Oracle® Server CLI Tools User's Guide

Part No: E25306–07
October 2013

ORACLE®

140528@25097

Contents

Using This Documentation

This section describes product information, documentation and feedback, and a document change history.

- "Documentation and Feedback" on page 7
- "About This Documentation" on page 7
- "Change History" on page 8

Documentation and Feedback

The following documentation is available related to the Oracle Hardware Management Pack.

Documentation	Link
All Oracle products	http://www.oracle.com/documentation
Oracle Hardware Management Pack	http://www.oracle.com/goto/OHMP/docs
Oracle ILOM	http://www.oracle.com/goto/ILOM/docs

Provide feedback on this documentation at:

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

About This Documentation

This documentation is available in both PDF and HTML and relates to software version 2.2.x. If there are any differences between software versions, they are noted. The information is presented in topic-based format (similar to online help) and therefore does not include chapters, appendixes, or section numbering.

You can get a PDF that includes all information about a particular topic subject (such as hardware installation or product notes) by clicking the PDF button in the upper left corner of the page.

Change History

The following changes have been made to the document..

- September 2010, initial publication.
- July 2011, updated document URLs.
- September 2011, updated to match software version 2.2. Changes include adding sections for hwmgmtcli and zoningcli, update of features for raidconfig, ilomconfig and fwupdate.
- November 2011, updated to integrate information related to Oracle Solaris OS 11. Fixed missing commands from ilomconfig and updated for various CRs.
- March 2012, updated functionality for fwupdate, ilomconfig, raidconfig. Added new tool, ubiosconfig.
- April 2012, fixed issue in ubiosconfig sections, improved information about raidconfig restore config and clear config subcommands.
- February 2013, updated to match software version 2.2.5.
- April 2013, updated to match software 2.2.6.
- July 2013, updated to match software 2.2.7.
- October 2013, updated to match software 2.2.8.

Oracle Server CLI Tools Overview

Oracle Server CLI Tools is part of Oracle Hardware Management Pack. Hardware Management Pack is a delivery mechanism for the operating system (OS) native tools and agents required for configuring and managing your server hardware.

The Hardware Management Pack download package includes the Oracle Hardware Management Pack Installer, which is a cross platform installer for the Hardware Management Pack components. For more information on installing Hardware Management Pack components, refer to the *Oracle Hardware Management Pack Installation Guide*.

Oracle Server CLI Tools consists of the following software:

Tool	Description	Link
`biosconfig`	Enables you to configure your server's BIOS CMOS settings and host boot order.	"Using the `biosconfig` Tool" on page 15
`ubiosconfig`	Enables you to import and export your server's UEFI BIOS settings to an XML file.	"Using the `ubiosconfig` Tool" on page 41
`fwupdate`	Enables you to update, query, and validate the firmware for Oracle server devices.	"Using the `fwupdate` Tool" on page 45
`raidconfig`	Enables you to configure RAID volumes.	"Using the `raidconfig` Tool" on page 63
`ilomconfig`	Enables you to manipulate Oracle ILOM configurations.	"Using the `ilomconfig` Tool" on page 87
`hwmgmtcli`	Enables you to get information from the Oracle ILOM service processor.	"Using the `hwmgmtcli` Tool" on page 107
`zoningcli`	Tool for Oracle SPARC T3-1 servers running Oracle Solaris OS. Enables you to configure systems that have the 16 disk backplane (SAS-2 expander) into two separated zones.	"Using the `zoningcli` Tool" on page 111

For more information on other Hardware Management Pack features, see *Oracle Hardware Management Pack Installation Guide* and *Oracle Server Management Agents User's Guide*.

For late-breaking issues and information about the CLI Tools, refer to the *Oracle Hardware Management Pack 2.2.x Release Notes*.

See also:

- "CLI Tools Command Syntax and Conventions" on page 11

CLI Tools Command Syntax and Conventions

This section describes the common syntax used by all CLI Tools.

- "CLI Tools Command Syntax" on page 11
- "CLI Tools Device-Naming Convention" on page 12

CLI Tools Command Syntax

Most CLI tools commands conform to one of the following two command syntax formats:

- *command* [*option*]
- *command subcommand target* [*option*]

Note – The `biosconfig` tool does not conform to the above syntax. See "Using the `biosconfig` Tool" on page 15 for more information.

