packets.............69
RX broadcast packets.............0
RX no buffer.....................0
RX CRC...........................0
RX runt..........................0
```

```
RX errors........................0
TX bytes........................8832
TX packets.......................69
TX Jumbo packets.................0
TX unicast packets...............0
TX multicast packets............69
TX broadcast packets............0
TX errors........................0
FabMan@gateway_name->
```

`getportstatus` Command

Displays port status.

Syntax

`getportstatus` *connector│ibdev port*

where:

- *connector* is the name of the connector (`0A` to `15A`, `0A-ETH`, `1A-ETH`, `0B` to `15B`).

Note – The subordinate names of the 10GbE connectors, `0A-ETH-1` to `0A-ETH-4` and `1A-ETH-1` to `1A-ETH-4`, are also valid *connector* names.

- *ibdev* is the InfiniBand device name (`Switch`, `Bridge-0-0`, `Bridge-0-1`, `Bridge-1-0`, `Bridge-1-1`)
- *port* is the number of the port (`1` to `36`).

Description

This hardware command returns the status of the specified *connector* or *port* of the switch chip.

The `getportstatus` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display the status of connector 9A with the `getportstatus` command.

Note – The parameters with the string `Active` indicate the current conditions.

```
FabMan@gateway_name->getportstatus 9A
Port status for connector 9A Switch Port 14
Adminstate:.....................Enabled
LinkWidthEnabled:...............1X or 4X
LinkWidthSupported:.............1X or 4X
LinkWidthActive:................4X
LinkSpeedSupported:.............2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkState:......................Active
PhysLinkState:..................LinkUp
LinkSpeedActive:................10.0 Gbps
LinkSpeedEnabled:...............2.5 Gbps or 5.0 Gbps or 10.0 Gbps
FabMan@gateway_name->
```

This example shows how to display the status of connector 0A-ETH-1 with the `getportstatus` command.

```
FabMan@gateway_name->getportstatus 0A-ETH-1
Port status for connector 0A-ETH-1 Bridge-0 Port Bridge-0-2
Adminstate......................Enabled
State...........................Up
Link state......................Up
Protocol........................Ethernet
Link mode.......................XFI
Speed...........................10Gb/s
MTU.............................9600
Tx pause........................Global
Rx pause........................Global
FabMan@gateway_name->
```

help Command (Hardware)

Displays help for a command.

Syntax

```
help command | class
```

where:

- *command* is the command for which you need help.
- *class* is the category of commands.

Description

This hardware command provides help information for supported commands. Typically, the same information is provided using the -h option.

The help command is available from the /SYS/Switch_Diag, /SYS/Gateway_Mgmt, and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to get help with the help command.

```
FabMan@gateway_name->help
List of avaliable classes:
general  -- General commands
diag     -- Switch diagnostics commands
ibdiag   -- Infiniband diagnostics commands
sm       -- Subnet Managager administration commands
other    -- Other managagement commands
all      -- Show all commands
Type "help" followed by a class name for a list of commands in that class.
Type "help" followed by a command name for description and usage of the command.
FabMan@gateway_name->
```

Related Information

- *Gateway Remote Management*, help command
- "Linux Shells for Hardware Commands" on page 4

`listlinkup` Command

Displays links presence.

Syntax

```
listlinkup [link][peer][internal|all|connected]
```

Description

This hardware command lists the presence of links and the up-down state of the associated ports on the switch chip. The `listlinkup` command also accepts these arguments:

- `link` – Displays additional information about the links.
- `peer` – Displays additional information about the peer ports.
- `internal` – Displays only links between the I4 switch chip and the BridgeX chips.
- `all` – Displays all link types.
- `connected` – Displays only links that are connected to other nodes.

The `listlinkup` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display comprehensive information about all link presence, associated ports, link state, and information about the peers with the `listlinkup` command.

```
FabMan@gateway_name->listlinkup link peer all
Connector  0A Not present
Connector  1A Not present
Connector  2A Not present
Connector  3A Present <-> Switch Port 26 down (Enabled)
 Link state: Down (Polling) speed: 4xSDR MTU: 4096 VLs: 0-7
 Peer: ""  Peerport: 0
  NodeGUID: 0x0000000000000000 LID: 0x0000
```

```
Connector  4A Not present
Connector  5A Present <-> Switch Port 30 up (Enabled)
 Link state: Active (LinkUp) speed: 4xQDR MTU: 2048 VLs: 0-7
 Peer: "o4test56 HCA-1"  Peerport: 1
  NodeGUID: 0x0021280001cf205a LID: 0x0008
Connector  6A Not present
Connector  7A Present <-> Switch Port 33 up (Enabled)
 Link state: Active (LinkUp) speed: 4xQDR MTU: 2048 VLs: 0-7
 Peer: "o4test57 HCA-1"  Peerport: 1
  NodeGUID: 0x0021280001cf3786 LID: 0x0014
Connector  8A Not present
Connector  9A Not present
Connector 10A Not present
Connector 11A Not present
Connector 12A Not present
Connector 13A Not present
Connector 14A Not present
Connector 15A Not present
Connector 0A-ETH Present
 Bridge-0 Port 0A-ETH-1 (Bridge-0-2) up (Enabled)
 Bridge-0 Port 0A-ETH-2 (Bridge-0-2) up (Enabled)
 Bridge-0 Port 0A-ETH-3 (Bridge-0-1) up (Enabled)
 Bridge-0 Port 0A-ETH-4 (Bridge-0-1) up (Enabled)
Connector 1A-ETH Present
 Bridge-1 Port 1A-ETH-1 (Bridge-1-2) up (Enabled)
 Bridge-1 Port 1A-ETH-2 (Bridge-1-2) up (Enabled)
 Bridge-1 Port 1A-ETH-3 (Bridge-1-1) up (Enabled)
 Bridge-1 Port 1A-ETH-4 (Bridge-1-1) up (Enabled)
Connector  0B Not present
Connector  1B Not present
Connector  2B Not present
Connector  3B Not present
Connector  4B Not present
Connector  5B Not present
Connector  6B Not present
Connector  7B Not present
Connector  8B Not present
Connector  9B Not present
Connector 10B Not present
Connector 11B Not present
Connector 12B Not present
Connector 13B Not present
Connector 14B Not present
Connector 15B Not present
Link Switch Port 1 <-> Bridge-1 Port Bridge-1-2 up
 Link state: Active (LinkUp) speed: 4xQDR MTU: 4096 VLs: 0-1
 Peer: "SUN IB QDR GW switch o4nm2-gw-6 10.172.144.77 Bridge 1"  Peerport: 2
  NodeGUID: 0x00212856d102c040 LID: 0x0005
Link Switch Port 2 <-> Bridge-1 Port Bridge-1-1 up
```

```
 Link state: Active (LinkUp) speed: 4xQDR MTU: 4096 VLs: 0-1
 Peer: "SUN IB QDR GW switch o4nm2-gw-6 10.172.144.77 Bridge 1"  Peerport: 1
  NodeGUID: 0x00212856d102c040 LID: 0x0004
Link Switch Port 3 <-> Bridge-0 Port Bridge-0-2 up
 Link state: Active (LinkUp) speed: 4xQDR MTU: 4096 VLs: 0-1
 Peer: "SUN IB QDR GW switch o4nm2-gw-6 10.172.144.77 Bridge 0"  Peerport: 2
  NodeGUID: 0x00212856d102c000 LID: 0x0003
Link Switch Port 4 <-> Bridge-0 Port Bridge-0-1 up
 Link state: Active (LinkUp) speed: 4xQDR MTU: 4096 VLs: 0-1
 Peer: "SUN IB QDR GW switch o4nm2-gw-6 10.172.144.77 Bridge 0"  Peerport: 1
  NodeGUID: 0x00212856d102c000 LID: 0x0002
FabMan@gateway_name->
```

Related Information

- "ibportstate Command" on page 153
- "Linux Shells for Hardware Commands" on page 4

localmkeypersistence Command

Sets persistence of the M_Key.

Syntax

```
localmkeypersistence enable|disable|show|help
```

Description

This hardware command sets the persistence of the M_Key used by the I4 switch chips to survive a reboot of the management controller. The command has four arguments:

- enable – After a reboot, the local M_Key is restored before enabling the I4 switch chip ports.
- disable – After a reboot, the previous local M_Key is lost and no local M_Key is set before the Subnet Manager becomes operational.
- show – Displays the local M_Key persistence status.
- help – Provides help.

The `localmkeypersistence` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to display the local M_Key persistence status with the `localmkeypersistence` command.

```
FabMan@gateway_name->localmkeypersistence show
Local M_Key persistence is enabled.
Local M_Key will be restored after reboot before switch ports are enabled.
FabMan@gateway_name->
```

Related Information

- *Gateway Administration*, enabling secret M_Key persistence
- "Linux Shells for Hardware Commands" on page 4

`matchtopology` Command

Compares topology file to InfiniBand fabric.

Syntax

`matchtopology` [`-s` *systemname*] *topofile*|[`-h`]

where:

- *systemname* is the local system name in the topology file, should it be other than the host name.
- *topofile* is the filename and path to the topology file used for matching.

Description

This hardware command compares the topology file created with the `generatetopology` command with the current InfiniBand fabric topology. An error is displayed upon mismatch.

The matchtopology command is available from the /SYS/Fabric_Mgmt Linux shell target of the Oracle ILOM CLI interface.

Options

This table describes the options to the matchtopology command and their purposes.

Option	Purpose
-s	Increases compatibility with topology files not created by the generatetopology command. Use this option and specify the local system name used in the topology file.
-h	Provides help.

Example

This example shows how to compare the topology file to the current topology with the matchtopology command.

```
FabMan@gateway_name->matchtopology AUG22.topo
Topology matching will take some time. Please wait!
-I------------------------------------------------
-I- Topology matching results
-I------------------------------------------------
-I- The topology defined in topo1.topo perfectly matches the
discoverd fabric.
-------------------------------------------------------------
FabMan@gateway_name->
```

Related Information

-
-
-

`modifyvnic` Command

Modifies a VNIC.

Syntax

`modifyvnic` *connector*|*lagname* *vnic_ID* `[-mac` *mac*`][-vlan` *vlan_ID*`][-n][-h]`

where:

- *connector* is the name of the connector (`0A-ETH-1` to `0A-ETH-4` and `1A-ETH-1` to `1A-ETH-4`).
- *lagname* is the identifier of the LAG
- *vnic_ID* is the VNIC identifier.
- *mac* is the MAC address.

Note – Only even numbers are supported for the most significant byte of the MAC address (unicast).

Note – A VNIC must have a MAC address. If the VNIC is without a MAC address, assign one with the `-mac` option.

- *vlan_ID* is the VLAN identifier (`NO` or `2` to `4094`).

Note – Do not use VLAN identifiers `0`, `1`, or `4095`. According to the IEEE 802.1Q specification, VLAN ID 0 is used for priority tag, and VLAN ID 1 is usually reserved for a switch or bridge management VLAN.

Description

This hardware command enables you to modify a VNIC.

The `modifyvnic` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Options

The following table describes the options to the `modifyvnic` command and their purposes:

Option	Purpose
-mac	Specifies the MAC address.
-vlan	Specifies the VLAN identifier.
-n	Use for nonpersistent VNICs.
-h	Provides help.

Example

The following example shows how to modify the MAC address for VNIC 9 with the `modifyvnic` command.

```
# modifyvnic 0A-ETH-1 9 -mac 02:02:02:02:02:08
VNIC ID 9 modified
#
```

Related Information

- "createvnic Command" on page 27
- "deletevnic Command" on page 35
- "disablevnic Command" on page 44
- "enablevnic Command" on page 53
- "showvnics Command" on page 119
- "Linux Shells for Hardware Commands" on page 4

`setcontrolledhandover` Command

Enables or disables controlled handover.

Syntax

`setcontrolledhandover` *state*|`list`

where *state* is either `TRUE`, or `FALSE`.

Description

This hardware command enables or disables the controlled handover feature. The `TRUE` option enables and the `FALSE` option disables the action. The `list` option provides a listing of the Subnet Manager settings in respect to priority, controlled handover, and subnet prefix.

Note – You must stop or disable the Subnet Manager before using the `setcontrolledhandover` command. See "disablesm Command" on page 42.

The InfiniBand specification identifies two parameters used to negotiate a master Subnet Manager - the Subnet Manager priority and the port GUID. Should two or more Subnet Managers have the same highest configured priority, the Subnet Manager with the lowest GUID becomes the master Subnet Manager.

If an un-controlled or automatic handover state exists, when the primary-and-master Subnet Manager fails, the secondary-and-standby Subnet Manager takes on the role of the new master Subnet Manager. During this time, traffic is temporarily suspended as the secondary-and-master Subnet Manager discovers and sets up the InfiniBand fabric.

If the primary Subnet Manager is able to recover from the failure, reboot, and reinitialize itself, it identifies itself as the intended master Subnet Manager. Again, traffic is temporarily suspended as the primary-and-master Subnet Manager re-discovers and sets up the InfiniBand fabric. The secondary Subnet Manager returns to the standby status.

If controlled handover is enabled, then some priority values become reserved. The Subnet Managers dynamically adjust their respective priorities to avoid the dual handover situation. A secondary Subnet Manager that became a master Subnet Manager due to a handover raises its priority to a reserved value and retains its master Subnet Manager status. Regardless if the primary Subnet Manager comes back online, the second handover does not occur.

The user-configured priorities of the Subnet Managers are retained, only during a handover are reserved priorities used.

The result of this scheme is that the user-configured priority is respected during system boot, but dual handover does not occur because the primary Subnet Manager loses its master Subnet Manager status (secondary Subnet Manager priority changes to reserve).

Note – Manually starting Subnet Managers individually is equivalent to the system boot sequence.

The `setcontrolledhandover` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example – shows how to enable a controlled handover of Subnet Managers with the `setcontrolledhandover` command.

```
FabMan@gateway_name->setcontrolledhandover TRUE
Current SM settings:
smpriority 5
controlled_handover TRUE
subnet_prefix 0xabbababe
M_Key None
FabMan@gateway_name->
```

Related Information

- "enablesm Command" on page 50
- "setsmpriority Command" on page 91
- "setsubnetprefix Command" on page 94
- "Linux Shells for Hardware Commands" on page 4

setdefaultgwdiscpkey Command

Sets the default pkey for the BridgeX Manager.

Syntax

```
setdefaultgwdiscpkey p_key
```

where *p_key* is the partition key (`1` to `7fff` or `default`).

Description

This hardware command changes the default PKey used by the BridgeX Manager to control traffic. By default, hosts are configured with the PKey `default`. If a different discovery PKey value is configured with the `setdefaultgwdiscpkey` command for the BridgeX Manager, hosts using the gateway resources must be configured with identical discovery PKeys, otherwise the VNICs will not become active. This condition requires the `/etc/modprobe.conf` file on a Linux host to be edited, where the options for the `mlx4_vnic` driver is to use discovery Pkeys. The entry into the `/etc/modprobe.conf` file would have this format:

```
options mlx4_vnic discovery_pkeys=comma_delimited_list_of_pkeys
```

For example:

```
options mlx4_vnic discovery_pkeys=0xffff,0xfff1,0x3
```

Note – If you have used the `setdefaultgwdiscpkey` command before creating a LAG, after creating the LAG, you must use the `setdefaultgwdiscpkey` command again to force the LAG to use the PKey.

The `setdefaultgwdiscpkey` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to set the default gateway discovery PKey with the `setdefaultgwdiscpkey` command.

```
FabMan@gateway_name->setdefaultgwdiscpkey 7ff0
Stopping Bridge Manager..                                    [  OK  ]
Starting Bridge Manager.                                     [  OK  ]
FabMan@gateway_name->
```

setgwethport Command

Configures an Ethernet port.

Syntax

setgwethport *connector* [-mtu *mtu*][-txpause *pause*][-rxpause *pause*][-clear][-h]

where:

- *connector* is the name of the connector (0A-ETH or 1A-ETH).
- *mtu* is the value of the MTU.
- *pause* is the priority flow control pause in quanta (0 to 255, Global, or None)

Description

This hardware command manually configures an Ethernet port of the gateway. The Ethernet ports are set to 10Gb/s by default. Should a slower Ethernet device be attached to the connector, there is no auto-negotiation. Using the setgwethport command without options returns the current Ethernet status of the connector.

The setgwethport command is available from the /SYS/Gateway_Mgmt and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the setgwethport command and their purposes.

Option	Purpose
-mtu	Specifies the MTU.

Option	Purpose
-txpause	Specifies the transmit pause.
-rxpause	Specifies the receive pause.
--clear	Specifies to clear configured values and revert to default values. • linkmode: XFI • MTU: 9600 • pause: Global
-h	Provides help.

Example

This example shows how to set the MTU of connector 1A-ETH to 9600 using the setgwethport command.

```
FabMan@gateway_name->setgwethport 1A-ETH -mtu 9600
Port status for connector 1A-ETH-1:
Adminstate......................Enabled
State...........................Up
Link state......................Up
Protocol........................Ethernet
Link Mode.......................XFI
Speed...........................10Gb/s
MTU.............................9600
Tx pause........................Global
Rx pause........................Global
Port status for connector 1A-ETH-2:
.
.
.
Port status for connector 1A-ETH-4:
Adminstate......................Enabled
State...........................Up
Link state......................Up
Protocol........................Ethernet
Link Mode.......................XFI
Speed...........................10Gb/s
MTU.............................9600
Tx pause........................Global
Rx pause........................Global
FabMan@gateway_name->
```

Related Information

-
-

_____

`setgwinstance` Command

Sets the gateway instance.

Syntax

`setgwinstance` *instance*|`list`

where *instance* is a number from `0` to `63`.

Description

When you configure multiple gateways, the gateways identify themselves to each other with a gateway instance number. No two gateways can have the same instance number.

_____

Note – The `setgwinstance` command automatically stops and restarts the BridgeX manager.

_____

Using the `list` option displays known gateways and their instance numbers.

The `setgwinstance` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to set the gateway instance of the gateway to 3 using the setgwinstance command.

```
FabMan@gateway_name->setgwinstance 3
Stopping Bridge Manager..                                    [  OK  ]
Starting Bridge Manager.                                     [  OK  ]
FabMan@gateway_name->
```

This example shows how to display the gateway instances of known gateways with the list option of the setgwinstance command.

```
FabMan@gateway_name->setgwinstance list
Current GW instance: 3
FabMan@gateway_name->
```

Related Information

■ "Linux Shells for Hardware Commands" on page 4

setgwsl Command

Sets the service level.

Syntax

setgwsl eoib|ctrl *level*

where *level* is the service level from 0 to 15.

Description

This hardware command sets the service level of the Ethernet over InfiniBand (EoIB) data traffic (eoib) or control traffic (ctrl). Use service level 1 only for low-latency, high-priority, small-message, low-bandwidth traffic. Use other service levels for regular, high-bandwidth traffic. The showgwconfig command displays the current service levels.

The `setgwsl` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to set the EoIB data traffic service level to 2 using the `setgwsl` command.

```
FabMan@gateway_name->setgwsl eoib 2
Stopping Bridge Manager..                                    [  OK  ]
Starting Bridge Manager.                                     [  OK  ]
FabMan@gateway_name->
```

Related Information

- ■ "showgwconfig Command" on page 100
- ■ "Linux Shells for Hardware Commands" on page 4

`setgwsystemname` Command

Sets the system name.

Syntax

setgwsystemname *systemname*|--list|--clear [-h]

where *systemname* is the BridgeX manager's system name.

Description

This hardware command displays or sets the system name as recognized by the BridgeX manager. By default, the host name is the system name. The system name cannot be more than 19 characters long, otherwise it is truncated to 19 characters.

The `setgwsystemname` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the setgwsystemname command and their purposes.

Option	Purpose
--clear	Clears the configured string and reverts back to the host name.
--list	Displays the current string.
-h	Provides help.

Note – If you want to use a double-dash (--) in the system name, you must delimit the string with escape characters. See the example.

Example

This example shows how to set the system name recognized by the BridgeX manager to --myname using the setgwsystemname command.

```
FabMan@gateway_name->setgwsystemname "\"--myname\""
Stopping Bridge Manager..                                    [  OK  ]
Starting Bridge Manager.                                     [  OK  ]
FabMan@gateway_name->setgwsystemname --list
BXM system name set to --myname
FabMan@gateway_name->
```

Related Information

- "showgwconfig Command" on page 100
- "Linux Shells for Hardware Commands" on page 4

sethostvniclimit Command

Limits VNICs created in Host Manual Mode.

Syntax

sethostvniclimit *connector*|*lagname* <-guid *guid*>|<-host *hostname* -port *port*>|<-node *string* -port *port*> -limit *number* -vlanlist *vlan_ID1* ... *vlan_IDn*

where:

- *connector* is the name of the connector (0A-ETH-1 to 0A-ETH-4 and 1A-ETH-1 to 1A-ETH-4).
- *lagname* is the identifier of the LAG
- *guid* is the global unique identifier of the target port on the host associated with the VNICs.
- *hostname* is the host name of the management controller or system name recognized by the BridgeX manager.
- *port* is the port of the host.
- *string* is the character string in the node description field.
- *number* is the maximum number of host-created VNICs for the IOA.
- *vlan_ID* is the VLAN identifier (NO or 2 to 4094).

Note – Do not use VLAN identifiers 0, 1, or 4095. According to the IEEE 802.1Q specification, VLAN ID 0 is used for priority tag, and VLAN ID 1 is usually reserved for a switch or bridge management VLAN.

Description

This hardware command sets the limit for the maximum number of host-created VNICs assigned to a host port for a gateway connector. The sethostvniclimit command is preferable to the allowhostconfig command because it permits the gateway to control VNIC resources. Similar to the allowhostconfig command, the sethostvniclimit command enables Host Manual Mode for a specified host.

Note – The sethostvniclimit and allowhostconfig commands cannot be used simultaneously. You can display which is available for use with the showgwconfig command. If in the output of the showgwconfig command the Running Value for the Allow host VNIC config parameter is yes, then the allowhostconfig command is in use. If you want to use the sethostvniclimit command, use the disallowhostconfig command to change the Running Value to no (as seen with the showgwconfig command), and then use the sethostvniclimit command

---

The `sethostvniclimit` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the `sethostvniclimit` command and their purposes.

Option	Purpose
`-guid`	Specifies the GUID of the target port on the host associated with the VNIC.
`-host`	Specifies the host name or system name.
`-port`	Specifies the port of the host.
`-node`	Specifies the node description to identify the respective host.
`-limit`	Specifies the maximum VNICs created on the host.
`-vlanlist`	Specifies a list of VLAN identifiers.

Example

This example shows how to limit the number of VNICs to 5 for connector 0A-ETH-1 with the `sethostvniclimit` command.

```
FabMan@gateway_name->sethostvniclimit 0a-eth-1 -guid 0002C90300032D69 -limit 4
-vlanlist 3
FabMan@gateway_name->
```

Related Information

- *Gateway Administration*, create VNICs for Host Manual Mode
- "allowhostconfig Command" on page 9
- "Linux Shells for Hardware Commands" on page 4

`setmsmlocationmonitor` Command

Sets monitoring of the Subnet Manager.

Syntax

```
setmsmlocationmonitor state|list [-h]
```

where *state* is either `enable` or `disable`.

Description

This hardware command sets the state of monitoring the location and condition of the primary (or Master) Subnet Manager for the InfiniBand fabric. You can set the state of monitoring to either `enabled` or `disabled`. When enabled, information about the MSM is displayed in the user interface.

Note – If the monitor is located on a spine switch and the MSM is on another spine switch, the MSM might not be reachable through LID routed path. This situation gives the false impression in the user interface that there is no MSM in the fabric. Under these conditions, disable MSM monitoring

---

The `list` option displays the current state of monitoring.

The `setmsmlocationmonitor` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to display the state of the Subnet Manager monitor with the `setmsmlocationmonitor` command.