The following table describes the command fields:

Command Field	Description	Examples
command	The action that you want to perform. Identifies that CLI tool that you are using. Consists of lower-case letters only.	`biosconfig`, `fwupdate`, `raidconfig`, `ilomconfig`
subcommand	Further defines the task to be performed by the *command*. Generally used as verbs. Consists of lower-case letters, hyphens, or the underscore character. The subcommand is not required when the `--version` or `--help` options is used immediately following the command.	`list`, `update`, `reset`, `expander-boot-record`
target	Describes the object or target that is being acted upon by the subcommand. Application specific.	`all`, `disk`, `expander`, `bridge`, `controller`, `user`, `snmp-community`

Command Field	Description	Examples
option	Modifies the command or subcommand and can be optional or mandatory depending on the command or subcommand. There are long and short options that have identical functionality and are provided for ease of use: Short-option is a hyphen followed by a single letter. Long-option is two hyphens followed by a string.	`-n` or *--device_name* `-f` or *--filename* `-r` or *--reset*

The following options apply to all CLI Tools commands:

Short Option	Long Option	Description
`-?`	`--help`	Help—Displays help information.
`-V`	`--version`	Version—Displays the tool version.
`-q`	`--quiet`	Quiet—Suppresses informational message output and returns only error codes.
`-y`	`--yes`	Yes—Confirms operation. Does not prompt user for confirmation on the operation when running.

When using a command option and its corresponding value or device name, you can use an equal sign (=) or a space as shown in the following examples:

- Using a command with spaces:

  ```
  raidconfig create raid -c c2 --raid-level 1 --number-disks 2
  ```

- Using a command with equal signs (=):

  ```
  raidconfig create raid -c=c2 --raid-level=1 --number-disks=2
  ```

See also:

- "CLI Tools Device-Naming Convention" on page 12

CLI Tools Device-Naming Convention

User-friendly, fully qualified device names are used with the CLI Tools commands. The single characters represent all of the nodes that make up the device as follows:

Character	Description
c	Controller — Using a unique logical ID.

Character	Description
r	RAID Volume (logical disk) — The logical ID name of the volume or disk.
d	Disk — The physical disk logical ID name.
x	Expander — The unique expander logical ID name.
j	Chassis — The unique chassis logical ID name.

All integers used to represent the device are 0 based. Disks are represented by logical ID names assigned by the tool at initialization. The disks are sorted by expander and slot ID to create unique numerical identifiers.

The following are examples of device names:

- `c1` — Controller 1
- `c1d2` — Disk with a logical ID 2 on controller 1
- `c2r1` — RAID 1 on controller 2

Multiple devices can be listed together in a comma-separated list, for example: `dev1,dev2,dev3`.

The following is an example from `RAIDconfig` when you create a RAID volume with three disks:

```
./raidconfig create --disks c1d2,c1d4,c1d5 --level 1
```

The following example shows an implementation of the disk-naming scheme.

```
ID      Brand       Model            Chassis  Slot    Type   Media  Size (GB) Firmware
Revision
-------------------------------------------------------------------------------------
c1d0    SEAGATE     ST373455SSUN72G     0       0      sas    HDD     73       0791
c1d1    SEAGATE     ST35000N            0       1      sata   HDD     500      3AZQ
c1d2    SEAGATE     ST373455SSUN72G     0       2      sas    HDD     73       0B92
c1d3    SEAGATE     ST373455SSUN72G     0       3      sas    HDD     73       0B92
c1d4    SEAGATE     ST35000N            0       4      sata   HDD     500      3AZQ
c1d5    SEAGATE     ST35000N            0       5      sata   HDD     500      3AZQ
c1d6    SEAGATE     ST35000N            0       6      sata   HDD     500      3AZQ
c1d7    SEAGATE     ST373455SSUN72G     0       7      sas    HDD     73       0B92
c1d8    SEAGATE     ST373455SSUN72G     0       8      sas    HDD     73       0B92
c1d9    SEAGATE     ST373455SSUN72G     0       9      sas    HDD     73       0B92
c1d10   SEAGATE     ST35000N            0       10     sata   HDD     500      3AZQ
c1d11   SEAGATE     ST373455SSUN72G     0       11     sas    HDD     73       0B92
c1d12   SEAGATE     ST373455SSUN72G     0       12     sas    HDD     73       0B92
c1d13   SEAGATE     ST373455SSUN72G     0       13     sas    HDD     73       0B92
c1d14   SEAGATE     ST373455SSUN72G     0       14     sas    HDD     73       0B92
c1d15   SEAGATE     ST373455SSUN72G     0       15     sas    HDD     73       0B92
c1d16   SEAGATE     ST373455SSUN72G     0       16     sas    HDD     73       0B92
c1d17   SEAGATE     ST373455SSUN72G     0       17     sas    HDD     73       0B92
c1d18   SEAGATE     ST373455SSUN72G     0       18     sas    HDD     73       0B92
c1d19   SEAGATE     ST373455SSUN72G     0       19     sas    HDD     73       0B92
```

```
c1d20   SEAGATE     ST35000N            0       20      sata    HDD     500     3AZQ
c1d21   SEAGATE     ST35000N            0       21      sata    HDD     500     3AZQ
c1d22   SEAGATE     ST35000N            0       22      sata    HDD     500     3AZQ
c1d23   SEAGATE     ST35000N            0       23      sata    HDD     500     3AZQ
c1d24   SEAGATE     ST373455SSUN72G     1       0       sas     HDD     73      0791
c1d25   SEAGATE     ST35000N            1       1       sata    HDD     500     3AZQ
c1d26   SEAGATE     ST373455SSUN72G     1       3       sas     HDD     73      0791
c1d27   SEAGATE     ST35000N            1       4       sata    HDD     500     3AZQ
c1d28   SEAGATE     ST373455SSUN72G     1       5       sas     HDD     73      0791
c1d29   SEAGATE     ST35000N            1       6       sata    HDD     500     3AZQ
c1d30   SEAGATE     ST373455SSUN72G     1       7       sas     HDD     73      0791
c1d31   SEAGATE     ST373455SSUN72G     1       8       sas     HDD     73      0791
c1d32   SEAGATE     ST373455SSUN72G     1       9       sas     HDD     73      0791
c1d33   SEAGATE     ST373455SSUN72G     1       10      sas     HDD     73      0791
c1d34   SEAGATE     ST373455SSUN72G     1       11      sas     HDD     73      0791
c1d35   SEAGATE     ST35000N            1       12      sata    HDD     500     3AZQ
c1d36   SEAGATE     ST373455SSUN72G     1       13      sas     HDD     73      0791
c1d37   SEAGATE     ST373455SSUN72G     1       14      sas     HDD     73      0791
c1d38   SEAGATE     ST35000N            1       15      sata    HDD     500     3AZQ
c1d39   SEAGATE     ST373455SSUN72G     1       16      sas     HDD     73      0791
c1d40   SEAGATE     ST373455SSUN72G     1       17      sas     HDD     73      0791
c1d41   SEAGATE     ST35000N            1       18      sata    HDD     500     3AZQ
c1d42   SEAGATE     ST35000N            1       19      sata    HDD     500     3AZQ
c1d43   SEAGATE     ST35000N            1       20      sata    HDD     500     3AZQ
c1d44   SEAGATE     ST35000N            1       21      sata    HDD     500     3AZQ
c1d45   SEAGATE     ST35000N            1       22      sata    HDD     500     3AZQ
c1d46   SEAGATE     ST35000N            1       23      sata    HDD     500     3AZQ
```

See also:

- "CLI Tools Command Syntax" on page 11

Using the `biosconfig` Tool

`biosconfig` is an application that runs on your server's OS and configures the BIOS CMOS settings, host boot order, and some service processor settings.

Note – The `biosconfig` tool is available on supported Oracle x86 servers. Servers that support UEFI BIOS must use the `ubiosconfig` tool. See "Using the `ubiosconfig` Tool" on page 41.

For a list of the tools and the systems that support them, refer to:

`http://www.oracle.com/goto/system-management`

`biosconfig` allows you to manipulate BIOS configurations from the OS command line. The configuration files and command-line interfaces are compatible with the Oracle Solaris, Windows, and Linux based OS versions of `biosconfig`.

This section covers the following information:

`biosconfig` Overview

This section covers the following topics:

biosconfig Requirements

- You must run biosconfig as root (Linux, Oracle Solaris OS) or Administrator (Windows) because it needs to use drivers that are in read- and write-protected physical address space.
- Close all other applications and quiesce your system before running biosconfig.
- Linux versions of biosconfig depend on access to /dev/nvram to guarantee serialized access to the CMOS.

  Red Hat Enterprise Linux 4 distributions do not include this device by default. Red Hat Enterprise Linux 5 and SUSE Linux Enterprise Server distributions do include the device by default.

  For you to use /dev/nvram, the driver needs to be compiled into the kernel (or loaded as a module), and /dev/nvram must exist (root can create it using mknod /dev/nvram c 10 144).
- The Sun System Management driver is required for biosconfig to run on Windows systems. For information on biosconfig for Windows, see: "biosconfig for Windows" on page 19.