```
FabMan@gateway_name->setmsmlocationmonitor list
Current state of Master Subnet Manager monitioring: enable
FabMan@gateway_name->
```

Related Information

-
-

`setsmmkey` Command

Sets the readable M_Key for the Subnet Manager.

Syntax

`setsmmkey` *m_key*|`none`|`list`

where *m_key* is a 12-digit hexadecimal number

Description

This hardware command adds an extra layer of security to administering the Subnet Managers of the InfiniBand fabric. The Management Key (M_Key) is a 48-bit value that is used like a password to authorize access to the Subnet Manager Agent, thereby regulating control of the HCAs, switches, and other InfiniBand devices within the fabric. Any attempt to initialize or configure an InfiniBand node without the correct readable M_Key will fail.

Note – For *secret* M_Key functionality, see "smsubnetprotection Command" on page 128.

The readable M_Key provides for two specific security features:

- Any unauthorized host-based Subnet Manager cannot assume control of the InfiniBand fabric.
- Prevents unauthorized modification of the Subnet Manager Agent from outside of the Subnet Manager.

The `none` and `list` arguments of the `setsmmkey` command disable the readable M_Key functionality or display the current readable M_Key value respectively.

Note – The same readable M_Key value must be configured on all Subnet Managers in the InfiniBand fabric. Additionally, you must disable the Subnet Manager before setting the readable M_Key value, and re-enable the Subnet Manager after setting the readable M_Key value.

Note – Once the readable M_Key is configured, the `ibportstate` command can no longer enable, disable, or reset a port. However, the `enableswitchport` and `disableswitchport` commands are still functional.

Note – Enabling *secret* M_Key functionality overrides any readable M_Key configured.

The `setsmmkey` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to set the readable M_Key for the Subnet Manager with the `setsmmkey` command.

```
FabMan@gateway_name->setsmmkey 0xabbababe0001
Current SM settings:
smpriority 1
controlled_handover FALSE
subnet_prefix 0xfe80000000000000
M_Key 0xabbababe0001
Routing engine FatTree
FabMan@gateway_name->
```

Related Information

- "disablesm Command" on page 42
- "enablesm Command" on page 50
- "Linux Shells for Hardware Commands" on page 4

`setsmpriority` Command

Sets the Subnet Manager priority.

Syntax

```
setsmpriority priority|list
```

where *priority* is a number from `0` (lowest) to `13` (highest).

Description

This hardware command sets the priority of the Subnet Manager within the management controller. You use this command when there are multiple Subnet Managers in the InfiniBand fabric. By setting a Subnet Manager to a higher priority than the other Subnet Managers, it becomes the or master Subnet Manager. When you set a Subnet Manager to a priority lower than the master Subnet Manager, it becomes a standby Subnet Manager. The `list` option provides a listing of the Subnet Manager settings in respect to priority, controlled handover, and subnet prefix.

The `setsmpriority` command writes the value of *priority* to the `sm_priority` parameter of the configuration file.

Note – You must stop or disable the Subnet Manager before using the `setsmpriority` command. See "disablesm Command" on page 42.

Note – Setting Subnet Managers of the same fabric to the same priority can have undesirable results.

The `setsmpriority` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to set the priority of the Subnet Manager to 5 using the `setsmpriority` command.

```
FabMan@gateway_name->setsmpriority 5
Current SM settings:
smpriority 5
controlled_handover FALSE
subnet_prefix 0xfe80000000000000
M_Key 0xabbababe0001
Routing engine FatTree
FabMan@gateway_name->
```

This example shows how to display the current settings of the Subnet Manager priority, controlled handover, subnet prefix, M_Key value, and routing algorithm with the list argument of the setsmpriority command.

```
FabMan@gateway_name->setsmpriority list
smpriority 0
controlled_handover FALSE
subnet_prefix 0xfe80000000000000
M_Key 0xabbababe0001
Routing engine FatTree
FabMan@gateway_name->
```

Related Information

- "enablesm Command" on page 50
- "setcontrolledhandover Command" on page 77
- "setsubnetprefix Command" on page 94
- "Linux Shells for Hardware Commands" on page 4

setsmrouting Command

Sets the Subnet Manager routing algorithm.

Syntax

setsmrouting fattree|minhop|list

Description

This hardware command sets the routing algorithm of the Subnet Manager within the management controller. By default, the routing algorithm is Fat Tree, which is ideal for routing between spine switches. For routing between leaf switches, you might consider the Min Hop routing algorithm. The list option provides a listing of the Subnet Manager settings in respect to priority, controlled handover, subnet prefix, M_Key, and routing algorithm.

The setsmrouting command writes the value of *engine* to the routing_engine parameter of the configuration file.

Note – You must stop or disable the Subnet Manager before using the
`setsmrouting` command. See "disablesm Command" on page 42.

The `setsmpriority` command is available from the `/SYS/Fabric_Mgmt` Linux
shell target of the Oracle ILOM CLI interface.

Example

This example shows how to set the Min Hop routing algorithm with the
`setsmrouting` command.

```
FabMan@gateway_name->setsmrouting minhop
Current SM settings:
smpriority 0
controlled_handover FALSE
subnet_prefix 0xfe80000000000000
M_Key 0xabbababe0001
Routing engine minhop
FabMan@gateway_name->
```

Related Information

- *Gateway Administration*, setting the routing algorithm
- "setcontrolledhandover Command" on page 77
- "setsmpriority Command" on page 91
- "setsubnetprefix Command" on page 94
- "Linux Shells for Hardware Commands" on page 4

setsubnetprefix Command

Sets the subnet prefix.

Syntax

setsubnetprefix *prefix*|list

where *prefix* is the hexadecimal prefix.

Description

This hardware command sets the subnet prefix for the InfiniBand fabric. The prefix is entered in hexadecimal, starting with `0x` and having lower case alphanumeric characters. The `list` option provides a listing of the Subnet Manager settings in respect to priority, controlled handover, and subnet prefix.

The `setsubnetprefix` command writes the value of *prefix* to the `subnet_prefix` parameter of the configuration file.

Note – You must stop or disable the Subnet Manager before using the `setsubnetprefix` command. See "disablesm Command" on page 42.

The `setsubnetprefix` command is available from the `/SYS/Fabric_Mgmt` Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to set the subnet prefix to `0xabbababe` with the `setsubnetprefix` command.

```
FabMan@gateway_name->setsubnetprefix 0xabbababe
Current SM settings:
smpriority 5
controlled_handover TRUE
subnet_prefix 0xabbababe
M_Key None
Routing engine FatTree
FabMan@gateway_name->
```

Related Information

`showdisk` Command

Displays NAN flash drive information.

Syntax

```
showdisk
```

Description

This hardware command displays comprehensive status and life information about the NAN flash drive within the management controller. This drive is the repository of the firmware, operating system, and filesystem of the management controller.

The NAN flash technology is limited to a finite number of data erases to memory cells, in that each erase to the cell "ages" the cell. Once the age limit is exceeded, the memory cell is "dead" as it can no longer reliably store data. To extend the life of a NAN flash drive, when data is written, it is written to new locations throughout the group, and the file allocation table is accordingly updated.

For example, instead of erasing and writing data to the same memory cell over the course of five writes, the data is written to five different memory cells, one per each write. The file allocation table is updated on each write to point to the memory cell with the most current data. Though the previous memory cells still contain data, they are considered empty and are not erased until they are needed to receive different data writes in the future.

The output of the `showdisk` command includes:

- FRUID type information – Model, serial number, and firmware version.
- SMART capabilities – Both supported and enabled.
- Average age – The average number of erases to any given memory cell within a group.
- Number of block erases – The total number of erases to all memory cells in a group.
- Number of blocks – Similar to clusters in hard drives, the blocks are the smallest group of memory that can be erased.
- Estimated life used – A calculated percentage of memory cells that are dead or soon to be dead, based upon current erasing behaviors and a finite number of erases per memory cell.

- Estimated life remaining – 100% minus estimated life used.

The showdisk command is available from the /SYS/Switch_Diag, /SYS/Gateway_Mgmt, and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to information about and status of the NAN flash drive with the showdisk command.

```
FabMan@gateway_name->showdisk
 Device Model    :  512MB NANDrive
 Serial Number   :  0000000000A1PsB87FAa
 Firmware Version:  ND B642J

 SMART supported :  yes
 SMART enabled   :  yes

 Average age (4 groups):
  group 0:  12817
  group 1:    533
  group 2:    715
  group 3:    954

 Block erase count (4 groups):
  group 0:  12782181
  group 1:    532378
  group 2:    713173
  group 3:    952668
  Total   :  14980400

 Total physical blocks: 4096

 Estimated Life Usage    : 12.817 %

 Estimated Remaining Life: 87.183 %

FabMan@gateway_name->
```

Related Information

- *Gateway Administration*, displaying flash drive information
- "showfree Command" on page 98
- "Linux Shells for Hardware Commands" on page 4

`showfree` Command

Displays available RAM and filesystem space.

Syntax

```
showfree [-m][-d][-k]
```

Description

This hardware command displays the free RAM memory available for use by the management controller and the free disk space in the filesystem in megabytes. The available RAM is described as an absolute number of bytes and as a percentage of total. The available disk space is described as the filesystem, the mount point, the absolute number of bytes, and as a percentage of total.

The `showfree` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the `showfree` command and their purposes.

Option	Purpose
-m	Displays the free RAM memory.
-d	Displays the free filesystem disk space.
-k	Specifies the values in kilobytes.

Example

This example shows how to display the available RAM and disk space in kilobytes with the showfree command.

```
FabMan@gateway_name->showfree -mdk
Memory:                 306564 kB    60.1%

Filesystems:
 /dev/hda5    /          130168 kB   32.2%
 /dev/hda3    /var/log     8644 kB   55.6%
 /dev/hda2    /config     10763 kB   69.2%
 tmpfs        /dev/shm   255000 kB  100.0%
 tmpfs        /tmp       254004 kB   99.6%
FabMan@gateway_name->
```

Related Information

- *Gateway Administration*, displaying free memory
- "showdisk Command" on page 96
- "Linux Shells for Hardware Commands" on page 4

showfruinfo Command

Displays chassis FRU information.

Syntax

showfruinfo

Description

This hardware command displays the contents of the chassis FRU ID registers.

The showfruinfo command is available from the /SYS/Switch_Diag, /SYS/Gateway_Mgmt, and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display the chassis FRU ID registers with the showfruinfo command.

```
FabMan@gateway_name->showfruinfo
Sun_Man1R:
 UNIX_Timestamp32              :  Fri Dec 14 01:24:28 2012
 Sun_Fru_Description           :  ASSY,NM2-GW
 Vendor_ID_Code                :  11 E1
 Vendor_ID_Code_Source         :  01
 Vendor_Name_And_Site_Location :  4577 CELESTICA CORP. SAN JOSE CA US
 Sun_Part_Number               :  5111402
 Sun_Serial_Number             :  0110SJC-1010NG0040
 Serial_Number_Format          :  4V3F1-2Y2W2X4S
 Initial_HW_Dash_Level         :  03
 Initial_HW_Rev_Level          :  50
 Sun_Fru_Shortname             :  NM2 gateway
 Sun_Hazard_Class_Code         :  Y
 Sun_SpecPartNo                :  885-1655-01
Sun_FRU_LabelR:
 Sun_Serial_Number             :  1013AK208D
 FRU_Part_Dash_Number          :  541-4188-01
FabMan@gateway_name->
```

Related Information

showgwconfig Command

Displays gateway configuration.

Syntax

showgwconfig

Description

This hardware command displays status, version, and operational information for the gateway. The output includes a table of configuration parameters, and indicates if those parameters have been configured or are using default values. This table describes each of the columns of the output

Column Heading	Description
Parameter	The aspect or characteristic of the gateway configuration.
Configured Value	Values that have been configured by the user through CLI commands.
Running Value	Values currently being used. User-configured values take precedence.

The `showgwconfig` command is available from the `/SYS/Gateway_Mgmt` and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display status, version, and operational information for the gateway using the `showconfig` command.

```
FabMan@gateway_name->showgwconfig
BXM (pid 13724) is running
BXM versions: bxm_user 2.0.0898-0, BXM-API 1.6.0, bxm_libs 2.0.0898-0, bxm_main
1.31 mlx_bx_core 1.31
Parameter               Configured Value     Running Value
----------------------------------------------------------
GWInstance              42                   42
SystemName              None                 o4nm2-gw-4
EoIB Data SL            None                 1
EoIB Control SL         None                 2
Allow host VNIC config  None                 no
LAG mode                yes                  yes
Default discover P_key  None                 0xffff
System MAC              Not applicable       00:3b:aa:bb:a0:15
FabMan@gateway_name->
```

Related Information

- "disablesm Command" on page 42
- "enablesm Command" on page 50

■ "Linux Shells for Hardware Commands" on page 4

showgwports Command

Displays gateway port status.

Syntax

```
showgwports [-v][-h]
```

Description

This hardware command displays information about the gateway Ethernet ports. The information includes:

■ GUIDs
■ Node descriptions and names
■ Ports
■ LIDs
■ States
■ Modes
■ Speed
■ MTU

This table describes each of the columns of the output

Column Heading	Description
NodeGUID	GUID for the BridgeX CA node
NodeDescription	Node description test for the BridgeX CA node.
Device	BridgeX device identifier.
Port	Internal BridgeX port number.
Portname	A construct of the BridgeX device identifier and the internal BridgeX port number.

Column Heading	Description
PeerPort	The I4 switch chip port connected to the internal BridgeX port.
PortGUID	The GUID for the internal BridgeX port.
LID	The LID for the internal BridgeX port.
IBState	The combination of the InfiniBand port logical and physical state. Can be: • Sleep • Polling • Disabled • Training • Recovery • Init • Armed • Active
GWState	The gateway state for the port. Can be: • Reset • Start • Join Mcast • Join Fabric • Up • Error
Speed	Maximum available speed for the link.
VLs	Number of operational virtual lanes.
MTU	MTU.
Bridge	A construct of the BridgeX device identifier and the internal BridgeX port number.
Adminstate	State set by the enablegwport and disablegwport commands. Either enabled or disabled.
Link	Physical state of the link. Either Up or Down.
State	Logical state of the gateway port. Can be: • Start • Join Mcast • Join Fabric • Up • Error
TxPause	Transmit flow control. Can be: • None • 0-255 PFC • Global

Column Heading	Description
RxPause	Receive flow control. Can be: • None • 0-255 PFC • Global
Linkmode	Link mode of the Ethernet port.

The showgwports command is available from the /SYS/Gateway_Mgmt and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the showgwports command and their purposes.

Option	Purpose
-v	Displays verbose output.
-h	Provides help.

Example

This example shows how to display verbose information about the gateway Ethernet ports using the showgwports command.

Note – Your output will differ from that of the example.

```
FabMan@gateway_name->showgwports -v

BRIDGE DEVICES:
---------------
NodeGUID          NodeDescrition
--------------------------------
0x00212856d102c000 SUN IB QDR GW switch o4nm2-gw-6 10.172.144.77 Bridge 0
0x00212856d102c040 SUN IB QDR GW switch o4nm2-gw-6 10.172.144.77 Bridge 1

INTERNAL PORTS:
---------------
Device   Port Portname  PeerPort PortGUID           LID    IBState  GWState
Speed  VLs MTU
```

```
-------------------------------------------------------------------------------
-----------------
Bridge-0 1  Bridge-0-1   4   0x00212856d102c001 0x0002 Active  Up      40Gbs
2 4096
Bridge-0 2  Bridge-0-2   3   0x00212856d102c002 0x0003 Active  Up      40Gbs
2 4096
Bridge-1 1  Bridge-1-1   2   0x00212856d102c041 0x0004 Active  Up      40Gbs
2 4096
Bridge-1 2  Bridge-1-2   1   0x00212856d102c042 0x0005 Active  Up      40Gbs
2 4096

CONNECTOR 0A-ETH:
-----------------
Port     Bridge      Adminstate Link  State      MTU TxPause  RxPause
Linkmode     Speed
-------------------------------------------------------------------------------
------------------
0A-ETH-1 Bridge-0-2 Enabled  Up   Up      9600 Global  Global    XFI
10Gb/s
0A-ETH-2 Bridge-0-2 Enabled  Up   Up      9600 Global  Global    XFI
10Gb/s
0A-ETH-3 Bridge-0-1 Enabled  Up   Up      9600 Global  Global    XFI
10Gb/s
0A-ETH-4 Bridge-0-1 Enabled  Up   Up      9600 Global  Global    XFI
10Gb/s

CONNECTOR 1A-ETH:
-----------------
Port     Bridge      Adminstate Link  State      MTU  TxPause  RxPause
Linkmode     Speed
-------------------------------------------------------------------------------
------------------
1A-ETH-1 Bridge-1-2 Enabled  Up   Up      9600 Global  Global    XFI
10Gb/s
1A-ETH-2 Bridge-1-2 Enabled  Up   Up      9600 Global  Global    XFI
10Gb/s
1A-ETH-3 Bridge-1-1 Enabled  Up   Up      9600 Global  Global    XFI
10Gb/s
1A-ETH-4 Bridge-1-1 Enabled  Up   Up      9600 Global  Global    XFI
10Gb/s
FabMan@gateway_name->
```

Related Information

`showioadapters` Command

Displays information about IOAs.

Syntax

```
showioadapters [--withlimit][-h]
```

Description

This hardware command displays the I/O adapter GUID to LID to gateway connector mapping. An I/O adapter (IOA) represents a HCA port with an EoIB driver, that is able to reach the relevant gateway instance. The output of the command is in tabular form. This table describes each of the columns of the output.

Column Heading	Description
IOA_GUID	The I/O adapter's GUID, presented as eight colon-delimited hexadecimal pairs. Not to be confused with a MAC address.
NODE	Node description. Might also be a fully qualified hostname.
LID	The LID for the internal BridgeX port.
FLAGS	• HD – I/O adapter created by Host Manual Mode. • ND – I/O adapter created by Gateway Manual Mode.
GW	The connectors of the gateway.
MAX_HOST_VNICS	The limit of VNICs for the I/O adapter, created in Host Manual Mode.
VLANS	The VLANs assigned to the I/O adapter.

The showioadapters command is available from the /SYS/Gateway_Mgmt and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the showioadapters command and their purposes.

Option	Purpose
--withlimit	Displays I/O adapters imposed with a maximum number of host-created VNICs and the limit.
-h	Provides help.

Example

This example shows how to display information about the I/O adapters with the `showioadapters` command.

```
FabMan@gateway_name->showioadapters
IOA_GUID                   NODE         LID   FLAGS   GW
------------------------   ----------   ----  ------  -----
0021280001CF205B           o4test56     8     ND      1A-ETH-2
0021280001CF3787           o4test57     14    HD      1A-ETH-2
0021280001CF205B           o4test56     8     ND      1A-ETH-1
.
.
.
0021280001CF3787           o4test57     14    ND      0A-ETH-3
0021280001CF205B           o4test56     8     HD      0A-ETH-4
0021280001CF3787           o4test57     14    ND      0A-ETH-4
FabMan@gateway_name->
```

Related Information

- "addlagport Command" on page 8
- "createlag Command" on page 23
- "deletelag Command" on page 33
- "dellagport Command" on page 36
- "Linux Shells for Hardware Commands" on page 4

showlag Command

Displays information about configured LAGs.

Syntax

`showlag [`*`lagname`*`][-h]`

where *lagname* is the identifier of the LAG.

Description

This hardware command displays information about configured LAGs. If no *lagname* is specified, then the command displays general information about all configured LAGs. The output of the command is in tabular form. This table describes each of the columns of the output.

Column Heading	Description
`Name`	Can be either the name of the LAG or the names of the connectors in the LAG.
`Iport`	The internal port used by a LAG, typically displayed as `Bridge-`*`x`*`-`*`y`*.
`Members`	The number of connectors assigned to the LAG.
`Vnics`	The number of VNICs using the LAG.
`GW Port ID`	Dynamically assigned ID for the LAG.
`BANDWIDTH`	The bandwidth, either 10, 20, 30, ..., or 80 Gbps. 10 Gbps per port.
`Distribution`	The distribution method of bandwidth over the connectors.
`LACP Mode`	The manner in which ports interact with peers. • `active` – Ports always send frames to peers. • `passive` – Ports only answer to peers. • `off` – LACP is disabled.
`DiscoveryPkey`	Discovery P_Key used by the BridgeX manager.
`Type`	Type of LAG. • `Global` – Global Gateway with unlimited VNICs. • `1k` – Maximum of 1024 VNICs associated to external ports. • `4k` – Maximum of 4096 VNICs associated to external ports.
`SL`	Data service level, `0` to `15`.
`enabled`	Connector is enabled (`1`) or disabled (`0`).
`active`	Connector is active (`1`) or inactive (`0`).

Column Heading	Description
RC	Inactive reason code. Can be: • 0x0 – No issue. • 0x1 – The connector is disabled. • 0x2 – Physical link is down. • 0x3 – No connectivity between connectors of the LAG. • 0x4 – Connector does not see the Subnet Manager or there a problems with multicast joins. • 0x5 – LACP is disabled the connector because not all connectors could be aggregated into the LAG. • 0x6 – Speed mismatch of connectors or LAG itself. • 0x7 – Mismatch of receive pause configuration. • 0x8 – Mismatch of transmit pause configuration. • 0x9 – LACP is disabled on the connector. • 0xA – Internal port link is down. • 0xB – P_Key misconfiguration. P_Keys for LAG connectors do not match.

The -h option provides help.

The showlag command is available from the /SYS/Gateway_Mgmt and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Example

This example show how to display information about all configured LAGs with the showlag command.

```
FabMan@gateway_name->showlag
Name       Iport      Members Vnics Gw Port ID BANDWIDTH Distribution LACP  Mode
DiscoverPkey Type   SL
---------- ---------- ------- ----- ---------- --------- ------------ ----------
------------ ------ ----
bxmtest0   Bridge-0-1 0       1000  2704       NA        NA           NA
0xffff       Global 1
FabMan@gateway_name->
```

This example shows how to display information about the LAG named test with the showlag command.

```
FabMan@gateway_name->showlag test
Name       enabled active RC
---------- ------- ------ ---
```

```
0A-ETH-2    1        1       0x0
0A-ETH-3    1        1       0x0
FabMan@gateway_name->
```

Related Information

- ■ "addlagport Command" on page 8
- ■ "createlag Command" on page 23
- ■ "deletelag Command" on page 33
- ■ "dellagport Command" on page 36
- ■ "Linux Shells for Hardware Commands" on page 4

showpsufru Command

Displays power supply FRU ID information.

Syntax

showpsufru *slot*

where *slot* is the power supply slot (0 or 1).

Description

This hardware command displays FRU ID information for the power supplies.

The showpsufru command is available from the /SYS/Switch_Diag,
/SYS/Gateway_Mgmt, and /SYS/Fabric_Mgmt Linux shell targets of the Oracle
ILOM CLI interface.

Example

This example shows how to display the FRU ID information for power supply 0 with the showpsufru command.