  .

See also:

- "biosconfig Terminology" on page 16
- "biosconfig Command Overview" on page 18

biosconfig Terminology

Term	Definition
BIOS	This is the software that initializes the computer hardware and then boots the operating system.
CMOS	In this context, it is the 128 or 256 bytes of battery backed-up RAM that stores the BIOS configuration when the system is powered off.
IPMI	A standard interface used to manage servers. For more information, go to: http://www.intel.com/design/servers/ipmi.
ipmitool	An open-source tool used to manage a system. ipmitool is provided with the software download for each Oracle server. You can find documentation at: http://ipmitool.sourceforge.net/manpage.html.
NVRAM	In this context, it is the portion of the BIOS ROM that holds the BIOS's boot information.

See also:

- "biosconfig Terminology" on page 16
- "biosconfig Requirements" on page 16
- "biosconfig Command Overview" on page 18

Device Terminology Used by biosconfig

The following notes explain how biosconfig describes devices:

- Floppy refers to whatever the BIOS considers a removable device.

  For example, this could be a USB flash drive.
- A USB flash drive bigger than 512 MB is referred to as a disk.
- A USB/CD-ROM is classed as a CD and not a removable device.
- PXE means a bootable network device.

  For example, this might be an Ethernet controller or an InfiniBand interface that has boot support in its expansion ROM.

See also:

- "Configuring the Device Boot Order" on page 30

Device Name Examples

The device name examples listed in the following table are used in XML file output in this chapter.

Output Text	Description of Hardware
SATA:3M-MRVLRD 200254-01SUN24G 0801	Flash mini-DIMM SATA (which is disk-like)
USB:Port1:Memorex DVD+-RAM 510L	USB DVD drive (which is CD-like)
USB:Port0:SanDisk Cruzer Contour	1 GB USB flash drive (which is disk-like)
IB:Slot2.F0:PXE:MLNX HCA IB 1.9.972	InfiniBand PXE (which is network-like)
PXE:IBA GE Slot 00C8 v1324	On-board GigEthernet NIC (which is a network interface)

Editing XML Files

biosconfig enables you to configure settings across multiple similar servers using a common XML configuration file. However, if the configuration that is being modified includes a peripheral or component that is not on both systems, then you need to customize the XML file.

The biosconfig command can be used to get current configuration settings or set configuration settings. When used to get configuration settings, biosconfig generates XML output showing the configuration. When used to set configuration settings, biosconfig reads XML input describing the configuration settings.

Caution – Do not use biosconfig to change BIOS settings that are not visible in the normal BIOS setup menu.

To use biosconfig, you must have a working knowledge of XML file editing. The process of editing the BIOS includes using biosconfig to do the following tasks:

1. Run biosconfig -get command.

   If an XML file name is specified with the -get option, the BIOS configuration will be saved to the XML file. If an XML file is not specified, the output is written to the terminal.
2. Review the XML file and modify it, as required.

   You can modify the XML files in a editor of your choice, such as vi.
3. Run biosconfig -set *filename*.xml to implement the changes.

   You can use the same XML file to modify multiple systems.

biosconfig Command Overview

The following table lists the available biosconfig options and their descriptions.

Command	Description
-get_version	Get version of this tool.
-get_boot_order	Get the boot devices list.
-set_boot_order	Set the boot devices list.
-set_boot_override	Set the first boot device for the next boot.
-get_bios_settings	Get setup configuration from BIOS.
-set_bios_settings	Get setup configuration to BIOS ROM.
-get_CMOS_dump	Get 256 bytes CMOS of set up data from BIOS.
-set_CMOS_dump	Set 256 bytes of CMOS set up data to BIOS.

The following table lists examples of how the -get and -set command options affect input and output.

Command	Description
# biosconfig -get_version	Outputs to screen.
# biosconfig -get_version *file*.xml	Outputs to *file*.xml.
# biosconfig -get_version> *file*.xml	Outputs to *file*.xml.
# biosconfig -get_version | *some-command*	Pipes the output to another command.
# biosconfig -set_bios_settings	Takes input from standard in.
# biosconfig -set_bios_settings *file*.xml	Takes input from *file*.xml.
# biosconfig -set_bios_settings < *file*.xml	Takes input from *file*.xml.

When a command fails, it returns one of the failure codes listed in "biosconfig Error Codes" on page 121.

Note – In the output examples in this chapter, all white space outside the XML elements, such as indentation, is optional. For example, see output in "How to Make a Persistent Change to Boot Order" on page 32.