```
FabMan@gateway_name->showpsufru 0
 Sun_SpecPartNo        :  885-1165-02
 UNIX_Timestamp32      :  Sun Jan  3 15:35:39 2010
 Sun_Part_Number       :  3002143
 Sun_Serial_No         :  BF15WA
 Vendor_ID_Code        :  02a2
 Initial_HW_Dash_Level :  02
 PSU_Voltage 1         :  0x04b0 (1200)
 PSU_Current 1         :  0x186a (6250)
 PSU_Voltage 2         :  0x014a (330)
 PSU_Current 2         :  0x012c (300)
 Sun_Hazard_Class_Code :  Y
 IPMI_Board_Manufacturer :  EMERSON
 IPMI_Board_Product_Name :  A237
 IPMI_Board_Serial_Number:  1357ZHO-0952BF15WA
 IPMI_Board_Part_Number  :  300-2143-02
FabMan@gateway_name->
```

Related Information

- "showfruinfo Command" on page 99
- "Linux Shells for Hardware Commands" on page 4

showsmlog Command

Displays the Subnet Manager log.

Syntax

showsmlog [-h]

Description

This hardware command first displays the parameters of the Subnet Manager configuration file, and then the chronological log entries for the Subnet Manager within the management controller. Tapping the space bar displays the next screen of the log. Pressing the Q key quits. The -h option provides help.

The showsmlog command is available from the /SYS/Switch_Diag, /SYS/Gateway_Mgmt, and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display the Subnet Manager log with the showsmlog command.

Note – The output of the example is a very small portion and will differ from your output.

```
FabMan@gateway_name->showsmlog
Jul 31 15:27:56 061000 [B7D328D0] 0x03 -> OpenSM 3.2.6_20120730 - Oracle patch
11.3
------------------------------------------------
OpenSM 3.2.6_20120730 - Oracle patch 11.3
 Reading Cached Option File: /etc/opensm/opensm.conf
 Loading Cached Option: sm_sl = 1
 Loading Cached Option: packet_life_time = 15
.
.
.
Sep 05 04:49:30 976000 [B765AB90] 0x02 -> osm_report_notice: Reporting
Informational Notice "CapabilityMask, NodeDescription, Link [Width|Spe
ed] Enabled changed" from LID 1, GUID 0x00212856d102c0a0, new CapabilityMask
0x4250084a
Sep 05 04:49:30 993000 [B6658B90] 0x02 -> Fabric has 1 switches - topology is
not fat-tree. Falling back to default routing
Sep 05 04:49:30 993000 [B6658B90] 0x01 -> ucast_mgr_route: ftree: cannot build
lid matrices.
(END)
FabMan@gateway_name->
```

Related Information
- "Linux Shells for Hardware Commands" on page 4

`showtemps` Command

Displays gateway temperatures.

Syntax

```
showtemps
```

Description

This hardware command displays internal temperatures for the gateway.

The `showtemps` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display gateway temperatures with the `showtemps` command.

```
FabMan@gateway_name->showtemps
Back temperature 29
Front temperature 29
SP temperature 47
Switch temperature 40, maxtemperature 42
Bridge-0 temperature 40, maxtemperature 41
Bridge-1 temperature 45, maxtemperature 46
All temperatures OK
FabMan@gateway_name->
```

Related Information

- "env_test Command" on page 54
- "Linux Shells for Hardware Commands" on page 4

`showtopology` Command

Displays the topology.

Syntax

```
showtopology [-h]
```

Description

This hardware command displays the InfiniBand fabric topology. The `showtopology` command displays the topology in a format different than the `ibnetdiscover` command.

The output of the `showtopology` command follows this basic format for each node:

```
device_type  device_name
panel_label  link_status-> connected_device_type  connected_device_name  connected_panel_label
and
panel_label  link_status-> connected_device_type  connected_device_name  connected_panel_label
.
.
for each subsequent port of that device_type
```

Where:

- *device_type* is the type of InfiniBand device. If the device is a Sun product, the *device_type* might be:
  - `SUNBQNEM48` – Sun Blade 6048 InfiniBand QDR Switched Network Express Module
  - `DEV26418_01P` – Single-port Sun Blade X6275 Server Module
  - `DEV26418_02P` – Dual-port Sun Blade X6275 Server Module
  - `SUNDCS36QDR` – Sun Datacenter InfiniBand Switch 36
  - `SUNDCS72QDR` – Sun Datacenter InfiniBand Switch 72
  - `SUNDCS648QDR` – Sun Datacenter InfiniBand Switch 648
  - `SUNIBQDRGW` – Sun Network QDR InfiniBand Gateway Switch
  - `I4_GENERIC` – An I4 switch ship

- *device_name* is the node description of the device, the hostname and number, or the modified GUID.
- *panel_label* is the name of the connector, hard wire link, or port as defined by the `ibnl` files. For example, `C-6A` is connector `6A`, `BX1` is BridgeX slice `1`, `A-SW` is switch chip `A`, and `P1` is port `1`.
- *link_status* is one of these:
  - `-1x-2.5G` – single data rate (SDR)
  - `-2x-5G` – dual data rate (DDR)
  - `-4x-10G` – quad data rate (QDR)
- *connected_device_type* is the device type at the other end of the link.
- *connected_device_name* is the name of the device at the other end of the link.
- *connected_panel_label* is the name of the connector or port at the other end of the link.

For example:

```
SUNIBQDRGW mnm34-96
  BX1/P1 -4x-10G-> SUNIBQDRGW mnm34-96 A-SW/P4
  BX3/P1 -4x-10G-> SUNIBQDRGW mnm34-96 A-SW/P2
  A-SW/P2 -4x-10G-> SUNIBQDRGW mnm34-96 BX3/P1
  A-SW/P4 -4x-10G-> SUNIBQDRGW mnm34-96 BX1/P1
```

The `showtopology` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display the InfiniBand fabric topology with the `showtopology` command.

Note – The output for your InfiniBand fabric will differ from that in the example.

```
FabMan@gateway_name->showtopology
SUNIBQDRGW mnm34-98 Bridge0
  BX1/P1 -4x-10G-> I4_GENERIC mnm34-98 P4
      -4x-10G-> I4_GENERIC mnm34-98 P3
  BX3/P1 -4x-10G-> I4_GENERIC mnm34-98 P2
      -4x-10G-> I4_GENERIC mnm34-98 P1
```

```
DEV26428_02P mnm34-55_HCA-1
  P1 -4x-10G-> SUNIBQDRGW mnm34-97 C-7A
  P2 -4x-10G-> I4_GENERIC mnm34-98 P33

SUNIBQDRGW mnm34-97
  A-SW/P2 -4x-10G-> SUNIBQDRGW mnm34-97 BX3/P1
  A-SW/P4 -4x-10G-> SUNIBQDRGW mnm34-97 BX1/P1
  C-0B -4x-10G-> I4_GENERIC mnm34-98 P21
  C-0A -4x-10G-> I4_GENERIC mnm34-98 P20
  C-1B -4x-10G-> I4_GENERIC mnm34-98 P19
  C-1A -4x-10G-> I4_GENERIC mnm34-98 P22
  C-2B -4x-10G-> I4_GENERIC mnm34-98 P24
  C-2A -4x-10G-> I4_GENERIC mnm34-98 P23
  C-8A -4x-10G-> DEV26428_02P mnm34-60_HCA-1 P1
  C-7A -4x-10G-> DEV26428_02P mnm34-55_HCA-1 P1
  C-6A -4x-10G-> DEV26428_02P mnm34-54_HCA-1 P1
  BX1/P1 -4x-10G-> SUNIBQDRGW mnm34-97 A-SW/P4
  BX3/P1 -4x-10G-> SUNIBQDRGW mnm34-97 A-SW/P2

I4_GENERIC mnm34-98
  P1 -4x-10G-> SUNIBQDRGW mnm34-98 Bridge 0
  P2 -4x-10G-> SUNIBQDRGW mnm34-98 Bridge 0 BX3/P1
  P3 -4x-10G-> SUNIBQDRGW mnm34-98 Bridge 0
  P4 -4x-10G-> SUNIBQDRGW mnm34-98 Bridge 0 BX1/P1
  P19 -4x-10G-> SUNIBQDRGW mnm34-97 C-1B
  P20 -4x-10G-> SUNIBQDRGW mnm34-97 C-0A
  P21 -4x-10G-> SUNIBQDRGW mnm34-97 C-0B
  P22 -4x-10G-> SUNIBQDRGW mnm34-97 C-1A
  P23 -4x-10G-> SUNIBQDRGW mnm34-97 C-2A
  P24 -4x-10G-> SUNIBQDRGW mnm34-97 C-2B
  P31 -4x-10G-> DEV26428_02P mnm34-60_HCA-1 P2
  P33 -4x-10G-> DEV26428_02P mnm34-55_HCA-1 P2
  P35 -4x-10G-> DEV26428_02P mnm34-54_HCA-1 P2

DEV26428_02P mnm34-54_HCA-1
  P1 -4x-10G-> SUNIBQDRGW mnm34-97 C-6A
  P2 -4x-10G-> I4_GENERIC mnm34-98 P35

DEV26428_02P mnm34-60_HCA-1
  P1 -4x-10G-> SUNIBQDRGW mnm34-97 C-8A
  P2 -4x-10G-> I4_GENERIC mnm34-98 P31

# Created from mnm34-98 at Thu Aug 19 13:32:25 PDT 2010
FabMan@gateway_name->
```

Related Information

- "generatetopology Command" on page 63

`showunhealthy` Command

Displays problematic components.

Syntax

```
showunhealthy
```

Description

This hardware command shows a list of gateway components that appear to have a problem. Unlike the `env_test` command, the `showunhealty` command *only* displays messages for components that have failed testing.

The `showunhealthy` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display a list of unhealthy components in the gateway with the `showunhealthy` command.

```
FabMan@gateway_name->showunhealthy
OK - No unhealthy sensors
FabMan@gateway_name->
```

Related Information

■ "env_test Command" on page 54

■ "Linux Shells for Hardware Commands" on page 4

`showvlan` Command

Displays VLANs.

Syntax

```
showvlan [-h]
```

Description

This hardware command displays the VLANs associated to the gateway connectors. The -h option provides help. This table describes each of the columns of the output

Column Heading	Description
Connector/LAG	LAG name or gateway connector (0A-ETH-1 to 0A-ETH-4 or 1A-ETH-1 to 1A-ETH-4).
VLN	VLAN identifier. A VLAN identifier of 0, -1, or NO indicates that there is no VLAN assigned.
PKEY	Partition key.

The showvlan command is available from the /SYS/Gateway_Mgmt and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display the VLANs associated with all connectors using the showvlan command.

Note – Your output will differ from that in the example.

```
FabMan@gateway_name->showvlan
  Connector/LAG  VLN  PKEY
  -------------  ---  ----
  0A-ETH-1        3   ffff
```

```
      0A-ETH-1          0    ffff
      0A-ETH-2          3    ffff
      0A-ETH-2          0    ffff
      0A-ETH-3          3    ffff
      0A-ETH-3          0    ffff
      0A-ETH-4          3    ffff
      0A-ETH-4          0    ffff
      1A-ETH-1          3    ffff
      1A-ETH-1          0    ffff
      1A-ETH-2          3    ffff
      1A-ETH-2          0    ffff
      1A-ETH-3          3    ffff
      1A-ETH-3          0    ffff
      1A-ETH-4          3    ffff
      1A-ETH-4          0    ffff
 FabMan@gateway_name->
```

Related Information

-
-
-

`showvnics` Command

Displays VNIC information.

Syntax

```
showvnics
```

Description

This hardware command displays VNICs and their parameters. The list of VNICs is in ascending order, with priority in this sequence:

- Connector number
- GUID

- State
- MAC address

This table describes each of the columns of the output

Column Heading	Description
ID	VNIC ID number. Roughly indicates creation sequence.
STATE	Logical state of the VNIC. Can be: • Up • Down • Active • Disabled
FLG	• H – VNIC created by Host Manual Mode. • N – VNIC created by Gateway Manual Mode. • A – VNIC created by Gateway Automatic Mode.
IOA_GUID	The IO adapter's GUID, presented as eight colon-delimited hexadecimal pairs. Not to be confused with a MAC address.
NODE	Node description. Might also be a fully qualified hostname.
IID	Identifier assigned to the VNIC by the host. Not to be confused for the ID (VNIC ID).
MAC	VNIC MAC address.
VLN	VLAN identifier. A VLAN identifier of 0, -1, or NO indicates that there is no VLAN assigned.
PKEY	Partition key.
GW	Gateway connector. 0A-ETH-1 to 0A-ETH-4 and 1A-ETH-1 to 1A-ETH-4.

The showvnics command is available from the /SYS/Gateway_Mgmt and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display the state of the VNICs with the showvnics command.

```
FabMan@gateway_name->showvnics
ID  STATE     FLG IOA_GUID                NODE       IID  MAC               VLN PKEY  GW
--- --------  --- ----------------------- ---------- ---- ----------------- ---
----     --------
 9 UP         N 00:03:BA:00:01:00:E3:71 mnm34-60 0000 02:02:02:02:02:07 NO  9002
0A-ETH-1
 8 UP         N 00:03:BA:00:01:00:E3:72 mnm34-60 0000 02:02:02:02:02:06 NO  9002
0A-ETH-1
 4 UP         N 00:03:BA:00:01:00:E3:71 mnm34-60 0000 02:02:02:02:02:03 NO  ffff
0A-ETH-2
 5 UP         N 00:03:BA:00:01:00:E3:71 mnm34-60 0002 02:02:02:02:02:04 NO  ffff
0A-ETH-2
 3 DISABLED   N 00:03:BA:00:01:00:E3:71 mnm34-60 0000 02:02:02:02:02:02 NO  ffff
0A-ETH-2
 6 UP         N 00:03:BA:00:01:00:E3:72 mnm34-60 0000 02:02:02:02:02:05 NO  ffff
0A-ETH-2
 7 UP         N 00:03:BA:00:01:00:E3:71 mnm34-60 0000 02:02:02:02:02:01 NO  ffff
0A-ETH-3
FabMan@gateway_name->
```

Related Information

- "createvnic Command" on page 27
- "deletevnic Command" on page 35
- "disablevnic Command" on page 44
- "enablevnic Command" on page 53
- "Linux Shells for Hardware Commands" on page 4

smconfigtest Command

Tests Subnet Manager configuration.

Syntax

smconfigtest [spine|leaf][-l][-h]

Description

This hardware command tests the Subnet Manager configuration and reports if it is corrupt.

The smconfigtest command is available from the /SYS/Fabric_Mgmt Linux shell target of the Oracle ILOM CLI interface.

Options

This table describes the options to the smconfigtest command and their purposes.

Option	Purpose
spine	Performs additional spine tests: • Routing engine is ftree. • Priority is 5 or 8. • Controlled handover is TRUE.
leaf	Performs additional leaf tests: • Routing engine is ftree. • Priority is 5. • Controlled handover is TRUE.
-l	Does not output to syslog.
-h	Provides help.

Example

This example shows how to test the Subnet Manager configuration (priority is 6) with the smconfigtest command.

```
FabMan@gateway_name->smconfigtest leaf
 smconfigtest:ERROR: /etc/opensm/opensm.conf file has illegal sm_priority : 6 ,
please change to 5
SubnetManager configuration is invalid for leaf switch.
FabMan@gateway_name->
```

Related Information

- ■ "getmaster Command" on page 65
- ■ "Linux Shells for Hardware Commands" on page 4

smnodes Command

Manages the Subnet Manager nodes file.

Syntax

smnodes add *IP_address* [*IP_address ...*]|delete *IP_address* [*IP_address ...*]|list

where *IP_address* is the IP address of the Subnet Manager nodes.

Description

This hardware command adds to, deletes from, or lists the Subnet Manager nodes of the Subnet Manager nodes file. The file is used when partitioning the InfiniBand fabric, to make partition managers aware of each other.

The smnodes command is available from the /SYS/Fabric_Mgmt Linux shell target of the Oracle ILOM CLI interface.

Example

This example shows how to list the Subnet Manager nodes of the file with the smnodes command.

```
FabMan@gateway_name->smnodes list
123.45.67.89
123.45.67.90
FabMan@gateway_name->
```

Related Information

- "smpartition Command" on page 124
- "Linux Shells for Hardware Commands" on page 4

`smpartition` Command

Manages the partition configuration.

Syntax

`smpartition` *subcommand* `[-h]`

This hardware command has subcommands that determine its functionality. This table describes the *subcommand*s and provides their syntax.

Subcommand Syntax	Description
`start`	Initiates a new configuration based upon a currently used configuration.
`create [-n` *partition_name*`] -pkey` *p_key* `[-flag ipoib [mtu` *mtu*`, rate` *rate*`, sl` *sl*`, scope` *scope*`]][-m` *defmember*`]`	Creates a new partition. The `-m` option configures the default membership for the partition.
`delete -n` *partition_name*`|-pkey` *p_key*	Deletes a partition.
`add -n` *partition_name*`|-pkey` *p_key* `-port` *port* `|ALL_CAS|ALL_SWITCHES|ALL_ROUTERS [-m` *member*`]`	Adds one or more ports to the partition. The `-m` option sets the membership for the port(s).
`remove -n` *partition_name*`|-pkey` *p_key* `-port` *port* `|ALL_CAS|ALL_SWITCHES|ALL_ROUTERS`	Removes one or more ports to the partition.
`modify -n` *partition_name*`|-pkey` *p_key*`[-flag [ipoib, mtu` *mtu*`, rate` *rate*`, sl` *sl*`, scope` *scope*`]]|[-port` *port* `|ALL_CAS|ALL_SWITCHES|ALL_ROUTERS [-m` *member*`]]`	Modifies a partition flag or port membership. The `-m` option sets the membership for the port(s).
`list active|modified [no-page]`	Displays the active or modified configuration. By default, the output is displayed one page at a time, advanced by pressing the spacebar. The `no-page` option enables a continuous stream of output without page breaks.
`commit`	Commits the modified configuration to become the active configuration.
`abort`	Abruptly ends the configuration session. All modified configuration information is lost and the active configuration remains unchanged.

where:

- *partition_name* is an alphanumeric tag to the InfiniBand partition (optional).

- *p_key* is the partition key (1 to 7fff or default).

Note – You cannot delete the pre-defined partitions with P_Keys 1 and 7fff.

- *mtu* is the number that maps to the actual MTU (1 to 5).

mtu **Number**	1	2	3	4	5
MTU Value	256	512	1024	2048	4096

- *rate* is the number that maps to the actual throughput of a link (link width + link speed) (2 to 10).

rate **Number**	2	3	4	5	6	7	8	9	10
Rate Value in Gbps	2.5	10	30	5	20	40	60	80	120

- *sl* is the service level (0 to 15).

Note – Use service level 1 (sl 1) only for low-latency, high-priority, small-message, low-bandwidth traffic. Use other service levels for regular, high-bandwidth traffic.

- *scope* is the multicast address scope value (1 to 14).

Note – The mtu, rate, sl, and scope parameters are for the multicast group created when ipoib (IP over InfiniBand) is configured for the partition. Typically, these values are not specified as the defaults are sufficient for the fabric configuration.

- *defmember* is the default membership type (full, limited, or both) for the partition.

Note – If ports are added to the partition without specifying the membership type, the default membership type is applied to the port

---

- *port* is the GUID of the port, or the special parameter, to add, remove, or modify:
  - ALL_CAS – All CAs in the InfiniBand fabric.
  - ALL_SWITCHES – All switches.

- ALL_ROUTERS – All routers.
- *member* is the membership type (full, limited, or both) for the port.

Description

This hardware command is used to manage the InfiniBand partition and is available only on management controllers that are hosting the primary (or Master) Subnet Manager. There are two configurations for the InfiniBand partition, the active configuration and the modified configuration. When configuring a partition, you must initiate the configuration session with the smpartition start command. During the session, you create a modified copy of the active configuration. To end the session, you must use the smpartition commit command to make the modified configuration the active configuration. Once committed, the active configuration is distributed to all Subnet Managers in the InfiniBand fabric where the management controller's IP addresses are listed in the Subnet Manager nodes file.

The Subnet Manager nodes file must exist in every management controller file system. The file contains a list of IP addresses of all active management controllers hosting a Subnet Manager in your fabric. The file should have an entry for every Sun Datacenter InfiniBand Switch 36 and Sun Network QDR InfiniBand Gateway Switch that runs a Subnet Manager in your InfiniBand fabric.

Note – If the Subnet Manager nodes of your InfiniBand fabric ever change (disabled, added, and so on), you must update all copies of the Subnet Manager nodes file and the fabric element configuration file. See "smnodes Command" on page 123 and "createfabric Command" on page 20.

Options

This table describes the options to the smpartition command and their purposes.

Option	Purpose
-n	Specifies the partition name.
-pkey	Specifies the partition key.

Option	Purpose
-flag	Specifies: • ipoib – If present, IP over InfiniBand is to be supported. • mtu – Sets the MTU. • rate – Sets the throughput of a link (link width + link speed). • sl – Sets the service level. • scope – Sets the multicast address scope.
-m	Specifies the membership type. If the -m option is used in the smpartition create command, the default membership type of the partition is specified. If the -m option is used with the smpartition add command or smpartition modify command, the membership type of the port is specified. If ports are added to the partition without specifying the membership type, the default membership type for the partition is applied to the port.
-port	Specifies the port or ports to be acted upon: • *port* – The GUID of the port to be acted upon. Alternatively, one these special parameters is specified instead of a GUID. • ALL_CAS – All CAs in the InfiniBand fabric. • ALL_SWITCHES – All switches. • ALL_ROUTERS – All routers.
-h	Provides help.

Example

This example shows how to display the active configuration of the InfiniBand partition with the smpartition command.