See also:

- "biosconfig Requirements" on page 16
- "biosconfig Error Codes" on page 121

biosconfig for Oracle Solaris OS

The BIOS configuration tool for Oracle Solaris OS (biosconfig) is a utility that runs on the host system and configures the host's BIOS CMOS settings, host boot order, and some service processor settings.

Oracle Solaris OS biosconfig consists of an Oracle Solaris OS biosdrv driver and the biosconfig application.

biosconfig for Windows

With biosconfig version 2.2.1 and above for Windows, biosconfig.exe is run only in its installation directory so it can access its low-level management driver. The Sun System Management driver is included as part of the Hardware Management Pack download.

The Sun System Management driver is required in order to run biosconfig on Windows Server 2008 SP2 64–bit or Windows Server 2008 R2 systems. This driver is not required for Windows 2008 32–bit.

The Sun System Management driver is not needed for other CLI Tools. Uninstall the driver to free system resources if biosconfig is not going be used. When biosconfig is not installed, the driver must be uninstalled manually.

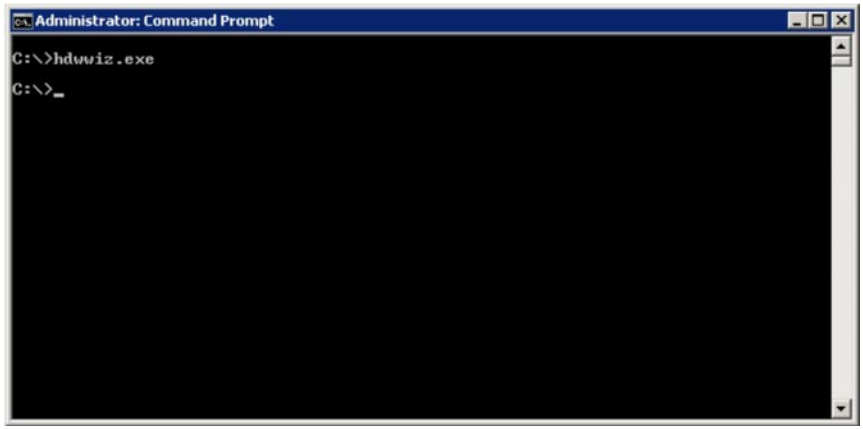
Note – On some systems, when using the Sun System Management Driver, biosconfig might take several minutes to complete some operations.

This section covers the following procedures:

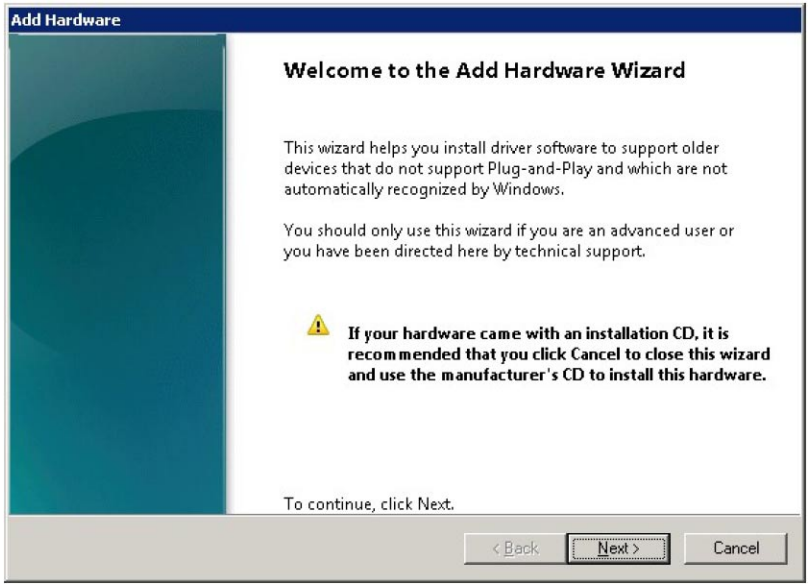
- "How to Install the biosconfig Sun System Management Driver on Windows 2008 R2 and Windows 2008 64–bit" on page 21
- "How to Uninstall the biosconfig Sun System Management Driver on Windows 2008 R2 and Windows 2008 64–bit" on page 26

▼ How to Install the biosconfig Sun System Management Driver on Windows 2008 R2 and Windows 2008 64–bit