```
FabMan@gateway_name->smpartition list active
# Sun DCS IB partition config file
# This file is generated, do not edit
#! version_number : 16
Default=0x7fff, ipoib : ALL_CAS=full, ALL_SWITCHES=full, SELF=
full;
SUN_DCS=0x0001, ipoib : ALL_SWITCHES=full;
part1 = 0x9001,ipoib:
0x0002c90300089138=full,
0x0002c9030008923b=full,
0x0002c9030008923c=full,
0x0002c90300089103=limited,
0x0002c90300089104=full,
0x0002c90300089137=limited;
```

```
part2 = 0x9002,ipoib:
0x0003ba000100e389=full,
0x0002c903000890cb=limited,
0x0002c903000890cc=full,
0x0002c903000890c8=full,
0x0002c903000890c7=limited;
FabMan@gateway_name->
```

Related Information

- *Gateway Administration*, partitioning the InfiniBand fabric
- "setdefaultgwdiscpkey Command" on page 79
- "smnodes Command" on page 123
- "createfabric Command" on page 20
- "Linux Shells for Hardware Commands" on page 4

smsubnetprotection Command

Manages the secret M_Key.

Syntax

smsubnetprotection *subcommand* [-h]

This hardware command has subcommands that determine its functionality. This table describes the *subcommand*s and provides their syntax.

Subcommand Syntax	Description
`start [-force][-enable][-addonly|-deleteonly]`	Initiates a new configuration based upon a currently used configuration. Use the `-force` option to ignore the partition daemon check.
`list active|modified`	Displays a list of active secret M_Keys, the current secret M_Key, and the enabled status, or displays a list of pending M_Keys and the M_Key to be assigned to current status.
`listlocalmkey`	Displays the current local M_Key for an I4 switch chip without a corresponding Subnet Manager and its status.
`setlocalsecretmkey` *m_key*	Sets the secret M_Key locally for an I4 switch chip without a corresponding Subnet Manager.
`clearlocalmkey`	Clears the local secret M_Key.
`add` *m_key*	Adds an M_Key to the configuration.
`delete` *m_key*	Deletes an M_Key from the configuration.
`undo`	Reverts the previous `add`, `delete`, or `set-current` operation.
`set-current` *m_key*	Sets the current M_Key.
`commit [-force]`	Commits the modified configuration to become the active configuration. Use the `-force` option to ignore the partition daemon check.
`abort`	Abruptly ends the configuration session. All modified configuration information is lost, and the active configuration remains unchanged.
`setreplicationpassword` *password*	Configures the replication (and encryption) password.
`enablesecretmkey [-force]`	Enables secret M_Key functionality. Use the `-force` option to ignore the partition daemon check.
`disablesecretmkey [-force]`	Disables secret M_Key functionality. Use the `-force` option to ignore the partition daemon check.

where:

- *m_key* is the management key (16 hexadecimal digits).
- *password* is encryption string for M_Key replication (8 alphanumeric characters).

Description

This hardware command manages the secret M_Key and its implementation. The secret M_Key is a passphrase used by trusted Subnet Managers to securely perform activities (enabling ports, setting parameters, and so on) on the I4 switch chips of the InfiniBand fabric. The secret M_Key is an encrypted M_Key that is implemented at a lower level in fabric management than the standard M_Key. Use the smsubnetprotection command and its subcommands to create and manage the list of secret M_Keys.

When configuring a list of secret M_Keys, you initiate the configuration session on the master Subnet Manager with the smsubnetprotection start command. During the session, you add or delete secret M_Keys to the configuration, set the current secret M_Key, and list the M_Keys configured.

Note – There is a maximum of 10 secret M_Keys for the configuration.

To end the session, you must use the smsubnetprotection commit command to make the configuration active. Once committed, the configuration is automatically distributed to all Subnet Managers in the InfiniBand fabric.

Note – You cannot both add and delete secret M_Keys within a single configuration session. You must perform these actions in separate configuration sessions.

Should a local secret M_Key be created for an I4 switch chip without a corresponding Subnet Manager, that secret M_Key is only recognized by that I4 switch chip, and is unrecognized by the other I4 switch chips in the InfiniBand fabric.

Because of the complexity of the secret M_Key functionality, this table describes the impact of certain scenarios and actions you can take.

Scenario	Impact and Actions
Setting up secret M_Key in a mixed firmware fabric.	If the master Subnet Manager has firmware 2.1, only other Subnet Managers with firmware 2.1 can administrate the fabric. For Subnet Managers with firmware 2.0 or lower, the fabric "disappears".
	If the master Subnet Manager has firmware 2.0 or lower, you can only set up local secret M_Keys for the I4 switch chips on their respective Subnet Managers with firmware 2.1.
	Both situations are unsupported and not recommended.
Downgrading firmware after secret M_Key has been enabled.	If the master Subnet Manager is downgraded to firmware 2.0 or lower and there is a standby Subnet Manager with firmware 2.1, the secret M_Key is maintained through the standby Subnet Manager during the master Subnet Manager's reboot. After the reboot, the situation becomes as described above.
	If you downgrade any other Subnet Manager to firmware 2.0 or lower, the situation becomes as described above.
	Before you downgrade any firmware, disable secret M_Key.
	Note - Standard M_Key is not affected by a downgrade from firmware 2.1 to 2.0.
Upgrading from a lower firmware version.	Do not enable secret M_Key until all Subnet Managers in the fabric are at firmware version 2.1 or higher.
Introducing a new Subnet Manager with firmware 2.1 or higher, yet no secret M_Key policy, into a secret M_Key fabric.	If the new Subnet Manager has a low priority, it is updated with the new M_Keys and functionality automatically.
	If the new Subnet Manager has the highest priority, manually apply the secret M_Key policy and it will become the master Subnet Manager.
	Alternatively, preconfigure the new master Subnet Manager with a low priority. Allow it to be automatically updated. Then set the priority to master.
	Before introducing the new Subnet Manager:
	• Update the smnodes list with the smnodes command.
	• Update the fabric configuration with the fdconfig command.
	• Update the fabric mapping with the createfabric command.
M_Key values are mismatched.	If you add a Subnet Manager with one set of M_Keys to a fabric with a different set of M_Keys, the added Subnet Manager is not recognized.
	The situation is the same as above, however manually apply the correct secret M_Key policy using the setlocalsecretmkey subcommand.

Scenario	Impact and Actions
Merging two or more subnets into one fabric.	If each subnet is configured with different secret M_Key policies, then the subnets will not "see" each other and will act independently. If each subnet is configured with identical secret M_Key policies, there might be master Subnet Manager contention. Before physically merging: • Update the smnode files for all smnodes of both subnets with the smnodes command. • Configure both subnets with the new partition information with the smpartition command. • Configure both subnets with the new fabric configuration with the fdconfig command. • Correlate both subnets to the new fabric mapping with the createfabric command. You might need to use the -force option. • Configure identical secret M_Keys for both subnets. • Set the priority of one master Subnet Manager lower than the other.

This table describes each of the columns of the output of the smsubnetprotection command.

Column Heading	Description
Mkey	M_Keys that are assigned to Subnet Managers, but are not secure.
Untrusted Mkey	M_Keys that are not configured for all Subnet Managers, and hence untrusted.
Smkey	Secret M_Keys that are secure and trusted.
Attribute	The attribute of the M_Key:. • C – The current secret M_Key. • S – The standby secret M_Key about to become current.

The smsubnetprotection command is available from the /SYS/Fabric_Mgmt Linux shell target of the Oracle ILOM CLI interface.

Options

This table describes the options to the smsubnetprotection command and their purposes.

Option	Purpose
-force	Specifies the action to ignore the partition daemon check.
-enable	Specifies that after committing the configuration session, the secret M_Key functionality is automatically enabled.
-addonly	Specifies that the session is only to add secret M_Keys to the configuration.
-deleteonly	Specifies that the session is only to delete secret M_Keys from the configuration.

Example

This example shows how to display the active secret M_Keys with the smsubnetprotection command.

```
FabMan@gateway_name->smsubnetprotection list active
# File_format_version_number 1
# Sun DCS IB mkey config file
# This file is generated, do not edit
# secretmkey=enabled
# nodeid=o4nm2-gw-6
# time=15 Sep 03:54:46
# checksum=378d9b09744e1d8b8ba6ae868c99d0c9
#! commit_number : 3
Mkey                 Untrusted Mkey       Smkey                Attribute
------------------   ------------------   ------------------   ---------
0x00abcdefabcdef01    0x1aa45124fee612ae    0x15fc26aea300f831
0x00abcdefabcdef02    0x4ccd8230de6cd348    0x3fc7e6ad701a8a2a
0x00abcdefabcdef03    0x9baa1debcc74de5e    0x1b253003600d137b     C
FabMan@gateway_name->
```

Related Information

- *Gateway Administration*, securing the fabric
- "Linux Shells for Hardware Commands" on page 4

`version` Command (Hardware)

Displays gateway version.

Syntax

```
version
```

Description

This hardware command shows the hardware and software versions, and date information for the gateway and management controller.

The `version` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Example

This example shows how to display the version information with the `version` command.

```
FabMan@gateway_name->version
SUN DCS gw version: 2.0.5-1
Build time: Nov 25 2011 12:56:05
FPGA version: 0x33
SP board info:
Manufacturing Date: 2009.06.23
Serial Number: "NCD3R0527"
Hardware Revision: 0x0006
Firmware Revision: 0x0102
BIOS version: NOW1R112
BIOS date: 04/24/2009
FabMan@gateway_name->
```

Related Information

■ *Gateway Administration*, displaying firmware version

- "fwverify Command" on page 60
- "Linux Shells for Hardware Commands" on page 4

Understanding InfiniBand Commands

The InfiniBand commands act upon or monitor many aspects of the InfiniBand fabric. The commands are available through the Oracle ILOM CLI interface with the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` targets. See "Linux Shells for InfiniBand Commands" on page 138.

With the standard Linux shell CLI interface, only the `root` user of the management controller can run *all* of the InfiniBand commands. The format of the InfiniBand commands is typically as follows:

FabMan@*gateway_name*–>*command* [*option*] [*option*] ...

Command Syntax	Links
`ibdiagnet` [-c *count*][-v][-r][-o *outputdir*][-t *topology*][-s *system*][-i *device*][-p *port*][-wt *topology*][-pm][-pc][-P *PM =* value][-lw 1x|4x|12x][-ls 2.5|5|10][-skip *checks*][-load_db *file*][-h][-V]	"ibdiagnet Command" on page 139
`ibhosts` [-h][*topology*|-C *ca_name*][-P *ca_port*][-t *timeout*]	"ibhosts Command" on page 144
`ibnetdiscover` [-d][-e][-v][-s][-l][-g][-H][-S][-R][-C *ca_name*][-P *ca_port*][-t *timeout*][-V][-p][-h][*topology*]	"ibnetdiscover Command" on page 146
`ibnetstatus` [-h]	"ibnetstatus Command" on page 149
`ibnodes` [-h][*topology*|-C *ca_name*][-P *ca_port*][-t *timeout*]	"ibnodes Command" on page 151
`ibportstate` [-d(ebug)][-D(irect)][-e(rr_show)][-G(uid)][-h][-s *smlid*][-v(erbose)][-C *ca_name*][-P *ca_port*][-t(imeout) *timeout*][-M(key) *m_key*] *lid*|*dr_path*|*guid port* [*op*]	"ibportstate Command" on page 153
`ibroute` [-d][-a][-n][-D][-e][-G][-h][-M][-s *smlid*][-v][-V][-C *ca_name*][-P *ca_port*][-t *timeout*][*lid*|*dr_path*|*guid* [*startlid* [*endlid*]]]	"ibroute Command" on page 155
`ibrouters` [-h][*topology*|-C *ca_name*][-P *ca_port*][-t *timeout*]	"ibrouters Command" on page 158

Command Syntax	Links
`ibstat [-d][-e][-h][-l][-s][-p][-v][-V]` *ca_name* `[`*ca_port*`]`	"ibstat Command" on page 159
`ibswitches [-h][`*topology*`|-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]`	"ibswitches Command" on page 161
`ibtracert [-d][-D][-G][-h][-m` *mlid*`][-s` *smlid*`][-v][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][`*lid*`|`*dr_path*`|`*guid* `[`*startlid* `[`*endlid*`]]]`	"ibtracert Command" on page 162
`perfquery [-d][-e][-G][-h][-a][-l][-r][-R][-v][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][`*lid*`|`*guid* `[[`*port*`][`*reset_mask*`]]]`	"perfquery Command" on page 164
`saquery [-h][-d][-p][-N][-D][-S][-I][-L][-l][-G][-O][-U][-c]` `[-s][-g][-m][-x][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`][--src-to-dst` *source*`:`*destination*`][--sgid-to-dgid` *source-destination*`][`*name*`|`*lid*`|`*guid*`]`	"saquery Command" on page 167
`smpquery [-d][-D][-e][-G][-h][-v][-V][-C` *ca_name*`][-P` *ca_port*`][-t` *timeout*`]` *lid*`|`*dr_path*`|`*guid* `[`*op* *params*`]`	"smpquery Command" on page 170

Related Information

- "Understanding Hardware Commands" on page 1
- "Understanding SNMP MIB OIDs" on page 173

Linux Shells for InfiniBand Commands

The preferred method of accessing the Linux shell is through the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Using the `show` command on the `/SYS/Switch_Diag` target opens a restricted Linux shell that enables the `ilom-admin` user, `ilom-operator` user, and users with similar permissions to run diagnostic commands.

Using the `show` command on the `/SYS/Gateway_Mgmt` target opens a different restricted Linux shell that enables the `ilom-admin` user and users with similar permissions to run both diagnostic and gateway management commands.

Using the `show` command on the `/SYS/Fabric_Mgmt` target opens still an other restricted Linux shell that enables the `ilom-admin` user and users with similar permissions to run diagnostic, gateway, and fabric management commands.

Note – The `ilom-operator` user cannot access the Linux shell from either the /SYS/Gateway_Mgmt or /SYS/Fabric_Mgmt target.

This table lists the InfiniBand commands and their availability from the respective Linux shell targets. Typing the `help all` command from within the restricted shells lists the commands available to that shell.

Note – InfiniBand commands that are not listed are unavailable from the /SYS/Switch_Diag, /SYS/Gateway_Mgmt, or /SYS/Fabric_Mgmt targets.

Command	/SYS/Switch_Diag	/SYS/Gateway_Mgmt	/SYS/Fabric_Mgmt
ibdiagnet			Available
ibhosts	Available	Available	Available
ibnetstatus	Available	Available	Available
ibnodes	Available	Available	Available
ibportstate	Available	Available	Available
ibroute	Available	Available	Available
ibrouters	Available	Available	Available
ibstat	Available	Available	Available
ibswitches	Available	Available	Available
ibtracert	Available	Available	Available
perfquery	Available	Available	Available
saquery			Available
smpquery	Available	Available	Available

Related Information

- "Linux Shells for Hardware Commands" on page 4

`ibdiagnet` Command

Performs InfiniBand fabric diagnostic.

Syntax

```
ibdiagnet [-c count][-v][-r][-o outputdir][-t topology][-s system][-i
device][-p port][-wt topology][-pm][-pc][-P PM = value][-lw
1x|4x|12x][-ls 2.5|5|10][-skip checks][-load_db file][-h][-V]
```

where:

- *count* is the number of packets.
- *outputdir* is the output directory.
- *topology* is the topology file.
- *system* is the local system name.
- *device* is the index of the device connecting to the InfiniBand fabric.
- *port* is the port of the device.
- *PM* is the Performance Manager counter number.
- *value* is the threshold of the Performance Manager counter.
- *checks* is one or more strings that identify the checks made:
  - dup_guids
  - zero_guids
  - pm
  - logical_state
  - part
  - ipoib
  - all
- *file* is the subnet database .db file.

Description

This InfiniBand command scans the InfiniBand fabric using directed route packets,
extracting all the available information regarding the connectivity and devices. This
command produces a set of files in the output directory. By default, the output
directory is /tmp. This table describes the files.

File Name	Description
ibdiagnet.log	Dump of all the application reports generated according to the provided flags.
ibdiagnet.lst	List of all the nodes, ports, and links in the fabric.
ibdiagnet.fdbs	Dump of the unicast forwarding tables of the fabric switches.

File Name	Description
ibdiagnet.mcfdbs	Dump of the multicast forwarding tables of the fabric switches.
ibdiagnet.masks	In case of duplicate port/node GUIDs, this file includes the map between masked GUIDs and real GUIDs.
ibdiagnet.sm	List of all the Subnet Manager (state and priority) in the fabric.
ibdiagnet.pm	Dump of the Performance Manager counters values, for the fabric links.
ibdiagnet.pkey	Dump of the existing partitions and their member host ports.
ibdiagnet.mcg	Dump of the multicast groups, their properties, and member host ports.
ibdiagnet.db	Dump of the internal subnet database. You can load this file in later runs using the -load_db option.

During the discovery phase, the command also checks for duplicate node/port GUIDs in the InfiniBand fabric. If such an error is detected, it is displayed on the standard output.

After the discovery phase is completed, directed route packets are sent multiple times to detect possible problematic paths on which packets might be lost. A report of suspected bad links is displayed on the standard output.

If requested with the -r option, a full report of fabric qualities is displayed, including:

- Subnet Manager report
- Number of nodes and systems
- Hop-count information containing maximal hop-count, an example path, and a hop-count histogram
- All CA-to-CA paths traced
- Credit loop report
- MGID-MLID-HCAs multicast group and report
- Partitions report
- IPoIB report

Note – If the InfiniBand fabric includes only one CA, then CA-to-CA paths are not reported. Additionally, if a topology file is provided, the ibdiagnet command uses the names defined in the topology file for the output reports.

The ibdiagnet command is available from the /SYS/Fabric_Mgmt Linux shell target of the Oracle ILOM CLI interface.

Options

This table describes the options to the ibdiagnet command and their purposes.

Option	Purpose
-c	Sets the minimum number of packets sent across each link.
-v	Provides verbose output.
-r	Provides a report of fabric qualities.
-t	Specifies the topology file name.
-s	Specifies the local system name.
-i	In the case of multiple devices on the local system, this option specifies the index of the device of the port used to connect to the InfiniBand fabric.
-p	Specifies the local device port number used to connect to the InfiniBand fabric.
-o	Specifies the output directory.
-lw	Specifies the expected link width.
-ls	Specifies the expected link speed.
-pm	Dumps all the fabric link Performance Manager counters into ibdiagnet.pm.
-pc	Resets all the fabric link Performance Manager counters.
-P	Uses the Performance Manager counter of PM set to the threshold of value.
-skip	Skips the executions of the selected checks. One or more checks can be specified.
-wt	Writes out the discovered topology into the given file.
-load_db	Loads subnet data from the given .db file and skips the subnet discovery stage. **Note -** Some checks require actual subnet discovery and are disabled if load_db is specified. Those checks are for duplicate/zero GUIDs, link state, and Subnet Manager status.
-h	Provides help.
-V	Displays the version information.

Example

This example shows how to test the InfiniBand fabric with the `ibdiagnet` command. The command checks for 4x link width and 10 Gbit/sec speed, dumps the Performance Manager counters, and then clears the counters.

Note – The output for your InfiniBand fabric will differ from that in the example.

```
FabMan@gateway_name->ibdiagnet -lw 4x -ls 10 -pm -pc -skip dup_guids
Loading IBDIAGNET from: /usr/lib/ibdiagnet1.2
-W- Topology file is not specified.
  Reports regarding cluster links will use direct routes.
Loading IBDM from: /usr/lib/ibdm1.2
-I- Using port 0 as the local port.
-I- Discovering ... 2 nodes (1 Switches & 1 CA-s) discovered.
-I---------------------------------------------------
-I- Bad Guids/LIDs Info
-I---------------------------------------------------
-I- No bad Guids were found
-I---------------------------------------------------
-I- Links With Logical State = INIT
-I---------------------------------------------------
-I- No bad Links (with logical state = INIT) were found
-I---------------------------------------------------
-I- PM Counters Info
-I---------------------------------------------------
-I- No illegal PM counters values were found
-I---------------------------------------------------
-I- Links With links width != 4x (as set by -lw option)
-I---------------------------------------------------
-I- No unmatched Links (with width != 4x) were found
-I---------------------------------------------------
-I- Links With links speed != 10 (as set by -ls option)
-I---------------------------------------------------
-I- No unmatched Links (with speed != 10) were found
-I---------------------------------------------------
-I- Fabric Partitions Report (see ibdiagnet.pkey for a full hosts list)
-I---------------------------------------------------
-I---------------------------------------------------
-I- IPoIB Subnets Check
-I---------------------------------------------------
-I- Subnet: IPv4 PKey:0x7fff QKey:0x00000b1b MTU:2048Byte rate:10Gbps SL:0x00
-W- No members found for group
-I---------------------------------------------------
-I- Bad Links Info
-I- No bad link were found
```

```
-I----------------------------------------------------
----------------------------------------------------------------
-I- Stages Status Report:
  STAGE                                 Errors Warnings
  Bad GUIDs/LIDs Check                  0      0
  Link State Active Check               0      0
  Performance Counters Report           0      0
  Specific Link Width Check             0      0
  Specific Link Speed Check             0      0
  Partitions Check                      0      0
  IPoIB Subnets Check                   0      1
Please see /tmp/ibdiagnet.log for complete log
----------------------------------------------------------------
-I- Done. Run time was 2 seconds.
FabMan@gateway_name->
```

Related Information

- ibdiagnet man page
- "Linux Shells for InfiniBand Commands" on page 138

ibhosts Command

Displays host nodes.

Syntax

ibhosts [-h][*topology*|-C *ca_name*][-P *ca_port*][-t *timeout*]

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand command is a script that discovers the InfiniBand fabric topology or uses the existing topology file to extract the channel adapter nodes.

The ibhosts command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the ibhosts command and their purposes.

Option	Purpose
-h	Provides help.
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.

Example

This example shows how to display the host node GUIDs with the ibhosts command.

Note – The output in the example is a portion of the full output and varies for each InfiniBand topology.

```
FabMan@gateway_name->ibhosts
Ca      : 0x00212856cd22c000 ports 2 "SUN IB QDR GW switch mnm34-97 Bridge 0"
Ca      : 0x00212856cd22c040 ports 2 "SUN IB QDR GW switch mnm34-97 Bridge 1"
Ca      : 0x0002c903000891aa ports 2 "mnm34-54 HCA-1"
Ca      : 0x00212800013ece9e ports 2 "mnm34-55 HCA-1"
Ca      : 0x0003ba000100e370 ports 2 "mnm34-60 HCA-1"
.
.
.
FabMan@gateway_name->
```

Related Information

- `ibhosts` man page
- "ibnetdiscover Command" on page 146
- "ibnodes Command" on page 151
- "Linux Shells for InfiniBand Commands" on page 138

`ibnetdiscover` Command

Discovers the InfiniBand topology.

Syntax

`ibnetdiscover [-d][-e][-v][-s][-l][-g][-H][-S][-R][-C ca_name][-P ca_port][-t timeout][-V][--node-name-map map][-p][-h][topology]`

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *map* is the file name of the node name map.
- *topology* is the topology file.

Description

This InfiniBand command performs InfiniBand fabric discovery and outputs a human readable topology file. Nodes, node types, node descriptions, links, port numbers, port LIDs, and GUIDs are displayed. The output is directed to a topology file, if the file name is specified.

The output of the topology file follows this basic format for each node:

```
vendid=vendor_ID_in_hex
devid=device _ID_in_hex
and
sysimgguid=GUID_in_hex
and/or
```

```
switchguid=GUID_in_hex(portGUID_in_hex)
Switch ports_total "type-nodeGUID_in_hex" # "NodeDescription" base port 0 lid LID lmc 0
or
caguid=GUID_in_hex
Ca ports_total "type-nodeGUID_in_hex" # "NodeDescription"
and
[port] "type-nodeGUID_in_hex"[port](portGUID_in_hex) # "NodeDescription" lid LID widthspeed
[port] "type-nodeGUID_in_hex"[port](portGUID_in_hex) # "NodeDescription" lid LID widthspeed
.
.
.
```

For example:

```
vendid=0x2c9
devid=0xbd36
sysimgguid=0x21283a8389a0a3
switchguid=0x21283a8389a0a0(21283a8389a0a0)
Switch  36 "S-0021283a8389a0a0" # "Sun DCS 36 QDR switch localhost" enhanced
port 0 lid 15 lmc 0
[23]    "H-0003ba000100e388"[2](3ba000100e38a) # "mnm33-43 HCA-1" lid 14 4xQDR
.
.
.
```

Note – The ibnetdiscover command is available to only the root user.

Options

This table describes the options to the ibnetdiscover command and their purposes.

Option	Purpose
-d	Sets the debug level. Can be used several times to increase the debug level.
-e	Displays send and receive errors.
-v	Provides verbose output.
-s	Shows more information.
-l	Lists the connected nodes.

Option	Purpose
-g	Shows the grouping and switch external ports correspondence.
-H	Lists the connected channel adapters.
-S	Lists the connected switches.
-R	Lists the connected routers.
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.
-V	Displays the version information.
--node-name-map	Reads the node name map file.
-p	Returns a list of connected ports, including status information: • LID • portnum • GUID • link width • link speed • NodeDescription
-h	Provides help.

Example

This example shows how to discover the InfiniBand fabric topology with the ibnetdiscover command.

Note – The output for your InfiniBand fabric will differ from that in the example.