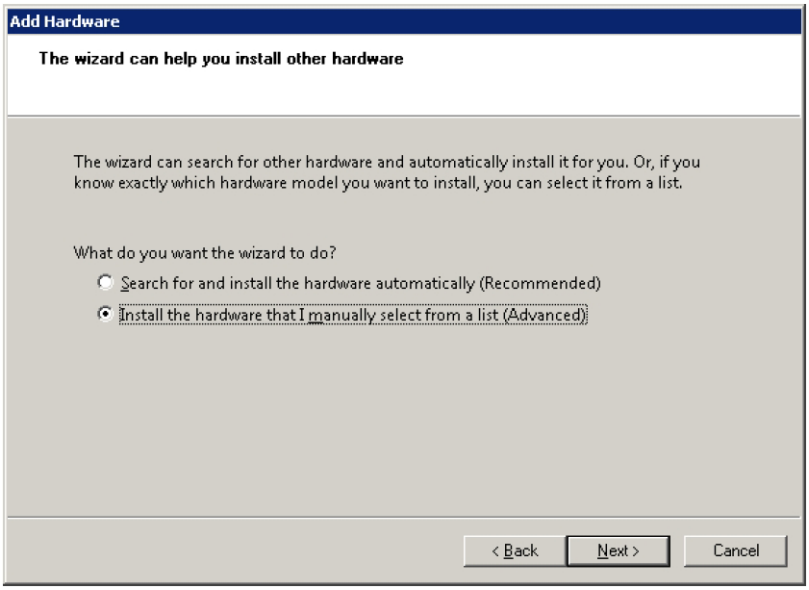
1 In an Administrator terminal window (cmd.exe), run the Hardware Wizard executable file, hdwiz.exe.



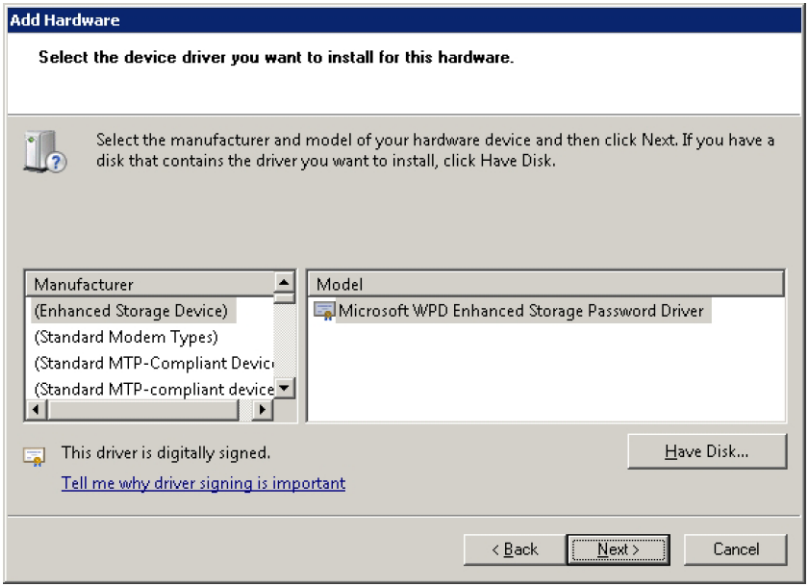
2 Read the Welcome screen and click Next.



3 **To install the hardware device manually, select Install the hardware that I manually select from a list.**



4 **Click Have Disk.**



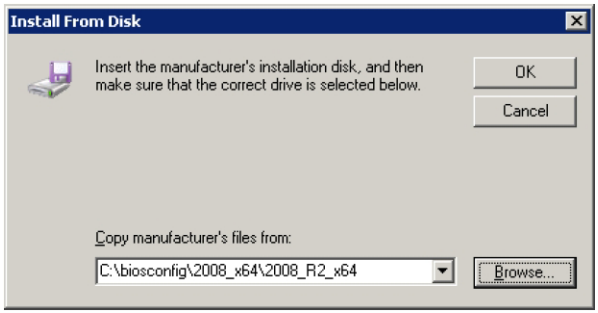
5 **Choose the driver. There are two common paths for the Sun System Management Driver.**

- **Navigate to the path where you extracted Hardware Management Pack.**
- **Open the `SOFTWARE/drivers` directory and navigate to the appropriate architecture (32 or 64 bit) and click OK.**
- **Navigate to the path where you installed `biosconfig`.**