```
FabMan@gateway_name->ibnetdiscover
#
# Topology file: generated on Fri Dec 14 03:58:21 2012
#
# Max of 1 hops discovered
# Initiated from node 0021283a8389a0a0 port 0021283a8389a0a0
vendid=0x2c9
devid=0xbd36
sysimgguid=0x21283a8389a0a3
switchguid=0x21283a8389a0a0(21283a8389a0a0)
Switch  36 "S-0021283a8389a0a0" # "Sun DCS 36 QDR switch localhost" enhanced port
0 lid 15 lmc 0
```

```
[23]     "H-0003ba000100e388"[2](3ba000100e38a) # "mnm33-43 HCA-1" lid 14 4xQDR

vendid=0x2c9
devid=0x673c
sysimgguid=0x3ba000100e38b
caguid=0x3ba000100e388
Ca      2 "H-0003ba000100e388" # "mnm33-43 HCA-1"
[2](3ba000100e38a)      "S-0021283a8389a0a0"[23] # lid 14 lmc 0 "Sun DCS 36 QDR
switch localhost" lid 15 4xQDR
FabMan@gateway_name->
```

Related Information

■ `ibnetdiscover` man page

ibnetstatus Command

Displays status of the InfiniBand fabric.

Syntax

```
ibnetstatus [-h]
```

Description

This InfiniBand command provides a short status report of the InfiniBand fabric. It provides output equivalent to `ibdiagnet -ls 10 -lw 4x`.

The `ibnetstatus` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

The `-h` option provides help.

Example

This example shows how to display the status of the InfiniBand fabric with the `ibnetstatus` command.

Note – The output for your InfiniBand fabric will differ from that in the example.

```
FabMan@gateway_name->ibnetstatus
Loading IBDIAGNET from: /usr/lib/ibdiagnet1.2
-W- Topology file is not specified.
  Reports regarding cluster links will use direct routes.
Loading IBDM from: /usr/lib/ibdm1.2
-I- Using port 0 as the local port.
-I- Discovering ... 9 nodes (2 Switches & 7 CA-s) discovered.
-I----------------------------------------------------
-I- Bad Guids/LIDs Info
-I----------------------------------------------------
-I- skip option set. no report will be issued
-I----------------------------------------------------
-I- Links With Logical State = INIT
-I----------------------------------------------------
-I- No bad Links (with logical state = INIT) were found
-I----------------------------------------------------
-I- PM Counters Info
-I----------------------------------------------------
-W- lid=0x0010 guid=0x00212856cfe2c0a0 dev=48438 Port=1
 Performance Monitor counter      : Value
 symbol_error_counter             : 0xffff (overflow)
-W- lid=0x0010 guid=0x00212856cfe2c0a0 dev=48438 Port=2
 Performance Monitor counter      : Value
 symbol_error_counter             : 0xffff (overflow)
-W- lid=0x0010 guid=0x00212856cfe2c0a0 dev=48438 Port=3
 Performance Monitor counter      : Value
 symbol_error_counter             : 0xffff (overflow)
-W- lid=0x0010 guid=0x00212856cfe2c0a0 dev=48438 Port=4
 Performance Monitor counter      : Value
 symbol_error_counter             : 0xffff (overflow)
-I----------------------------------------------------
-I- Links With links width != 4x (as set by -lw option)
-I----------------------------------------------------
-I- No unmatched Links (with width != 4x) were found
-I----------------------------------------------------
-I- Links With links speed != 10 (as set by -ls option)
-I----------------------------------------------------
-I- No unmatched Links (with speed != 10) were found
-I----------------------------------------------------
```

```
-I- Fabric Partitions Report (see ibdiagnet.pkey for a full hosts list)
-I----------------------------------------------------
-I----------------------------------------------------
-I- IPoIB Subnets Check
-I----------------------------------------------------
-I- Subnet: IPv4 PKey:0x0001 QKey:0x00000b1b MTU:2048Byte rate:10Gbps SL:0x00
-W- No members found for group
-I- Subnet: IPv4 PKey:0x7fff QKey:0x00000b1b MTU:2048Byte rate:10Gbps SL:0x00
-W- No members found for group
-I----------------------------------------------------
-I- Bad Links Info
-I- No bad link were found
-I----------------------------------------------------
-I- Stages Status Report:
 STAGE                                Errors Warnings
 Bad GUIDs/LIDs Check                 0      0
 Link State Active Check              0      0
 Performance Counters Report          0      4
 Specific Link Width Check            0      0
 Specific Link Speed Check            0      0
 Partitions Check                     0      0
 IPoIB Subnets Check                  0      2
--------------------------------------------------------------
-I- Done. Run time was 22 seconds.
FabMan@gateway_name->
```

Related Information

- ■ `ibnetstatus` man page
- ■ "ibdiagnet Command" on page 139
- ■ "Linux Shells for InfiniBand Commands" on page 138

ibnodes Command

Displays InfiniBand nodes in topology.

Syntax

ibnodes [-h][*topology*|-C *ca_name*][-P *ca_port*][-t *timeout*]

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand command is a script that discovers the InfiniBand fabric topology or uses the existing topology file to extract the InfiniBand nodes of the channel adapters, switches, and routers.

The ibnodes command is available from the /SYS/Switch_Diag, /SYS/Gateway_Mgmt, and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the ibnodes command and their purposes.

Option	Purpose
-h	Provides help.
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.

Example

This example shows how to display the node GUIDs with the ibnodes command.

Note – The output for your InfiniBand fabric will differ from that in the example.

```
FabMan@gateway_name->ibnodes
Ca      : 0x0003ba000100e388 ports 2 "mnm33-43 HCA-1"
Switch : 0x0021283a8389a0a0 ports 36 "Sun DCS 36 QDR switch localhost" enhanced
port 0 lid 15 lmc 0
FabMan@gateway_name->
```

Related Information

- `ibnodes` man page

`ibportstate` Command

Manages the state and link speed of an InfiniBand port.

Syntax

```
ibportstate [-d(ebug)][-D(irect)][-e(rr_show)][-G(uid)][-h][-s
smlid][-v(erbose)][-C ca_name][-P ca_port][-t(imeout) timeout][-M(key)
m_key] lid|dr_path|guid port [op]
```

where:

- *smlid* is the Subnet Manager LID.
- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *m_key* is the management key (16 hexadecimal digits).
- *lid* is the local identifier.
- *dr_path* is the directed path.
- *guid* is the global unique identifier.
- *port* is the port being validated.
- *op* is the operation to perform on the port:
  - `enable`
  - `disable`
  - `reset`
  - `speed` *number* (where *number* is 1 for SDR, 2 for DDR, and 4 for QDR)
  - `query` (default)

Description

This InfiniBand command queries the logical and physical state of an InfiniBand port. The command can return the link width and speed of a switch chip port, as well as enabling, disabling, or resetting the port. The command can also set the link speed of any InfiniBand port.

Note – State changes made with the `ibportstate` command are not recognized by the `disableswitchport`, `enableswitchport`, or `listlinkup` commands.

Note – Speed changes are not affected until the port undergoes link renegotiation. Additionally, speed values are additive for enabling. For example, `speed 7` is 2.5, 5.0, and 10.0 Gbit/sec.

The `ibportstate` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the `ibportstate` command and their purposes.

Option	Purpose
-d	Sets the debug level. Can be used several times to increase the debug level.
-D	Uses the directed path address. The path is a comma-delimited sequence of out ports.
-e	Displays send and receive errors.
-G	Uses the port GUID address.
-h	Provides help.
-s	Uses *smlid* as the target LID for Subnet Manager or Subnet Administrator queries.
-v	Provides verbose output.
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.
-M	Specifies the M_Key to use.

Example

This example shows how to query the state and link speed of LID 15, port 23 with the `ibportstate` command.

```
FabMan@gateway_name->ibportstate 15 23
PortInfo:
# Port info: Lid 15 port 23
LinkState:......................Active
PhysLinkState:...................LinkUp
LinkWidthSupported:..............1X or 4X
LinkWidthEnabled:................1X or 4X
LinkWidthActive:.................4X
LinkSpeedSupported:..............2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkSpeedEnabled:................2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkSpeedActive:.................10.0 Gbps
Peer PortInfo:
# Port info: Lid 15 DR path slid 15; dlid 65535; 0,23
LinkState:......................Active
PhysLinkState:...................LinkUp
LinkWidthSupported:..............1X or 4X
LinkWidthEnabled:................1X or 4X
LinkWidthActive:.................4X
LinkSpeedSupported:..............2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkSpeedEnabled:................2.5 Gbps or 5.0 Gbps or 10.0 Gbps
LinkSpeedActive:.................10.0 Gbps
FabMan@gateway_name->
```

Related Information

- `ibportstate` man page
- "disableswitchport Command" on page 43
- "enablegwport Command" on page 47
- "enablesm Command" on page 50
- "listlinkup Command" on page 71
- "Linux Shells for InfiniBand Commands" on page 138

ibroute Command

Queries InfiniBand switch forwarding tables.

Syntax

```
ibroute [-d][-a][-n][-D][-e][-G][-h][-M][-s smlid][-v][-V][-C
ca_name][-P ca_port][-t timeout][lid|dr_path|guid [startlid [endlid]]]
```

where:

- *smlid* is the Subnet Manager LID.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *dr_path* is the directed path.
- *guid* is the global unique identifier.
- *startlid* is the starting local identifier.
- *endlid* is the ending local identifier.

Description

This InfiniBand command uses SMPs to display the forwarding tables for the specified switch LID and optionally, the LID range. By default, the range is all valid entries from 1 to FDBTop.

The ibroute command is available from the /SYS/Switch_Diag, /SYS/Gateway_Mgmt, and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the ibroute command and their purposes.

Option	Purpose
-a	Shows all LIDs in the range, including invalid entries.
-n	Does not try to resolve destinations.
-d	Sets the debug level. Can be used several times to increase the debug level.
-D	Uses the directed path address. The path is a comma-delimited sequence of out ports.
-e	Displays send and receive errors.

Option	Purpose
-G	Uses the port GUID address.
-h	Provides help.
-M	Shows multicast forwarding tables.
-s	Uses *smlid* as the target LID for Subnet Manager or Subnet Administrator queries.
-v	Provides verbose output.
-V	Displays the version information.
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.

Example

This example shows how to display the forwarding table for LID 15 with the
ibroute command.

Note – The output for your InfiniBand fabric will differ from that in the example.

```
FabMan@gateway_name->ibroute 15
Unicast lids [0x0-0xf] of switch Lid 15 guid 0x0021283a8389a0a0 (Sun DCS 36 QDR
switch localhost):
  Lid  Out   Destination
      Port    Info
0x000e 023 : (Channel Adapter portguid 0x0003ba000100e38a: 'mnm33-43 HCA-1')
0x000f 000 : (Switch portguid 0x0021283a8389a0a0: 'Sun DCS 36 QDR switch
localhost')
2 valid lids dumped
FabMan@gateway_name->
```

Related Information

- ibroute man page
- "ibtracert Command" on page 162
- "Linux Shells for InfiniBand Commands" on page 138

`ibrouters` Command

Displays InfiniBand router nodes in topology.

Syntax

`ibrouters [-h][`*topology*`|-C `*ca_name*`][-P `*ca_port*`][-t `*timeout*`]`

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand software command is a script that discovers the InfiniBand fabric topology or uses an existing topology file to extract the router nodes.

The `ibrouters` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the `ibrouters` command and their purposes.

Option	Purpose
-h	Provides help.
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.

Example

This example shows how to display router nodes with the `ibrouters` command.

Note – The output in the example is a portion of the full output and varies for each InfiniBand topology.

```
FabMan@gateway_name->ibrouters
FabMan@gateway_name->
```

Related Information

- `ibrouters` man page
- "ibnetdiscover Command" on page 146
- "ibnodes Command" on page 151
- "Linux Shells for InfiniBand Commands" on page 138

`ibstat` Command

Queries basic status of InfiniBand devices.

Syntax

`ibstat [-d][-e][-h][-l][-s][-p][-V]` *ca_name* [*ca_port*]

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.

Description

This InfiniBand software command displays basic information retrieved from the local InfiniBand driver. Output of the command includes:

- `LID`
- `SMLID`

- `port logical state`
- `link width`
- `port physical state`

The `ibstat` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the `ibstat` command and their purposes.

Option	Purpose
-d	Sets the debug level. Can be used several times to increase the debug level.
-e	Displays send and receive errors.
-h	Provides help.
-l	Lists all InfiniBand devices.
-s	Provides short output.
-p	Shows port list.
-V	Displays the version information.

Example

This example shows how to display the basic status from the local InfiniBand driver with the `ibstat` command.

```
FabMan@gateway_name->ibstat
Switch 'is4_0'
  Switch type: MT48436
  Number of ports: 0
  Firmware version: 7.3.0
  Hardware version: a0
  Node GUID: 0x00212856cfe2c0a0
  System image GUID: 0x00212856cfe2c0a3
  Port 0:
      State: Active
      Physical state: LinkUp
      Rate: 40
      Base lid: 16
```

```
      LMC: 0
      SM lid: 16
      Capability mask: 0x4250084a
      Port GUID: 0x00212856cfe2c0a0
 FabMan@gateway_name->
```

Related Information

- `ibstat` man page
- "Linux Shells for InfiniBand Commands" on page 138

`ibswitches` Command

Displays InfiniBand switch nodes in the topology.

Syntax

`ibswitches` [-h][*topology*|-C *ca_name*][-P *ca_port*][-t *timeout*]

where:

- *topology* is the topology file.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.

Description

This InfiniBand command is a script the discovers the InfiniBand fabric topology or uses an existing topology file to extract the switch nodes.

The `ibswitches` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the `ibswitches` command and their purposes.

Option	Purpose
-h	Provides help.
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.

Example

This example shows how to display the switch GUIDs with the `ibswitches` command.

Note – The output for your InfiniBand fabric will greatly differ from that in the example.

```
FabMan@gateway_name->ibswitches
Switch  : 0x00212856cfe2c0a0 ports 36 "SUN IB QDR GW switch mnm34-98" enhanced
port 0 lid 6 lmc 0
FabMan@gateway_name->
```

Related Information

- `ibswitches` man page
- "ibnetdiscover Command" on page 146
- "ibnodes Command" on page 151
- "Linux Shells for InfiniBand Commands" on page 138

`ibtracert` Command

Traces the InfiniBand path.

Syntax

```
ibtracert [-d][-D][-G][-h][-m mlid][-s smlid][-v][-V][-C
ca_name][-P ca_port][-t timeout][lid|dr_path|guid [startlid [endlid]]]
```

where:

- *mlid* is the multicast LID.
- *smlid* is the Subnet Manager LID.
- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *dr_path* is the directed path.
- *guid* is the global unique identifier.
- *startlid* is the starting LID for a range.
- *endlid* is the ending LID for a range.

Description

This InfiniBand command uses SMPs to trace the path from a source GID or LID to a destination GID or LID. Each responding hop in the path is displayed. The -m option enables multicast path tracing between source and destination nodes.

The ibtracert command is available from the /SYS/Switch_Diag, /SYS/Gateway_Mgmt, and /SYS/Fabric_Mgmt Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the ibtracert command and their purposes.

Option	Purpose
-d	Sets the debug level. Can be used several times to increase the debug level.
-D	Uses the directed path address. The path is a comma-delimited sequence of out ports.
-G	Uses the port GUID address.
-h	Provides help.

Option	Purpose
-m	Shows the multicast trace of the specified MLID.
-s	Uses *smlid* as the target LID for Subnet Manager or Subnet Administrator queries.
-v	Provides verbose output.
-V	Displays the version information.
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.

Example

This example shows how to display the path from LID 25 to LID 24 with the ibtracert command.

Note – The output for your InfiniBand fabric will differ from that in the example.

```
FabMan@gateway_name->ibtracert 15 14
From switch {0x00212856cd22c0a0} portnum 0 lid 15-15 "SUN IB QDR GW switch
mnm34-97"
[1] -> ca port {0x00212856cd22c042}[2] lid 14-14 "SUN IB QDR GW switch mnm34-97
Bridge 1"
To ca {0x00212856cd22c040} portnum 2 lid 14-14 "SUN IB QDR GW switch mnm34-97
Bridge 1"
FabMan@gateway_name->
```

Related Information

- ibtracert man page
- "ibroute Command" on page 155
- "Linux Shells for InfiniBand Commands" on page 138

perfquery Command

Queries InfiniBand port counters.

Syntax

```
perfquery [-d][-e][-G][-h][-a][-l][-r][-R][-v][-V][-C ca_name][-P
ca_port][-t timeout][[lid|guid [[port][reset_mask]]]
```

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *guid* is the global unique identifier.
- *port* is the port being queried.
- *reset_mask* is the two-byte mask.

Description

This InfiniBand command uses the performance management GMPs to acquire the `PortCounters` or `PortExtendedCounters` from the Performance Manager agent at the node or port specified.

Note – The data values retrieved from `PortCounters` and `PortExtendedCounters` are represented as octets divided by 4.

Note – Providing a *port* value of 255 ensures that the operation is performed on all ports.

The `perfquery` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the `perfquery` command and their purposes.

Option	Purpose
-d	Sets the debug level. Can be used several times to increase the debug level.

Option	Purpose
-e	Shows extended port counters.
-a	Shows aggregate counters for all ports.
-G	Uses the port GUID address.
-h	Provides help.
-l	Loops through all ports.
-r	Resets the counters after reading.
-R	Resets counters only.
-v	Provides verbose output.
-V	Displays the version information.
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.

Example

This example shows how to display all port counters for LID 15, port 23 with the
perfquery command.

```
FabMan@gateway_name->perfquery 15 23
# Port counters: Lid 15 port 23
PortSelect:.....................23
CounterSelect:..................0x1b01
SymbolErrors:...................0
LinkRecovers:...................0
LinkDowned:.....................0
RcvErrors:......................0
RcvRemotePhysErrors:............0
RcvSwRelayErrors:...............0
XmtDiscards:....................0
XmtConstraintErrors:............0
RcvConstraintErrors:............0
LinkIntegrityErrors:............0
ExcBufOverrunErrors:............0
VL15Dropped:....................0
XmtData:........................20232
RcvData:........................20232
XmtPkts:........................281
RcvPkts:........................281
FabMan@gateway_name->
```

Related Information

- `perfquery` man page
-

`saquery` Command

Queries InfiniBand fabric administration attributes.

Syntax

```
saquery [-h][-d][-p][-N][-D][-S][-I][-L][-l][-G][-O][-U][-c][-s]
[-g][-m][-x][-C ca_name][-P ca_port][-t timeout][--src-to-dst
source:destination][--sgid-to-dgid source-destination][name|lid|guid]
```

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *lid* is the local identifier.
- *guid* is the global unique identifier.
- *name* is the query name described in this table.

Query Names	Alias for *name*	Format
ClassPortInfo	CPI	
NodeRecord	NR	
PortInfoRecord	PIR	
SL2VLTableRecord	SL2VL	[[*lid*]/[*in_port*]/[*out_port*]]
PKeyTableRecord	PKTR	[[*lid*]/[*port*]/[*block*]]
VLArbitrationTableRecord	VLAR	[[*lid*]/[*port*]/[*block*]]
InformInfoRecord	IIR	
LinkRecord	LR	[[*from_lid*]/[*from_port*]] [[*to_lid*]/[*to_port*]]
ServiceRecord	SR	

Query Names	Alias for *name*	Format
PathRecord	PR	
MCMemberRecord	MCMR	
LFTRecord	LFTR	[[*lid*]/[*block*]]
MFTRecord	MFTR	[[*mlid*]/[*position*]/[*block*]]

Description

This InfiniBand command performs the selected Subnet Administrator query. Node records are queried by default.

The saquery command is available from the /SYS/Fabric_Mgmt Linux shell target of the Oracle ILOM CLI interface.

Options

This table describes the options to the saquery command and their purposes.

Option	Purpose
-h	Provides help.
-d	Sets the debug level. Can be used several times to increase the debug level.
-p	Displays the PathRecord information.
-N	Displays the NodeRecord information.
-D	Displays the NodeDescriptions of channel adapters only.
-S	Displays ServiceRecord information.
-I	Displays InformInfoRecord information.
-L	Returns the LIDs of the specified name.
-l	Returns the unique LID of the specified name.
-G	Returns the GUIDs of the specified name.
-O	Returns the name of the specified LID.
-U	Returns the name of the specified GUID.
-G	Uses the port GUID address.
-c	Displays the Subnet Administrator class port information.

Option	Purpose
-s	Returns the PortInforRecords with the `isSM` or `isSMdisabled` capability mask bit enabled.
-g	Displays multicast group information.
-m	Displays multicast member information. If a group is specified, provides only the GUID and node description for each entry.
-x	Displays LinkRecord information.
--src-to-dst	Displays a PathRecord for *source:destination*, where *source* and *destination* are either node names or LIDs.
--sgid-to-dgid	Displays a PathRecord for *source-destination*, where *source* and *destination* are GIDs in an IPv6 format acceptable to inet_pton.
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.

Example

This example shows how to produce a node record dump of every LID in the InfiniBand fabric with the `saquery` command.

Note – The output in the example is a portion of the full output.

```
FabMan@gateway_name->saquery
NodeRecord dump:
     lid.....................0xE
     reserved................0x0
     base_version............0x1
     class_version...........0x1
     node_type...............Channel Adapter
     num_ports...............0x2
     sys_guid................0x0003ba000100e38b
     node_guid...............0x0003ba000100e388
     port_guid...............0x0003ba000100e38a
     partition_cap...........0x80
     device_id...............0x673C
     revision................0xA0
     port_num................0x2
     vendor_id...............0x2C9
     NodeDescription.........mnm33-43 HCA-1
 NodeRecord dump:
```

```
        lid.....................0xF
        reserved................0x0
.
.
.
FabMan@gateway_name->
```

Related Information

- `saquery` man page
- "Linux Shells for InfiniBand Commands" on page 138

smpquery Command

Queries InfiniBand fabric management attributes.

Syntax

`smpquery [-d][-D][-e][-G][-h][-v][-V][-C ca_name][-P ca_port][-t timeout] op lid|dr_path|guid [params]`

where:

- *ca_name* is the channel adapter name.
- *ca_port* is the channel adapter port.
- *timeout* is the timeout in milliseconds.
- *op* is the supported operation.
- *lid* is the local identifier.
- *dr_path* is the directed path.
- *guid* is the global unique identifier.
- *params* is the parameter of the operation.

Description

This InfiniBand command permits a subset of the standard SMP queries, including these:

- Node information

- Node description
- Switch information
- Port information

Output is in human-readable format.

Supported operations and parameters are as follows:

- `nodeinfo` *addr*
- `nodedesc` *addr*
- `portinfo` *addr* [*portnum*]
- `switchinfo` *addr*
- `pkeys` *addr* [*portnum*]
- `sl2vl` *addr* [*portnum*]
- `vlarb` *addr* [*portnum*]
- `guids` *addr*

where:

- *addr* is the address.
- *portnum* is the port number.

The `smpquery` command is available from the `/SYS/Switch_Diag`, `/SYS/Gateway_Mgmt`, and `/SYS/Fabric_Mgmt` Linux shell targets of the Oracle ILOM CLI interface.

Options

This table describes the options to the `smpquery` command and their purposes.

Option	Purpose
-d	Sets the debug level. Can be used several times to increase the debug level.
-D	Uses the directed path address. The path is a comma-delimited sequence of out ports.
-e	Displays send and receive errors.
-G	Uses the port GUID address.
-h	Provides help.
-v	Provides verbose output.
-V	Displays the version information.

Option	Purpose
-C	Uses the specified channel adapter name.
-P	Uses the specified channel adapter port.
-t	Overrides the default timeout.

Example

This example shows how to display node information for LID 15 with the smpquery command.