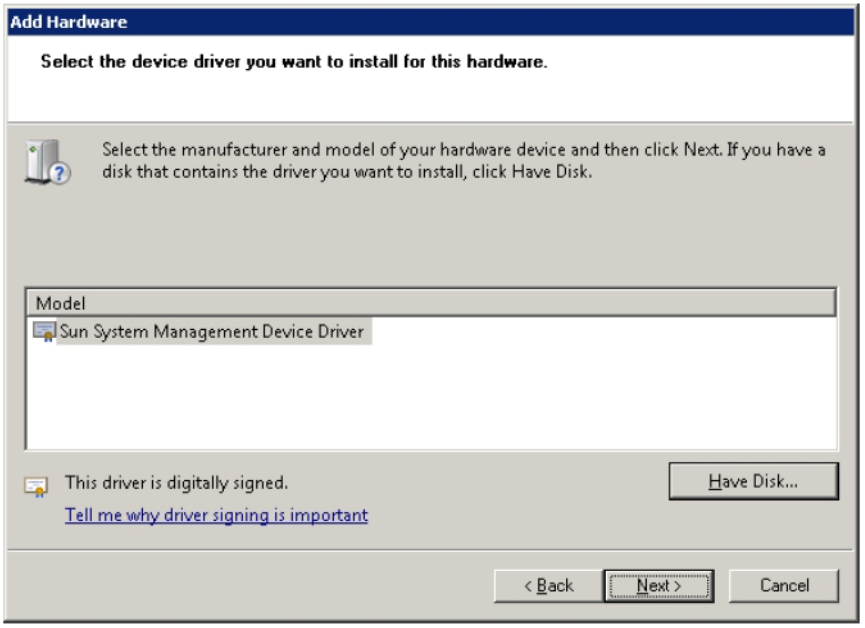
  The location of the driver depends on which version of Oracle HMP you have.

  - *For version 2.2.7 or later*: The driver is located in the 2008_x64 directory.
  - *For version 2.2.6 and earlier*: The `2008_R2_x64`directory is for Windows Server 2008 R2 64–bit, and the `2008_SP2_x64` directory is for Windows Server 2008 SP2 64–bit.

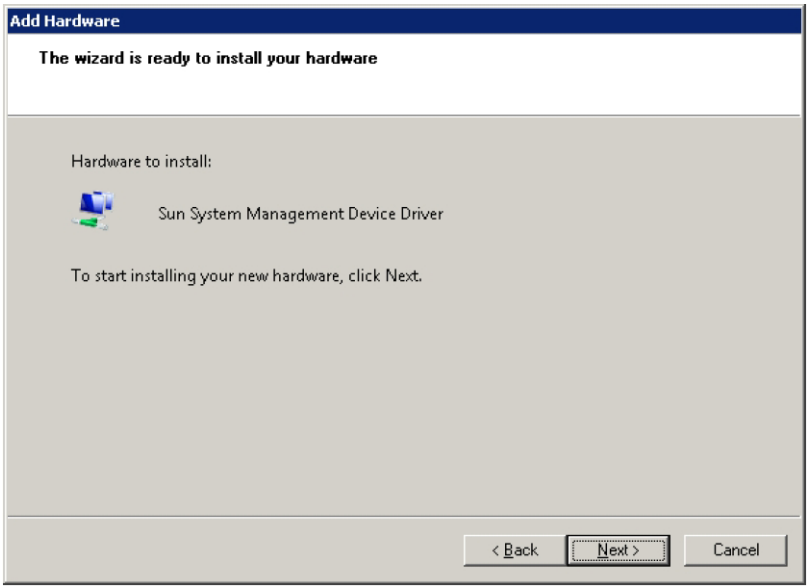
  For example:



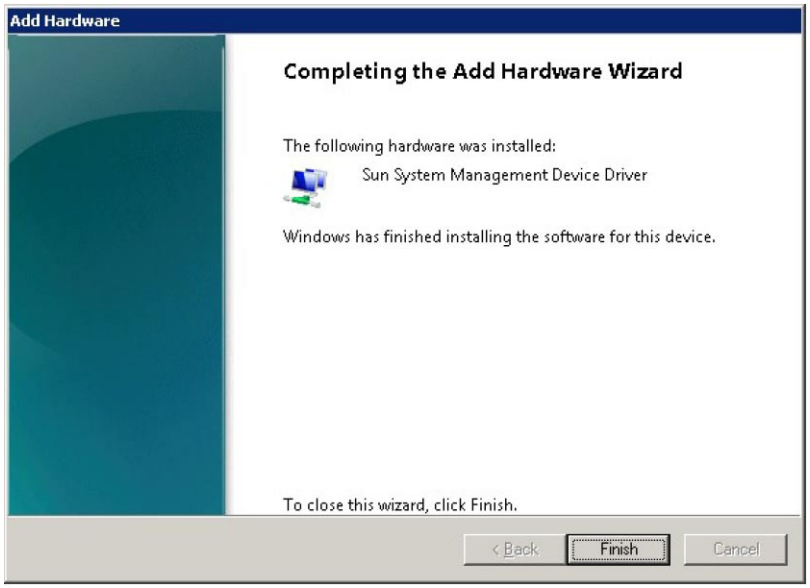
6 **Select the Sun System Manager Device Driver, and click Next.**