```
FabMan@gateway_name->smpquery nodeinfo 15
# Node info: Lid 15
BaseVers:........................1
ClassVers:.......................1
NodeType:........................Switch
NumPorts:........................36
SystemGuid:......................0x0021283a8389a0a3
Guid:............................0x0021283a8389a0a0
PortGuid:........................0x0021283a8389a0a0
PartCap:.........................8
DevId:...........................0xbd36
Revision:........................0x000000a0
LocalPort:.......................1
VendorId:........................0x0002c9
FabMan@gateway_name->
```

Related Information

- smpquery man page
- "saquery Command" on page 167
- "Linux Shells for InfiniBand Commands" on page 138

Understanding SNMP MIB OIDs

These topics provide tables of object identifiers (OIDs) for their respective management information bases (MIBs) in the gateway.

Related Information

OID Tables Overview

Each OID table has three columns of information. From left to right, the columns are:

- **Description of Task or Activity** – The text under this heading describes what you want to do or information you want to see.
- **Action** – The verbs under this heading describe the action respective to the SNMP client software interface. For example, a verb of read means to use the snmpget, snmpwalk, or snmptable command in the NetSNMP CLI client.
- **MIB OID** – The string under this heading is the object identifier of the respective MIB that is used to accomplish the task or activity.

Note – The OID provided in the right column might not be complete. You might need to appended the OID with a `.0` if the OID is a scalar OID or a `.x` (where *x* is 1 to a maximum value) if the OID is a tabular OID.

A table OID has a `Table` suffix, and the OIDs listed beneath it are most likely to be columns of that table.

An index OID has an `Index` suffix, and while the OID is not directly accessible, the OID is still used to return a specific row of a table OID.

Related Information

- "Understanding the SUN-DCS-IB-MIB MIB OIDs" on page 174
- "Understanding the SUN-FABRIC-MIB MIB OIDs" on page 189
- "Understanding the SUN-HW-TRAP-MIB MIB OIDs" on page 198
- "Understanding the SUN-ILOM-CONTROL-MIB MIB OIDs" on page 203
- "Understanding the SUN-PLATFORM-MIB MIB OIDs" on page 213
- "Understanding the ENTITY-MIB MIB OIDs" on page 221

Understanding the `SUN-DCS-IB-MIB` MIB OIDs

This topic provides tables of the hardware, Subnet Manager, Performance Manager, and SNMP trap object identifiers.

Tables of object identifiers are provided in these topics:

- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175
- "Enclosure OIDs" on page 176
- "Enclosure Nodes OIDs" on page 176
- "Neighbor Node OIDs" on page 177
- "Subnet Manager Info OIDs" on page 178
- "Subnet Manager Agent Data Port OIDs" on page 178
- "Understanding Performance Manager Agent OIDs" on page 180
- "Understanding SNMP Trap OIDs" on page 183
- "Understanding Gateway Specific OIDs" on page 187

Related Information

- *"OID Tables Overview" on page 173*
- *"Understanding the SUN-FABRIC-MIB MIB OIDs" on page 189*
- *"Understanding the SUN-HW-TRAP-MIB MIB OIDs" on page 198*
- *"Understanding the SUN-ILOM-CONTROL-MIB MIB OIDs" on page 203*
- *"Understanding the SUN-PLATFORM-MIB MIB OIDs" on page 213*
- *"Understanding the ENTITY-MIB MIB OIDs" on page 221*

`SUN-DCS-IB-MIB` MIB OID Command Syntax

When you use the object identifiers in a command syntax, you must include the `SUN-DCS-IB-MIB` MIB name. For example, using the V2c protocol:

```
$ command -v2c -c public mc_IP SUN-DCS-IB-MIB::object_id argument
```

Or using the V3 protocol:

```
$ command -v3 -u snmp_user -l security_level -a authentication_protocol -A authentication_password
-x DES -X privacy_password mc_IP SUN-DCS-IB-MIB::object_id argument
```

where:

- *command* is the command of the SNMP client that performs the action.
- *snmp_user* is a configured user of the SNMP services.
- *security_level* is:
    - `authNoPriv` – There is authentication, but no privacy.
    - `authPriv` – There is authentication and privacy.
- *authentication_protocol* is either `MD5` or `SHA`
- *authentication_password* is the *snmp_user*'s authentication password.
- *privacy_password* is the *snmp_user*'s privacy password.
- *mc_IP* is the IP address of the management controller.
- *object_id* is the object identifier, as listed in the right column of the tables.
- *argument* is a combination of options and variables that support the object identifier.

For example, to securely display the version of the MIB on the management controller with IP address of 123.45.67.89:

```
$ snmpget -v3 -u usersnmp -l authPriv -a MD5 -A authpass -x DES -X privpass
123.45.67.89 SUN-DCS-IB-MIB::mibVersion
```

Related Information

- "SUN-FABRIC-MIB MIB OID Command Syntax" on page 190
- "SUN-ILOM-CONTROL-MIB MIB OID Command Syntax" on page 204
- "SUN-PLATFORM-MIB MIB OID Command Syntax" on page 214
- "ENTITY-MIB MIB OID Command Syntax" on page 221

Enclosure OIDs

This table lists enclosure object identifiers supported by the SUN-DCS-IB-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display the MIB version.	read	mibVersion
Display the type of platform (Sun DCS GW, and so on).	read	platformName
Display the service processor firmware version.	read	spFwVersion
Display the FPGA/CPLD firmware version.	read	fpgaVersion

Related Information

- "OID Tables Overview" on page 173
- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175

Enclosure Nodes OIDs

This table lists enclosure node object identifiers supported by the SUN-DCS-IB-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display a table of information about the InfiniBand nodes belonging to the enclosure.	read	ibEncNodesTable
Display a row of the ibEncNodesTable.	read	ibEncNodesEntry
Display an index of InfiniBand nodes belonging to the enclosure.	read	ibEncNodesIndex
Display the type of InfiniBand node (switch, CA, and so on).	read	ibEncNodesType
Display the GUID of the node.	read	ibEncNodesGuid
Display the number of ports of the node.	read	ibEncNodesPorts
Display the NodeDescription of the node.	read	ibEncNodesDescr
Display the LID of the node.	read	ibEncNodesLid
Display the firmware version of the node.	read	ibEncNodesFwVersion
Display the board or FRU on which the InfiniBand node is located.	read	ibEncNodesBoard
Display additional identification information of the InfiniBand node.	read	ibEncNodesName

Related Information

-
-

Neighbor Node OIDs

This table lists neighbor node object identifiers supported by the SUN-DCS-IB-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display a table of information about neighboring InfiniBand nodes.	read	ibNeighborTable
Display a row of the ibNodeNeighborTable.	read	ibNeighborEntry
Display an index of neighboring InfiniBand nodes.	read	ibNeighNodeIndex
Display an index of neighboring InfiniBand node data ports.	read	ibNeighPortIndex
Display the type of remote InfiniBand node (switch, CA, and so on).	read	ibNeighRemoteNodeType
Display the GUID of the remote node.	read	ibNeighRemoteNodeGuid

Description of Task or Activity	Action	MIB OID
Display the number of ports of the remote node.	read	ibNeighRemoteNodePort
Display the NodeDescription of the remote node.	read	ibNeighRemoteNodeDescr
Display the LID of the remote node.	read	ibNeighRemoteNodeLid
Display a mapping of neighboring InfiniBand device port to connector.	read	ibNeighPortConnectorPosition

Related Information

- "OID Tables Overview" on page 173
- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175

Subnet Manager Info OIDs

This table lists Subnet Manager object identifiers supported by the SUN-DCS-IB-MIB MIB.

Description of Task or Activity	Action	MIB OID
Identify the master Subnet Manager in the subnet.	read	smMasterSMPresent
Display the LID of the master Subnet Manager.	read	smMasterSMLid
Display the GUID of the master Subnet Manager.	read	smMasterSMGuid
Display information about the Subnet Manager. Disable or enable the Subnet Manager.	read write	smLocalSMRunning
Display the state of the Subnet Manager.	read	smLocalSMState

Related Information

- "OID Tables Overview" on page 173
- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175

Subnet Manager Agent Data Port OIDs

This table lists SMA object identifiers supported by the SUN-DCS-IB-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display a table of SMA `PortInfo` information.	read	`ibSmaPortInfoTable`
Display a row of the `ibSmaPortInfoTable`.	read	`ibSmaPortInfoEntry`
Display an index of SMA InfiniBand nodes.	read	`ibSmaNodeIndex`
Display an index of SMA InfiniBand data ports.	read	`ibSmaPortIndex`
Display SMA enabled link width.	read	`ibSmaPortLinkWidthEnabled`
Display SMA supported link width.	read	`ibSmaPortLinkWidthSupported`
Display SMA currently active link width.	read	`ibSmaPortLinkWidthActive`
Display SMA supported link speed.	read	`ibSmaPortLinkSpeedSupported`
Display the SMA link state of the port.	read	`ibSmaPortLinkState`
Display the SMA physical state of the port.	read	`ibSmaPortPhysState`
Display the SMA `LinkDownDefault` state.	read	`ibSmaPortLinkDownDefaultState`
Display the SMA LID mask control count for multipath support of CA and router ports.	read	`ibSmaPortLidMaskCount`
Display the currently active SMA link speed.	read	`ibSmaPortLinkSpeedActive`
Display the enabled SMA link speed.	read	`ibSmaPortLinkSpeedEnabled`
Display the active maximum MTU enabled on the SMA port for transmit.	read	`ibSmaPortNeighborMtu`
Display the virtual lane configuration supported on the SMA port.	read	`ibSmaPortVirtLaneSupport`
Display the number of high priority packets that are transmitted before a low priority packet is sent.	read	`ibSmaPortVlHighPriorityLimit`
Display the number of virtual lane - weight pairs supported on the SMA port in the `ibSmaHiPriVlArbTable` for high priority.	read	`ibSmaPortVlArbHighCapacity`
Display the number of virtual lane - weight pairs supported on the SMA port in the `ibSmaHiPriVlArbTable` for low priority.	read	`ibSmaPortVlArbLowCapacity`
Display the maximum MTU supported by the SMA port.	read	`ibSmaPortMtuCapacity`
Display or specify the number of sequential packets dropped that causes the SMA port to enter the `VLStalled` state.	read	`ibSmaPortVlStallCount`
Display or specify the time a packet can live at the head of a virtual lane queue.	read	`ibSmaPortHeadOfQueueLife`

Description of Task or Activity	Action	MIB OID
Display the virtual lanes operational on the SMA port.	read	ibSmaPortOperationalVls
Display support of optional inbound partition enforcement.	read	ibSmaPortPartEnforceInbound
Display support of optional outbound partition enforcement.	read	ibSmaPortPartEnforceOutbound
Display support of optional inbound raw packet enforcement.	read	ibSmaPortFilterRawPktInbound
Display support of optional outbound raw packet enforcement.	read	ibSmaPortFilterRawPktOutbound
Display the local physical error threshold value. When the threshold is exceeded, a local link integrity error is determined.	read	ibSmaPortLocalPhysErrorThreshold
Display overrun error threshold value. When the number of buffer overruns exceeds the threshold, an excessive buffer overrun error is determined.	read	ibSmaPortOverrunErrorThreshold
Display a mapping of an InfiniBand device SMA port to connector.	read	ibSmaPortConnectorPosition

Related Information

- "OID Tables Overview" on page 173
- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175

Understanding Performance Manager Agent OIDs

Performance Manager object identifiers are provided in these tables:

- "PMA Port Counters Table OIDs" on page 181
- "PMA Extended Port Counters Table OIDs" on page 182

Related Information

- "Understanding SNMP Trap OIDs" on page 183
- "Understanding Gateway Specific OIDs" on page 187
- "OID Tables Overview" on page 173
- "Understanding the SUN-HW-TRAP-MIB MIB OIDs" on page 198
- "Understanding the SUN-ILOM-CONTROL-MIB MIB OIDs" on page 203

PMA Port Counters Table OIDs

This table lists PMA port counter object identifiers supported by the `SUN-DCS-IB-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display a table of PMA Port Counters information.	read	ibPmaPortCntrsTable
Display a row of the ibPmaPortCntrsTable	read	ibPmaPortCntrsEntry
Display an index of PMA nodes.	read	ibPmaPortCntrsNode
Display an index of the PMA InfiniBand data ports.	read	ibPmaPortCntrsPort
Display the total number of symbol errors detected on one or more physical lanes.	read	ibPmaSymbolErrCounter
Display the number of times the port training state machine has successfully completed the link error recovery process.	read	ibPmaLinkErrRecoveryCntr
Display the number of times the port training state machine has failed the link error recovery process and downed the link.	read	ibPmaLinkDownedCntr
Display the number of packets containing an error that were received on the port.	read	ibPmaPortRcvErr
Display the number of packets marked with the End-of-Bad-Packets (EBP) delimiter received on the port.	read	ibPmaPortRcvRemPhysErr
Display the number of packets received on the port that could not be forwarded by the switch relay.	read	ibPmaPortRcvSwitchRelayErr
Display the number of outbound packets discarded because the port is down or congested.	read	ibPmaPortXmitDiscard
Display the number of packets not transmitted from the port because the packet is raw, or fails partition key check or IP version check.	read	ibPmaPortXmitConstraintErr
Display the number of packets received on the port that are discarded because the packet is raw, or fails partition key check or IP version check.	read	ibPmaPortRcvConstraintErr
Display the number of times that the local physical errors threshold was exceeded.	read	ibPmaLocalLinkIntegrityErr
Display the number of times that buffer overrun errors occurred.	read	ibPmaExcessBufOverrunErr

Description of Task or Activity	Action	MIB OID
Display the number of incoming VL 15 packets dropped due to lack of buffers.	read	ibPmaVl15Dropped
Display a mapping of PMA InfiniBand device port to connector.	read	ibPmaPortConnector
Display the number of symbol errors for a port during the last 10 minutes.	read	ibPmaSymErrors10min
Display the number of symbol errors for a port during the last hour.	read	ibPmaSymErrors1hour
Display the number of symbol errors for a port during the last 24 hours.	read	ibPmaSymErrors24hours

Related Information

- "OID Tables Overview" on page 173
- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175

PMA Extended Port Counters Table OIDs

This table lists extended PMA port counter object identifiers supported by the `SUN-DCS-IB-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display a table of PMA extended port counters information.	read	ibPmaExtPortCntrsTable
Display a row of the `ibPmaExtPortCntrsTable`.	read	ibPmaExtPortCntrsEntry
Display an index of external PMA nodes.	read	ibPmaExtPortCntrsNode
Display an index of external PMA InfiniBand data ports.	read	ibPmaExtPortCntrsPort
Display the number of data octets, divided by 4, transmitted on all virtual lanes from the port.	read	ibPmaPortXmitData
Display the number of data octets, divided by 4, received on all virtual lanes at the port.	read	ibPmaPortRcvData
Display the number of packets transmitted on all virtual lanes from the port.	read	ibPmaPortXmitPkts
Display the number of packets received from all virtual lanes on the port.	read	ibPmaPortRcvPkts
Display the number of unicast packets transmitted on all virtual lanes from the port.	read	ibPmaPortUnicastXmitPkts

Description of Task or Activity	Action	MIB OID
Display the number of unicast packets received from all virtual lanes on the port.	read	ibPmaPortUnicastRcvPkts
Display the number of multicast packets transmitted on all virtual lanes from the port.	read	ibPmaPortMulticastXmitPkts
Display the number of multicast packets received from all virtual lanes on the port.	read	ibPmaPortMulticastRcvPkts
Display a mapping of external PMA InfiniBand device port to connector.	read	ibPmaExtPortConnector

Related Information

- "OID Tables Overview" on page 173
- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175

Understanding SNMP Trap OIDs

SNMP trap object identifiers are provided in these tables:

- "Trap Control OIDs" on page 183
- "Trap Threshold OIDs" on page 184
- "Trap Notification OIDs" on page 185
- "IB Trap OIDs" on page 186

Related Information

- "Understanding Performance Manager Agent OIDs" on page 180
- "Understanding Gateway Specific OIDs" on page 187
- "OID Tables Overview" on page 173
- "Understanding the SUN-HW-TRAP-MIB MIB OIDs" on page 198
- "Understanding the SUN-ILOM-CONTROL-MIB MIB OIDs" on page 203
- "Understanding the SUN-PLATFORM-MIB MIB OIDs" on page 213
- "Understanding the ENTITY-MIB MIB OIDs" on page 221

Trap Control OIDs

This table lists trap control object identifiers supported by the SUN-DCS-IB-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display or set the state whether performance counters traps are generated.	read write	ibSnmpPerfCountersTrapEnable
Display or set whether master Subnet Manager traps are generated.	read write	ibSnmpMasterSMTrapEnable
Display or set whether link state and speed traps are generated.	read write	ibSnmpLinkTrapEnable
Display or set whether high error rate traps are generated.	read write	ibSnmpPerfHighErrorRateTrapEnable
Display or set whether attention traps are generated.	read write	ibSnmpSwitchAttentionEnable

Related Information

-
-

Trap Threshold OIDs

This table lists trap threshold object identifiers supported by the SUN-DCS-IB-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display or set whether a trap is generated, should the SymbolErrCounter threshold be exceeded.	read write	ibSnmpPerfSymbolErrCounterThreshold
Display or set whether a trap is generated, should the LinkErrRecoveryCntr threshold be exceeded.	read write	ibSnmpPerfLinkErrRecoveryCntrThreshold
Display or set whether a trap is generated, should the LinkDownedCntr threshold be exceeded.	read write	ibSnmpPerfLinkDownedCntrThreshold
Display or set whether a trap is generated, should the PortRcvErr threshold be exceeded.	read write	ibSnmpPerfPortRcvErrThreshold

Description of Task or Activity	Action	MIB OID
Display or set whether a trap is generated, should the PortRcvRemPhysErr threshold be exceeded.	read write	ibSnmpPerfPortRcvRemPhysErrThreshold
Display or set whether a trap is generated, should the PortRcvSwitchRelayErr threshold be exceeded.	read write	ibSnmpPerfPortRcvSwitchRelayErrThreshold
Display or set whether a trap is generated, should the PortXmitDiscards threshold be exceeded.	read write	ibSnmpPerfPortXmitDiscardThreshold
Display or set whether a trap is generated, should the PortXmitConstraintErr threshold be exceeded.	read write	ibSnmpPerfPortXmitConstraintErrThreshold
Display or set whether a trap is generated, should the PortRcvConstraintErr threshold be exceeded.	read write	ibSnmpPerfPortRcvConstraintErrThreshold
Display or set whether a trap is generated, should the LocalLinkIntegrityErr threshold be exceeded.	read write	ibSnmpPerfLocalLinkIntegrityErrThreshold
Display or set whether a trap is generated, should the ExcessBufOverrunErr threshold be exceeded.	read write	ibSnmpPerfExcessBufOverrunErrThreshold
Display or set whether a trap is generated, should the Vl15Dropped threshold be exceeded.	read write	ibSnmpPerfVl15DroppedThreshold

Related Information

- "OID Tables Overview" on page 173
- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175

Trap Notification OIDs

This table lists trap notification object identifiers supported by the SUN-DCS-IB-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display an index of InfiniBand nodes originating a trap.	read	ibSnmpTrapNodeIndex
Display an index of InfiniBand data ports originating a trap.	read	ibSnmpTrapPortIndex
Display the performance counter causing a trap.	read	ibSnmpTrapPerfErrCounter

Description of Task or Activity	Action	MIB OID
Display the counter value causing a trap.	read	ibSnmpTrapCounterVal
Display a mapping of SNMP InfiniBand device port to connector causing a trap.	read	ibSnmpTrapPortConnector
Identify the master Subnet Manager causing a trap.	read	ibSnmpTrapMasterSMPresent
Display the LID of the master Subnet Manager of a trap.	read	ibSnmpTrapMasterSMLid
Display the currently active link width of a trap.	read	ibSnmpTrapLinkWidthActive
Display the state of the link on the port of a trap.	read	ibSnmpTrapLinkState
Display the currently active link speed of a trap.	read	ibSnmpTrapLinkSpeedActive
Display the GUID of the trap's node.	read	ibSnmpTrapNodeGuid
Display the LID of the trap's node.	read	ibSnmpTrapNodeLid
Display the NodeDescription of the trap's node.	read	ibSnmpTrapNodeDescr
Display how much the symbol error counter increased during the error rate monitoring interval.	read	ibSnmpSymErrIncrease
Display and set the error rate monitoring interval.	read write	ibSnmpErrRateInterval
Display a description of what requires attention.	read	ibSnmpAttDecription
Display the perceived severity of the issue requiring attention.	read	ibSnmpAttSeverity
Display the probable cause of the issue requiring attention.	read	ibSnmpAttProbableCause
Display the repair action for the issue requiring attention.	read	ibSnmpAttRepairAction

Related Information

- "OID Tables Overview" on page 173
- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175

IB Trap OIDs

This table lists the InfiniBand trap object identifiers supported by the SUN-DCS-IB-MIB MIB.

Trap or Notification	Description
ibSnmpPerfErrCounterTrap	A performance error counter threshold was exceeded. Node, port, counter and its value are passed inside the trap.
ibSnmpNoMasterSMTrap	Trap is generated when a change happens in master Subnet Manager presence.
ibSnmpLinkTrap	Trap is generated when a change in LinkState, LinkSpeedActive, or LinkWidthActive happens on a port.
ibSnmpHighErrRateTrap	Trap is generated when error rate on a port is high (according to the configured values).
ibSnmpSwitchAttention	Trap is generated when there is low disk space in the filesystem.

Related Information

- "OID Tables Overview" on page 173
- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175

Understanding Gateway Specific OIDs

Gateway specific object identifiers are provided in these tables:

- "Gateway Port Status OIDS" on page 188
- "Gateway Port Counter OIDS" on page 188

Related Information

- "Understanding Performance Manager Agent OIDs" on page 180
- "Understanding SNMP Trap OIDs" on page 183
- "OID Tables Overview" on page 173
- "Understanding the SUN-HW-TRAP-MIB MIB OIDs" on page 198
- "Understanding the SUN-ILOM-CONTROL-MIB MIB OIDs" on page 203
- "Understanding the SUN-PLATFORM-MIB MIB OIDs" on page 213
- "Understanding the ENTITY-MIB MIB OIDs" on page 221

Gateway Port Status OIDS

This table lists gateway port status object identifiers supported by the
SUN-DCS-IB-MIB MIB.

Description of Task or Activity	Action	MIB OID
Displays a table of gateway port information.	read	gwPortStateTable
Displays a row of the gwPortStateTable.	read	gwPortStateEntry
Display the gateway's port's short names.	read	gwPortShortName
Display the gateway's port's long names.	read	gwPortLongName
Display the BridgeX manager's port's state.	read	gwPortState
Display the BridgeX manager's port's physical link state.	read	gwPortLinkState
Identifies that a port is enabled.	read	gwPortEnabled
Display the port protocol as either Ethernet or Fiber Channel.	read	gwPortProtocol
Display the Ethernet protocol.	read	gwPortEthProto
Display the supported Ethernet protocols.	read	gwPortEthSupportProto
Display the Ethernet MTU.	read	gwPortEthMTU
Display the Ethernet transmit pause.	read	gwPortEthTxPause
Display the Ethernet receive pause.	read	gwPortEthRxPause

Related Information

- "OID Tables Overview" on page 173
- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175

Gateway Port Counter OIDS

This table lists gateway port counter object identifiers supported by the
SUN-DCS-IB-MIB MIB.

Description of Task or Activity	Action	MIB OID
Displays a table of gateway port counters.	read	gwEthPortCntrsTable
Displays a row of the gwEthPortCntrsTable.	read	gwEthPortCntrsEntry
Display the gateway's port's short name.	read	gwEthPortName
Display the Ethernet bytes received counter.	read	gwEthRxBytes

Description of Task or Activity	Action	MIB OID
Display the Ethernet packets received counter.	read	gwEthRxPkts
Display the Ethernet Jumbo packets received counter.	read	gwEthRxJumboPkts
Display the Ethernet unicast packets received counter.	read	gwEthRxUcastPkts
Display the Ethernet multicast packets received counter.	read	gwEthRxMcastPkts
Display the Ethernet broadcast packets received counter.	read	gwEthRxBcastPkts
Display the Ethernet No Buffer received counter.	read	gwEthRxNoBuffer
Display the Ethernet CRC received counter.	read	gwEthRxCRC
Display the Ethernet runt received counter.	read	gwEthRxRunt
Display the Ethernet errors received counter.	read	gwEthRxErrors
Display the Ethernet bytes transmitted counter.	read	gwEthTxBytes
Display the Ethernet packets transmitted counter.	read	gwEthTxPkts
Display the Ethernet Jumbo packets transmitted counter.	read	gwEthTxJumboPkts
Display the Ethernet unicast packets transmitted counter.	read	gwEthTxUcastPkts
Display the Ethernet multicast packets transmitted counter.	read	gwEthTxMcastPkts
Display the Ethernet broadcast packets transmitted counter.	read	gwEthTxBcastPkts
Display the Ethernet errors transmitted counter.	read	gwEthTxErrors

Related Information

- "OID Tables Overview" on page 173
- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175

Understanding the `SUN-FABRIC-MIB` MIB OIDs

This topic provides tables of the fabric element, management, and trap object identifiers.

Tables of object identifiers are provided in these topics:

- "SUN-FABRIC-MIB MIB OID Command Syntax" on page 190
- "Fabric Element OIDs" on page 191
- "Fabric Management OIDs" on page 193

Related Information

`SUN-FABRIC-MIB` MIB OID Command Syntax

When you use the object identifiers in a command syntax, you must include the `SUN-FABRIC-MIB` MIB name. For example, using the V2c protocol:

```
$ command -v2c -c public mc_IP SUN-FABRIC-MIB::object_id argument
```

Or using the V3 protocol:

```
$ command -v3 -u snmp_user -l security_level -a authentication_protocol -A authentication_password
-x DES -X privacy_password mc_IP SUN-FABRIC-MIB::object_id argument
```

where:

- *command* is the command of the SNMP client that performs the action.
- *snmp_user* is a configured user of the SNMP services.
- *security_level* is:
  - `authNoPriv` – There is authentication, but no privacy.
  - `authPriv` – There is authentication and privacy.
- *authentication_protocol* is either `MD5` or `SHA`
- *authentication_password* is the *snmp_user*'s authentication password.
- *privacy_password* is the *snmp_user*'s privacy password.
- *mc_IP* is the IP address of the management controller.

- *object_id* is the object identifier, as listed in the right column of the tables.
- *argument* is a combination of options and variables that support the object identifier.

For example, to securely display the IP address of the Fabric Manager:

```
$ snmpget -v3 -u usersnmp -l authPriv -a MD5 -A authpass -x DES -X privpass
123.45.67.89 SUN-FABRIC-MIB::fabricElemMgrIpAddress
```

Related Information

Fabric Element OIDs

This table lists fabric element object identifiers supported by the `SUN-FABRIC-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display the host name and IP address of the fabric element.	read	fabricElemName
Display the product name of a fabric element.	read	fabricElemType
Display the current aggregated operational state of the fabric element.	read	fabricElemOperStatus
Display the IP address of the fabric manager.	read	fabricElemMgrIpAddress
Display whether the fabric manager is currently available to the fabric element.	read	fabricElemMgrAlive
Display the type of fabric the element supports (InfiniBand or Ethernet).	read	fabricElemFabType
Display the number of logical ports the element supports.	read	fabricElemNumPorts
Display a table of information about the logical ports associated with a specific fabric element.	read	fabricElemPortTable
Display information about a specific fabric element.	read	fabricElemPortEntry
Display an integer which uniquely identifies the instance among all of the fabric element's port instances.	read	fabricElemPortIndex

Description of Task or Activity	Action	MIB OID
Display the logical port identifier for the fabric element.	read	fabricElemPortID
Display the status for the specific port.	read	fabricElemPortOperStatus
Display the physical address of neighbor/remote elements. For Ethernet it is the MAC address, for IB it is the GUID + port.	read	fabricElemPortRemAddr
Display the ID of the connector table entry associated with the fabric element.	read	fabricElemPortConnID
Display the fabric element port's position in the connector. For single port connectors, the position is always 1.	read	fabricElemPortConnPosition
Display the physical address associated with the local port. For Ethernet it is the MAC address, for InfiniBand it is the GUID + port.	read	fabricElemPortLocalAddr
Display the number of connectors in the connector table.	read	fabricElemNumConnectors
Display a table of information about the physical connectors associated with a specific fabric element.	read	fabricElemConnectorTable
Display information about a specific fabric element's connector.	read	fabricElemConnEntry
Display an integer which uniquely identifies the instance among all of the fabric element's connector instances.	read	fabricElemConnIndex
Display the connector type. The types are: • cable based - QSFP, CXP, SFP, and so on • chassis/backplane based • chip to chip connections	read	fabricElemConnType
Display the network type supported by the connector.	read	fabricElemConnNetworkType
Display the number of ports associated with the connector.	read	fabricElemConnNumPorts
Display the connector position in the fabric enclosure.	read	fabricElemConnDescr
Display the identifier for the line card associated with the connector.	read	fabricElemConnCardNum
Display the identifier for the connector.	read	fabricElemConnID
Display the roles assigned to the fabric element.	read	fabricElemRoles
Display the datacenter positional information (x coordinate).	read	fabricElemRackPosX
Display the datacenter positional information (y coordinate).	read	fabricElemRackPosY
Display the datacenter positional information (z coordinate). This coordinate is also the position within the rack.	read	fabricElemRackPosZ

Related Information

- "OID Tables Overview" on page 173
- "SUN-FABRIC-MIB MIB OID Command Syntax" on page 190

Fabric Management OIDs

This table lists fabric management object identifiers supported by the `SUN-FABRIC-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display the name of the fabric.	read	fabricMgmtFabricName
Display or set the description of the fabric.	read	fabricMgmtFabricDescr
Display the type of fabric (InfiniBand or Ethernet).	read	fabricMgmtFabricType
Display the topology of the fabric.	read	fabricMgmtFabricTopology
Display a summary of operational health and status for the fabric.	read	fabricMgmtFabricOperStatus
Display the last trap sequence number issued by the fabric manager.	read	fabricMgmtFabricLastTrapSeqNum
Display the number of elements in the fabric element table.	read	fabricMgmtNumFabricElems

Related Information

- "OID Tables Overview" on page 173
- "SUN-FABRIC-MIB MIB OID Command Syntax" on page 190

Fabric Management Element OIDs

This table lists fabric management element object identifiers supported by the `SUN-FABRIC-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display a table of information about the fabric elements.	read	fabricMgmtElemTable
Display information about a specific fabric element.	read	fabricMgmtElemEntry
Display an integer which uniquely identifies the instance among all of the fabric element instances.	read	fabricMgmtElemIndex
Display the product name.	read	fabricMgmtElemType
Display a logical value which uniquely identifies the specific fabric element.	read	fabricMgmtElemID
Display the roles assigned to the fabric element.	read	fabricMgmtElemRoles
Display whether the fabric element is visible on the fabric or the management network.	read	fabricMgmtElemPresent
Display whether the element is part of the fabric definition or blue print.	read	fabricMgmtElemExpected
Display the current aggregated operational state of the fabric element.	read	fabricMgmtElemOperStatus
Display the name of the fabric element.	read	fabricMgmtElemName
Display or set the description of the fabric element.	read	fabricMgmtElemDescr
Display the IP address used for management access to the fabric element.	read	fabricMgmtElemIpAddress
Display the number of external ports on the fabric element.	read	fabricMgmtElemNumExtPorts
Display the datacenter positional information (x coordinate).	read	fabricMgmtElemRackPosX
Display the datacenter positional information (y coordinate).	read	fabricMgmtElemRackPosY
Display the datacenter positional information (z coordinate).	read	fabricMgmtElemRackPosZ
Display the IP address of the management controller.	read	fabricMgmtElemChassisMgrAddr
Display the position of the fabric element in the chassis.	read	fabricMgmtElemChassisPos

Related Information

- "OID Tables Overview" on page 173
- "SUN-FABRIC-MIB MIB OID Command Syntax" on page 190

Fabric Management External Element OIDs

This table lists fabric management external element object identifiers supported by the `SUN-FABRIC-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display the number of elements in the external element table.	read	`fabricMgmtNumExtElems`
Display a table of information about external elements connected to the fabric.	read	`fabricMgmtExtElemTable`
Display information about a specific external element.	read	`fabricMgmtExtElemEntry`
Display an integer which uniquely identifies the instance among all of the external element instances.	read	`fabricMgmtExtElemIndex`
Display the name of the fabric external element. For a host, it is the host name.	read	`fabricMgmtExtElemName`
Display a logical value which uniquely identifies the specific external element.	read	`fabricMgmtExtElemID`
Display a description of the external element.	read	`fabricMgmtExtElemDescr`
Display the physical address of the fabric external element. For Ethernet it is the MAC address, for InfiniBand it is the GUID.	read	`fabricMgmtExtElemPhysAddress`

Related Information

- "OID Tables Overview" on page 173
- "SUN-FABRIC-MIB MIB OID Command Syntax" on page 190

Fabric Management Gateway OIDs

This table lists fabric management gateway object identifiers supported by the `SUN-FABRIC-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display the number of elements in the gateway element table.	read	`fabricMgmtNumGatewayElems`
Display a table of information about fabric gateways.	read	`fabricMgmtGatewayTable`
Display information about a specific fabric gateway.	read	`fabricMgmtGatewayEntry`

Description of Task or Activity	Action	MIB OID
Display an integer which uniquely identifies the instance among all of the gateway instances.	read	fabricMgmtGatewayIndex
Display a logical value which uniquely identifies the gateway.	read	fabricMgmtGatewayID
Display the logical identifier for the corresponding fabric element in the fabric element table which supports the gateway functionality.	read	fabricMgmtGatewayFabElemID
Display the type of gateway functionality.	read	fabricMgmtGatewayType

Related Information

- "OID Tables Overview" on page 173
- "SUN-FABRIC-MIB MIB OID Command Syntax" on page 190

Fabric Management Link OIDs

This table lists fabric management link object identifiers supported by the SUN-FABRIC-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display the number of link elements in the link table.	read	fabricMgmtNumLinkElems
Display a table of information about fabric links.	read	fabricMgmtLinkTable
Display information about a specific link between elements (fabric or external).	read	fabricMgmtLinkEntry
Display an integer which uniquely identifies the instance among all of the link instances.	read	fabricMgmtLinkIndex
Display a description of the fabric link.	read	fabricMgmtLinkDescr
Display a logical value which uniquely identifies the link.	read	fabricMgmtLinkID
Display the presence of a link.	read	fabricMgmtLinkPresent
Display the state of the fabric link.	read	fabricMgmtLinkOperStatus
Display the type of link: • fabric-internal • fabric-external • gateway-external • gateway-internal • element-internal	read	fabricMgmtLinkType

Description of Task or Activity	Action	MIB OID
Display the network type of the link (InfiniBand, Ethernet, Fibre Channel).	read	fabricMgmtLinkNetworkType
Display the type of element at the end of the link (external, fabric, or gateway).	read	fabricMgmtLinkEnd1Type
Display the identifier of the element in the external, fabric, or gateway element table where the link terminates.	read	fabricMgmtLinkEnd1ElemID
Display the logical port identifier where the link terminates.	read	fabricMgmtLinkEnd1Port
Display the physical address of the end point. For Ethernet links it is the MAC address, for InfiniBand links it is the GUID, for Fibre Channel links it is the WWN.	read	fabricMgmtLinkEnd1Addr
Display the type of element at the end of the link.	read	fabricMgmtLinkEnd2Type
Display the identifier of the element in the external, fabric, or gateway element table where the link terminates.	read	fabricMgmtLinkEnd2ElemID
Display the logical port identifier where the link terminates.	read	fabricMgmtLinkEnd2Port
Display the physical address of the end point. For Ethernet links it is the MAC address, for InfiniBand links it is the GUID, for Fibre Channel links it is the WWN.	read	fabricMgmtLinkEnd2Addr

Related Information

- "OID Tables Overview" on page 173
- "SUN-FABRIC-MIB MIB OID Command Syntax" on page 190

Fabric Trap Configuration OIDs

This table lists fabric trap configuration object identifiers supported by the SUN-FABRIC-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display or set whether fabricStateChange traps are generated.	read write	fabricStateTrapEnable
Display or set whether fabricElemTableStateChange traps are generated.	read write	fabricElemStateTrapEnable
Display or set whether fabricLinkTableStateChange traps are generated.	read write	fabricLinkStateTrapEnable

Description of Task or Activity	Action	MIB OID
Display or set whether `fabricManagerFailover` traps are generated.	read write	`fabricManagerFailoverTrapEnable`
Display the logical identifier of the fabric element responsible for the trap.	read	`fabricTrapElemID`
Display the index identifier of the fabric link responsible for the trap.	read	`fabricTrapLinkID`
Display the operational state for the trapping entity.	read	`fabricTrapOperStatus`
Display the present status of the trapping entity.	read	`fabricTrapPresentStatus`
Display the type of presence for the fabric element (fabric or management).	read	`fabricTrapElemPresentType`
Display the monotonically increasing sequence number of the trap.	read	`fabricTrapSeqNumber`
Display the name of the fabric.	read	`fabricTrapFabricName`
Display the IP address of the fabric manager.	read	`fabricTrapMgrIpAddress`
Display the identifier of the fabric element sending the trap.	read	`fabricTrapSenderID`
Display the identifier of the previous fabric manager.	read	`fabricTrapOldMgrID`
Display the identifier of the new fabric manager.	read	`fabricTrapNewMgrID`

Related Information

Understanding the `SUN-HW-TRAP-MIB` MIB OIDs

This topic provides tables of the hardware trap object identifiers and the traps.

Tables of object identifiers are provided in these topics:

Related Information

Hardware Trap Configuration OIDs

This table lists hardware trap configuration object identifiers supported by the `SUN-HW-TRAP-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display the text string used to identify the source of the trap. Typically the host name.	`accessible-for-notify`	`sunHwTrapSystemIdentifier`
Display the name of the component which is the source of the trap.	`accessible-for-notify`	`sunHwTrapComponentName`
Display the threshold type that the sensor is reporting.	`accessible-for-notify`	`sunHwTrapThresholdType`
Display the threshold value that has been exceeded.	`accessible-for-notify`	`sunHwTrapThresholdValue`
Display the threshold sensor's reading at the time of the trap.	`accessible-for-notify`	`sunHwTrapSensorValue`
Display an optional description which provides additional information.	`accessible-for-notify`	`sunHwTrapAdditionalInfo`
Display the object in the `entPhysicalTable` to which the notification applies.	`accessible-for-notify`	`sunHwTrapAssocObjectId`
Display an indication of the severity of the notification.	`accessible-for-notify`	`sunHwTrapSeverity`
Display the text string containing the chassis serial number.	`accessible-for-notify`	`sunHwTrapChassisId`

Description of Task or Activity	Action	MIB OID
Display the text string containing the product name.	accessible-for-notify	sunHwTrapProductName
Display an indication as to why the component was disabled.	accessible-for-notify	sunHwTrapDisableReason
Display the message used by the test trap.	accessible-for-notify	sunHwTrapTestMessage

Related Information

-

Hardware Traps

This table lists the hardware traps supported by the SUN-HW-TRAP-MIB MIB.

Trap or Notification	Description
sunHwTrapVoltageFatalThresholdExceeded	Trap is generated when a voltage sensor's value goes above an upper fatal threshold setting or below a lower fatal threshold value.
sunHwTrapVoltageFatalThresholdDeasserted	Trap is generated when a voltage sensor's value goes below an upper fatal threshold setting or above a lower fatal threshold value.
sunHwTrapVoltageCritThresholdExceeded	Trap is generated when a voltage sensor's value goes above an upper critical threshold setting or below a lower critical threshold value.
sunHwTrapVoltageCritThresholdDeasserted	Trap is generated when a voltage sensor's value goes below an upper critical threshold setting or above a lower critical threshold value.
sunHwTrapVoltageNonCritThresholdExceeded	Trap is generated when a voltage sensor's value goes above an upper non-critical threshold setting or below a lower non-critical threshold value.
sunHwTrapVoltageOk	Trap is generated when a voltage sensor's value returns to its normal operating range
sunHwTrapTempFatalThresholdExceeded	Trap is generated when a temperature sensor's value goes above an upper fatal threshold setting or below a lower fatal threshold value.
sunHwTrapTempFatalThresholdDeasserted	Trap is generated when a temperature sensor's value goes below an upper fatal threshold setting or above a lower fatal threshold value.

Trap or Notification	Description
sunHwTrapTempCritThresholdExceeded	Trap is generated when a temperature sensor's value goes above an upper critical threshold setting or below a lower critical threshold value.
sunHwTrapTempCritThresholdDeasserted	Trap is generated when a temperature sensor's value goes below an upper critical threshold setting or above a lower critical threshold value.
sunHwTrapTempNonCritThresholdExceeded	Trap is generated when a temperature sensor's value goes above an upper non-critical threshold setting or below a lower non-critical threshold value.
sunHwTrapTempOk	Trap is generated when a temperature sensor's value returns to its normal operating range
sunHwTrapFanSpeedFatalThresholdExceeded	Trap is generated when a fan speed sensor's value goes above an upper fatal threshold setting or below a lower fatal threshold value.
sunHwTrapFanSpeedFatalThresholdDeasserted	Trap is generated when a fan speed sensor's value goes below an upper fatal threshold setting or above a lower fatal threshold value.
sunHwTrapFanSpeedCritThresholdExceeded	Trap is generated when a fan speed sensor's value goes above an upper critical threshold setting or below a lower critical threshold value.
sunHwTrapFanSpeedCritThresholdDeasserted	Trap is generated when a fan speed sensor's value goes below an upper critical threshold setting or above a lower critical threshold value.
sunHwTrapFanSpeedNonCritThresholdExceeded	Trap is generated when a fan speed sensor's value goes above an upper non-critical threshold setting or below a lower non-critical threshold value.
sunHwTrapFanSpeedOk	Trap is generated when a fan speed sensor's value returns to its normal operating range
sunHwTrapComponentError	Trap is generated when a generic type sensor detects an error. (All aggregate sensors are generic type sensors. Refer to *Gateway Remote Management*, aggregate sensor states.) The sunHwTrapComponentName and sunHwTrapAdditionalInfo objects provide more information on the sensor and its value.
sunHwTrapComponentOk	Trap is generated when a generic type sensor returns to its normal state. The sunHwTrapComponentName and sunHwTrapAdditionalInfo objects provide more information on the sensor and its value.

Trap or Notification	Description
sunHwTrapFruInserted	Trap is generated when a field replaceable unit like a power supply or fan module is inserted into the system.
sunHwTrapFruRemoved	Trap is generated when a field replaceable unit like a power supply or fan module is removed from the system.
sunHwTrapTestTrap	Test trap.

Note – For all threshold traps, the sunHwTrapThresholdType object indicates whether the threshold is an upper threshold or a lower threshold.

Related Information

- "Hardware Fault Traps" on page 202
- "OID Tables Overview" on page 173

Hardware Fault Traps

This table lists the hardware fault traps supported by the SUN-HW-TRAP-MIB MIB.

Trap or Notification	Description
sunHwTrapPowerSupplyFault	Trap is generated on a power supply fault.
sunHwTrapPowerSupplyFaultCleared	Trap is generated when the power supply fault is cleared.
sunHwTrapFanFault	Trap is generated on a fan fault.
sunHwTrapFanFaultCleared	Trap is generated when the fan fault is cleared.
sunHwTrapComponentFault	Trap is generated on either a flash disk lifetime or real time clock battery fault.
sunHwTrapComponentFaultCleared	Trap is generated when either a flash disk lifetime or real time clock battery fault is cleared.

Related Information

- "Hardware Traps" on page 200
- "OID Tables Overview" on page 173

Understanding the SUN-ILOM-CONTROL-MIB MIB OIDs

This topic provides many tables of the Oracle ILOM object identifiers.

Tables of object identifiers are provided in these topics:

Related Information

SUN-ILOM-CONTROL-MIB MIB OID Command Syntax

When you use the object identifiers in a command syntax, you must include the SUN-ILOM-CONTROL-MIB MIB name. For example, using the V2c protocol:

```
$ command -v2c -c public mc_IP SUN-ILOM-CONTROL-MIB::object_id argument
```

Or using the V3 protocol:

```
$ command -v3 -u snmp_user -l security_level -a authentication_protocol -A authentication_password
-x DES -X privacy_password mc_IP SUN-ILOM-CONTROL-MIB::object_id argument
```

where:

- *command* is the command of the SNMP client that performs the action.
- *snmp_user* is a configured user of the SNMP services.
- *security_level* is:
  - authNoPriv – There is authentication, but no privacy.
  - authPriv – There is authentication and privacy.
- *authentication_protocol* is either MD5 or SHA
- *authentication_password* is the *snmp_user*'s authentication password.
- *privacy_password* is the *snmp_user*'s privacy password.
- *mc_IP* is the IP address of the management controller.
- *object_id* is the object identifier, as listed in the right column of the tables.
- *argument* is a combination of options and variables that support the object identifier.

For example, to securely display a table of information about the current local users on the management controller with IP address of 123.45.67.89:

```
$ snmpwalk -v3 -u usersnmp -l authPriv -a MD5 -A authpass -x DES -X privpass
123.45.67.89 SUN-ILOM-CONTROL-MIB::ilomCtrlLocalUserTable
```

Related Information

- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175
- "SUN-FABRIC-MIB MIB OID Command Syntax" on page 190
- "SUN-PLATFORM-MIB MIB OID Command Syntax" on page 214
- "ENTITY-MIB MIB OID Command Syntax" on page 221

NTP OIDs

This table lists network time protocol object identifiers supported by the `SUN-ILOM-CONTROL-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display or set the IP address of the first NTP server used by the device.	read write	ilomCtrlDeviceNTPServerOneIP
Display or set the IP address of the second NTP server used by the device.	read write	ilomCtrlDeviceNTPServerTwoIP

Related Information

-
-

Remote Syslog OIDs

This table lists remote syslog object identifiers supported by the `SUN-ILOM-CONTROL-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display or set the IP address of the first remote syslog destination (log host).	read write	ilomCtrlRemoteSyslogDest1
Display or set the IP address of the second remote syslog destination (log host).	read write	ilomCtrlRemoteSyslogDest2

Related Information

-
-

HTTP OIDs

This table lists HTTP service object identifiers supported by the `SUN-ILOM-CONTROL-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display or set whether or not the embedded web server is running and listening on the HTTP port.	read write	ilomCtrlHttpEnabled
Display or set the port number that the embedded web server should listen to for HTTP requests.	read write	ilomCtrlHttpPortNumber
Display or set whether or not the embedded web server should redirect HTTP connections to HTTPS.	read write	ilomCtrlHttpSecureRedirect

Related Information

- "OID Tables Overview" on page 173
- "SUN-ILOM-CONTROL-MIB MIB OID Command Syntax" on page 204

HTTPS OIDs

This table lists HTTPS service object identifiers supported by the SUN-ILOM-CONTROL-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display or set whether or not the embedded web server is running and listening on the HTTPS port.	read write	ilomCtrlHttpsEnabled
Display or set the port number that the embedded web server should listen to for HTTPS requests.	read write	ilomCtrlHttpsPortNumber

Related Information

- "OID Tables Overview" on page 173
- "SUN-ILOM-CONTROL-MIB MIB OID Command Syntax" on page 204

Network OIDs

This table lists network object identifiers supported by the SUN-ILOM-CONTROL-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display a table of targets whose networks can be controlled.	read	ilomCtrlNetworkTable
Display information about a target which can be reset.	read	ilomCtrlNetworkEntry
Display the MAC address of the service processor or system controller.	read	ilomCtrlNetworkMacAddress
Display whether the current target is configured to have static IP settings or whether these settings are retrieved dynamically from DHCP.	read	ilomCtrlNetworkIpDiscovery
Display the current IP address for the given target.	read	ilomCtrlNetworkIpAddress
Display the current IP gateway for the given target.	read	ilomCtrlNetworkIpGateway
Display the current IP netmask for the given target.	read	ilomCtrlNetworkIpNetmask
Display or set the pending value for the mode of IP discovery for the given target.	read write	ilomCtrlNetworkPendingIpDiscovery
Display or set the pending IP address for the given target.	read write	ilomCtrlNetworkPendingIpAddress
Display or set the pending IP gateway for the given target.	read write	ilomCtrlNetworkPendingIpGateway
Display or set the pending IP netmask for the given target.	read write	ilomCtrlNetworkPendingIpNetmask
Commit pending properties.	read write	ilomCtrlNetworkCommitPending
Display the address of the DHCP server for the given target.	read	ilomCtrlNetworkDHCPServerAddr
Display or set whether or not the parameters are enabled.	read write	ilomCtrlNetworkState

Related Information

- "OID Tables Overview" on page 173
- "SUN-ILOM-CONTROL-MIB MIB OID Command Syntax" on page 204

User OIDs

This table lists user object identifiers supported by the `SUN-ILOM-CONTROL-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display a table of information about the current local users with their password state.	read	`ilomCtrlLocalUserTable`
Display information about a local user in the database.	read	`ilomCtrlLocalUserEntry`
Set the password of a local user on the device.	read write	`ilomCtrlLocalUserPassword`
Display or set the role that is associated with a user.	read write	`ilomCtrlLocalUserRoles`
Create a new user or delete an existing user.	write	`ilomCtrlLocalUserRowStatus`
Display or set the CLI mode a user is configured.	read write	`ilomCtrlLocalUserCLIMode`

Related Information

- "OID Tables Overview" on page 173
- "SUN-ILOM-CONTROL-MIB MIB OID Command Syntax" on page 204

Session OIDs

This table lists session object identifiers supported by the `SUN-ILOM-CONTROL-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display a table of information about the current user sessions.	read	`ilomCtrlSessionsTable`
Display the user name of the user associated with the session.	read	`ilomCtrlSessionsUsername`
Display the type of connection that the given user is using to access the device.	read	`ilomCtrlSessionsConnectionType`
Display the date and time that the user logged into the device.	read	`ilomCtrlSessionsLoginTime`

Related Information

- "OID Tables Overview" on page 173
- "SUN-ILOM-CONTROL-MIB MIB OID Command Syntax" on page 204

Event Log OIDs

This table lists event log object identifiers supported by the
SUN-ILOM-CONTROL-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display a table of information about the current entries in the event log.	read	ilomCtrlEventLogTable
Display an integer representing the type of event.	read	ilomCtrlEventLogType
Display the date and time that the event log entry was recorded.	read	ilomCtrlEventLogTimestamp
Display an integer representing the class of event.	read	ilomCtrlEventLogClass
Display the event severity corresponding to the given log entry.	read	ilomCtrlEventLogSeverity
Display the description of the event.	read	ilomCtrlEventLogDescription
Clear the event log.	write	ilomCtrlEventLogClear

Related Information

- "OID Tables Overview" on page 173
- "SUN-ILOM-CONTROL-MIB MIB OID Command Syntax" on page 204

Alert OIDs

This table lists alert object identifiers supported by the SUN-ILOM-CONTROL-MIB
MIB.

Description of Task or Activity	Action	MIB OID
Display a table of information used to view and add alert rules.	read	ilomCtrlAlertsTable
Display or set the minimum event severity which should trigger an alert, for a given class.	read write	ilomCtrlAlertSeverity
Display or set the type of notification for a given alert.	read write	ilomCtrlAlertType
Display or set the IP address to receive alert notifications when the alert type is snmptrap(2) or ipmipet(3).	read write	ilomCtrlAlertDestinationIP
Display or set the email address to send alert notifications when the alert type is email(1).	read write	ilomCtrlAlertDestinationEmail
Display or set the version of SNMP trap that is used for the given alert rule.	read write	ilomCtrlAlertSNMPVersion
Display or set the community string to be used when the ilomCtrlAlertSNMPVersion property is set to v1 or v2c. Display or set the user name to be used when the ilomCtrlAlertSNMPVersion property is set to v3.	read write	ilomCtrlAlertSNMPCommunityOrUsername
Display or set the destination port for SNMP traps.	read write	ilomCtrlAlertDestinationPort
Display or set the class name to filter emailed alerts on.	read write	ilomCtrlAlertEmailEventClassFilter
Display or set the type name to filter emailed alerts on.	read write	ilomCtrlAlertEmailEventTypeFilter
Display or set an optional format to identify the sender or the from address.	read write	ilomCtrlAlertEmailCustomSender
Display or set an optional string added to the beginning of the message body.	read write	ilomCtrlAlertEmailMessagePrefix

Related Information

- "OID Tables Overview" on page 173
- "SUN-ILOM-CONTROL-MIB MIB OID Command Syntax" on page 204

Clock OIDs

This table lists clock object identifiers supported by the `SUN-ILOM-CONTROL-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display or set the date and time of the device.	read write	`ilomCtrlDateAndTime`
Display or set whether or not Network Time Protocol is enabled.	read write	`ilomCtrlNTPEnabled`
Display or set the configured time zone string.	read write	`ilomCtrlTimezone`

Related Information

- "OID Tables Overview" on page 173
- "SUN-ILOM-CONTROL-MIB MIB OID Command Syntax" on page 204

Backup and Restore OIDs

This table lists backup and restore object identifiers supported by the `SUN-ILOM-CONTROL-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Set the target destination of a configuration XML file during backup and restore. The syntax is: `{protocol}://[user:passwword]@]host[/][path/][file]`.	write	`ilomCtrlBackupAndRestoreTargetURI`
Set the passphrase for encrypting or decrypting sensitive data during backup and restore.	write	`ilomCtrlBackupAndRestorePassphrase`
Set the action to backup or restore.	read write	`ilomCtrlBackupAndRestoreAction`
Display the current status of backup or restore.	read	`ilomCtrlBackupAndRestoreActionStatus`

Related Information

- "OID Tables Overview" on page 173

Identification OIDs

This table lists identification object identifiers supported by the `SUN-ILOM-CONTROL-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display or set the system identifier, which is often the host name of the server associated with Oracle ILOM to be sent out in the `varbind` for all traps that Oracle ILOM generates.	read write	ilomCtrlSystemIdentifier
Display or set the host name for Oracle ILOM.	read write	ilomCtrlHostName

Related Information

SMTP OIDs

This table lists SMTP object identifiers supported by the `SUN-ILOM-CONTROL-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display or set whether or not the SMTP client is enabled.	read write	ilomCtrlSMTPEnabled
Display or set the IP address of the SMTP server used as a name service for user accounts.	read write	ilomCtrlSMTPServerIP
Display or set the port number for the SMTP client.	read write	ilomCtrlSMTPPortNumber
Display or set an optional format to identify the sender or the from address.	read write	ilomCtrlSMTPCustomSender

Related Information

DNS OIDs

This table lists DNS object identifiers supported by the `SUN-ILOM-CONTROL-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display or set the nameservers for DNS.	read write	ilomCtrlDNSNameServers
Display or set the domains or search path for DNS lookups.	read write	ilomCtrlDNSSearchPath
Display or set whether the DNS configuration is provided by the DHCP server.	read write	ilomCtrlDNSdhcpAutoDns
Display or set the timeout(default 5 seconds).	read write	ilomCtrlDNSTimeout
Display or set the number of retries (default 1).	read write	ilomCtrlDNSRetries

Related Information

Understanding the `SUN-PLATFORM-MIB` MIB OIDs

This topic provides tables of the hardware, sensor, alarms, and watchdog object identifiers.

Tables of object identifiers are provided in these topics:

Related Information

SUN-PLATFORM-MIB MIB OID Command Syntax

When you use the object identifiers in a command syntax, you must include the SUN-PLATFORM-MIB MIB name. For example, using the V2c protocol:

```
$ command -v2c -c public mc_IP SUN-PLATFORM-MIB::object_id argument
```

Or using the V3 protocol:

```
$ command -v3 -u snmp_user -l security_level -a authentication_protocol -A authentication_password
-x DES -X privacy_password mc_IP SUN-PLATFORM-MIB::object_id argument
```

where:

- *command* is the command of the SNMP client that performs the action.
- *snmp_user* is a configured user of the SNMP services.
- *security_level* is:
    - authNoPriv – There is authentication, but no privacy.
    - authPriv – There is authentication and privacy.
- *authentication_protocol* is either MD5 or SHA
- *authentication_password* is the *snmp_user*'s authentication password.
- *privacy_password* is the *snmp_user*'s privacy password.
- *mc_IP* is the IP address of the management controller.
- *object_id* is the object identifier, as listed in the right column of the tables.
- *argument* is a combination of options and variables that support the object identifier.

For example, to securely display a table of gateway components addressable by the management controller with IP address of 123.45.67.89:

```
$ snmpwalk -v3 -u usersnmp -l authPriv -a MD5 -A authpass -x DES -X privpass
123.45.67.89 SUN-PLATFORM-MIB::sunPlatEquipmentTable
```

Related Information

- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175
- "SUN-FABRIC-MIB MIB OID Command Syntax" on page 190
- "SUN-ILOM-CONTROL-MIB MIB OID Command Syntax" on page 204
- "ENTITY-MIB MIB OID Command Syntax" on page 221

Understanding MIB Physical OIDs

Sensor and hardware information object identifiers are provided in these tables:

- "High-Level OIDs" on page 215
- "NIM Equipment Table OIDs" on page 216
- "Physical Class Extension Table OIDs" on page 217
- "Sun Platform Sensor Table OIDs" on page 217
- "Sun Platform Numeric Sensor Table OIDs" on page 218
- "Discrete Sensor Table OIDs" on page 219
- "Sun Platform Fan Table OIDs" on page 219
- "Sun Platform Alarm Table OIDs" on page 220

Related Information

- "OID Tables Overview" on page 173
- "Understanding the SUN-DCS-IB-MIB MIB OIDs" on page 174
- "Understanding the SUN-HW-TRAP-MIB MIB OIDs" on page 198
- "Understanding the SUN-ILOM-CONTROL-MIB MIB OIDs" on page 203
- "Understanding the ENTITY-MIB MIB OIDs" on page 221

High-Level OIDs

This table lists the high level object identifier supported by the SUN-PLATFORM-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display the time at which the agent was last started.	read	sunPlatStartTime

Related Information

- "OID Tables Overview" on page 173
- "SUN-PLATFORM-MIB MIB OID Command Syntax" on page 214

NIM Equipment Table OIDs

This table lists NIM equipment object identifiers supported by the SUN-PLATFORM-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display the NIM equipment table.	read	sunPlatEquipmentTable
Display information about a particular piece of equipment within the network element of the type specified by entPhysicalClass.	read	sunPlatEquipmentEntry
Display the administrative state of the managed object.	read	sunPlatEquipmentAdministrativeState
Display the operational state of the managed object.	read	sunPlatEquipmentOperationalState
Display the alarm status of the managed object.	read	sunPlatEquipmentAlarmStatus
Display the unknown or known status of the managed object.	read	sunPlatEquipmentUnknownStatus
Display the specific or general location name of the component.	read	sunPlatEquipmentLocationName

Related Information

- "OID Tables Overview" on page 173
- "SUN-PLATFORM-MIB MIB OID Command Syntax" on page 214

Physical Class Extension Table OIDs

This table lists physical class extension object identifiers supported by the
`SUN-PLATFORM-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display a table of information about the physical class of the entity if it is classified as `other(1)`.	read	`sunPlatPhysicalTable`
Display information about a piece of equipment with an `entPhysicalClass` of `other(1)`.	read	`sunPlatPhysicalEntry`
Display possible physical classes where `entPhysicalClass` is specified as `other(1)`.	read	`sunPlatPhysicalClass`

Related Information

-
-

Sun Platform Sensor Table OIDs

This table lists sensor object identifiers supported by the `SUN-PLATFORM-MIB` MIB.

Description of Task or Activity	Action	MIB OID
Display a table of attributes common to all sensors.	read	`sunPlatSensorTable`
Display information about a piece of equipment with an `entPhysicalClass` of `sensor(8)`.	read	`sunPlatSensorEntry`
Display possible sensor classes where `entPhysicalClass` is specified as `sensor(8)`.	read	`sunPlatSensorClass`
Display possible sensor types where `entPhysicalClass` is specified as `sensor(8)`.	read	`sunPlatSensorType`
Display the update interval for the sensor measured in milliseconds.	read	`sunPlatSensorLatency`

Related Information

-
-

Sun Platform Numeric Sensor Table OIDs

This table lists numeric sensor object identifiers supported by the
SUN-PLATFORM-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display a table of attributes specific to numeric sensors.	read	sunPlatNumericSensorTable
Display information about a sensor with a sunPlatSensorClass value of numeric(2).	read	sunPlatNumericSensorEntry
Display the base unit of the value returned by the sensor. For example, volts.	read	sunPlatNumericSensorBaseUnits
Display the exponent of 10 to be multiplied by the base unit. For example, 2. So the multiplier is 10^2 or 100.	read	sunPlatNumericSensorExponent
Display the rate associated with the base units, if any. For example, per second.	read	sunPlatNumericSensorRateUnits
Display the current reading of the sensor.	read	sunPlatNumericSensorCurrent
Display the lowest expected value from the sensor.	read	sunPlatNumericSensorNormalMin
Display the highest expected value from the sensor.	read	sunPlatNumericSensorNormalMax
Display the accuracy of the sensor expressed as a +/-% value in units of 100ths of a percent.	read	sunPlatNumericSensorAccuracy
Display the lower threshold at which a noncritical condition occurs.	read	sunPlatNumericSensorLowerThresholdNonCritical
Display the upper threshold at which a noncritical condition occurs.	read	sunPlatNumericSensorUpperThresholdNonCritical
Display the lower threshold at which a critical condition occurs.	read	sunPlatNumericSensorLowerThresholdCritical
Display the upper threshold at which a critical condition occurs.	read	sunPlatNumericSensorUpperThresholdCritical

Description of Task or Activity	Action	MIB OID
Display the lower threshold at which a fatal condition occurs.	read	sunPlatNumericSensorLowerThresholdFatal
Display the upper threshold at which a fatal condition occurs.	read	sunPlatNumericSensorUpperThresholdFatal
Display the hysteresis around the thresholds.	read	sunPlatNumericSensorHysteresis
Display the thresholds of the sensor.	read	sunPlatNumericSensorEnabledThresholds

Related Information

- "OID Tables Overview" on page 173
- "SUN-PLATFORM-MIB MIB OID Command Syntax" on page 214

Discrete Sensor Table OIDs

This table lists discrete sensor object identifiers supported by the SUN-PLATFORM-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display the discrete sensor table.	read	sunPlatDiscreteSensorTable
Display information about a sensor with entPlatSensorClass value of discrete(3).	read	sunPlatDiscreteSensorEntry
Display the current reading of the sensor.	read	sunPlatDiscreteSensorCurrent

Related Information

- "OID Tables Overview" on page 173
- "SUN-PLATFORM-MIB MIB OID Command Syntax" on page 214

Sun Platform Fan Table OIDs

This table lists fan object identifiers supported by the SUN-PLATFORM-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display a table of attributes common to all fans and cooling devices.	read	sunPlatFanTable
Display information about a piece of equipment with an entPhysicalClass of fan(7).	read	sunPlatFanEntry
Display the class of the cooling device.	read	sunPlatFanClass

Related Information

- "OID Tables Overview" on page 173
- "SUN-PLATFORM-MIB MIB OID Command Syntax" on page 214

Sun Platform Alarm Table OIDs

This table lists alarm object identifiers supported by the SUN-PLATFORM-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display a table with indications of alarm states.	read	sunPlatAlarmTable
Display information about a piece of equipment with an sunPlatPhysicalClass of alarm(2).	read	sunPlatAlarmEntry
Display the type of alarm.	read	sunPlatAlarmType
Display the state of the alarm.	read	sunPlatAlarmState
Display the urgency of the alarm.	read	sunPlatAlarmUrgency

Related Information

- "OID Tables Overview" on page 173
- "SUN-PLATFORM-MIB MIB OID Command Syntax" on page 214

Understanding the `ENTITY-MIB` MIB OIDs

This topic provides tables of physical and logical entity object identifiers of the Sun Network QDR InfiniBand Gateway Switch from Oracle.

Topics include:

- "ENTITY-MIB MIB OID Command Syntax" on page 221
- "Physical Entity Table OIDs" on page 222

Related Information

- "OID Tables Overview" on page 173
- "Understanding the SUN-DCS-IB-MIB MIB OIDs" on page 174
- "Understanding the SUN-FABRIC-MIB MIB OIDs" on page 189
- "Understanding the SUN-HW-TRAP-MIB MIB OIDs" on page 198
- "Understanding the SUN-ILOM-CONTROL-MIB MIB OIDs" on page 203
- "Understanding the SUN-PLATFORM-MIB MIB OIDs" on page 213

`ENTITY-MIB` MIB OID Command Syntax

When you use the object identifiers in a command syntax, you must include the `ENTITY-MIB` MIB name. For example, using the V2c protocol:

```
$ command -v2c -c public mc_IP ENTITY-MIB::object_id argument
```

Or using the V3 protocol:

```
$ command -v3 -u snmp_user -l security_level -a authentication_protocol -A authentication_password
-x DES -X privacy_password mc_IP ENTITY-MIB::object_id argument
```

where:

- *command* is the command of the SNMP client that performs the action.
- *snmp_user* is a configured user of the SNMP services.
- *security_level* is:
  - `authNoPriv` – There is authentication, but no privacy.

- **authPriv** – There is authentication and privacy.

- *authentication_protocol* is either MD5 or SHA

- *authentication_password* is the *snmp_user*'s authentication password.

- *privacy_password* is the *snmp_user*'s privacy password.

- *mc_IP* is the IP address of the management controller.

- *object_id* is the object identifier, as listed in the right column of the tables.

- *argument* is a combination of options and variables that support the object identifier.

For example, to securely display the description of physical entity 4 on the management controller with IP address of 123.45.67.89:

```
$ snmpget -v3 -u usersnmp -l authPriv -a MD5 -A authpass -x DES -X privpass
123.45.67.89 ENTITY-MIB::entPhysicalDescr.4
```

Related Information

- "SUN-DCS-IB-MIB MIB OID Command Syntax" on page 175
- "SUN-FABRIC-MIB MIB OID Command Syntax" on page 190
- "SUN-ILOM-CONTROL-MIB MIB OID Command Syntax" on page 204
- "SUN-PLATFORM-MIB MIB OID Command Syntax" on page 214

Physical Entity Table OIDs

This table lists physical entity object identifiers supported by the ENTITY-MIB MIB.

Description of Task or Activity	Action	MIB OID
Display a table of information about the physical entities.	read	entPhysicalTable
Display information about a particular physical entity.	read	entPhysicalEntry
Display the index of the physical entity.	read	entPhysicalIndex
Display the description of the physical entity.	read	entPhysicalDescr
Display the vendor-specific hardware type of the physical entity.	read	entPhysicalVendorType
Display the value of entPhysicalIndex for the physical entity which is the container for this physical entity.	read	entPhysicalContainedIn
Display the general hardware type of the physical entity.	read	entPhysicalClass

Description of Task or Activity	Action	MIB OID
Display an indication of the relative position of the child component among all its sibling components.	read	entPhysicalParentRelPos
Display the name of the physical entity.	read	entPhysicalName
Display the vendor-specific hardware revision string of the physical entity.	read	entPhysicalHardwareRev
Display the vendor-specific firmware revision string of the physical entity.	read	entPhysicalFirmwareRev
Display the vendor-specific software revision string of the physical entity.	read	entPhysicalSoftwareRev
Display the vendor-specific serial number string for the physical entity.	read	entPhysicalSerialNum
Display the name of the manufacturer of the physical component.	read	entPhysicalMfgName
Display the vendor-specific model name string associated with the physical component.	read	entPhysicalModelName
Display a handle or alias for the physical entity as specified by the network manager.	read	entPhysicalAlias
Display the nonvolatile asset tracking identifier for the physical entity as specified by a network manager.	read	entPhysicalAssetID
Display whether or not the physical entity is considered a field replaceable unit by the vendor.	read	entPhysicalIsFRU

Related Information

- "OID Tables Overview" on page 173
- "ENTITY-MIB MIB OID Command Syntax" on page 221

Index