7 **Click Next to start installing.**



8 **When the installation is complete, click Finish.**



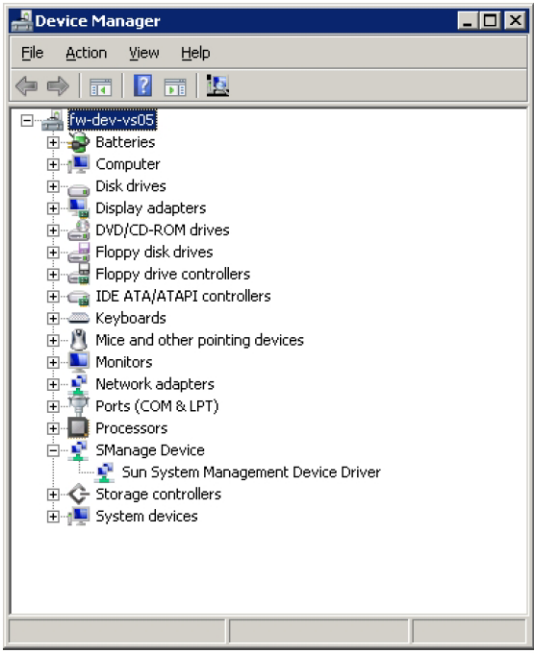
Next Steps "biosconfig Command Overview" on page 18

▼ How to Uninstall the biosconfig Sun System Management Driver on Windows 2008 R2 and Windows 2008 64–bit

1 Open the Windows Control Panel and select Hardware.

2 In the hardware window, select Device Manager.

3 **Right click on the Sun System Management Device Driver and select Uninstall.**



4 **Check Delete the driver software for this device and click OK.**

The driver is removed from the system.

Viewing biosconfig Command Options and Version Information

The following procedures are included in this section:

- "How to View biosconfig Command Options" on page 28
- "How to View biosconfig Version Information" on page 28

▼ How to View biosconfig Command Options

● **To view the help output, type the biosconfig command without arguments.**

For example:

```
# biosconfig
Copyright (C) SUN Microsystems 2009.
BIOSconfig Utility Version 2.2.5
Build Date: Jan 11 2010
Build Time: 01:22:05

BIOSconfig Specification Version 2.4

Usage: biosconfig [-v] option [filename]
Example: biosconfig -get_version output.xml

[-v] Verbose on. Only valid if a xml input/output filename is provided
[Filename] Name of the XML output (or input) file for get (or set)
command (optional).
get commands will output to the console if the filename
is not provided
set commands will get input from the console if the filename
is not provided

Available options (Required):
-get_version Get version of this tool
-get_boot_order Get the BOOT Devices list
-set_boot_order Set the BOOT Devices list
-get_bios_settings Get setup configuration from BIOS
-set_bios_settings Set setup configuration to BIOS ROM
-get_CMOS_dump Get 256 bytes CMOS setup data from BIOS
-set_CMOS_dump Set 256 bytes of CMOS setup data to BIOS
```

See Also

- "How to View biosconfig Version Information" on page 28

▼ How to View biosconfig Version Information

1 **To view version information, type the following command:**

biosconfig -get_version *filename*.**xml**

For example:

```
# biosconfig -get_version ver.xml

Copyright (C) SUN Microsystems 2009.
BIOSconfig Utility Version 2.1
Build Date: Jul 16 2009
Build Time: 15:55:12

BIOSconfig Specification Version 2.4

Success
```

2 **View the created *filename*.xml file.**

The following is an example of the version information in an .xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<BIOSCONFIG>
    <BIOSCONFIG_VERSION>2.1</BIOSCONFIG_VERSION>
    <SPEC_VERSION>2.4</SPEC_VERSION>
    <SP_NETWORK_CONFIG>
        <DISCOVERY></DISCOVERY>
        <IP></IP>
        <NETMASK></NETMASK>
        <GATEWAY></GATEWAY>
    </SP_NETWORK_CONFIG>
    <PASSWORD_CONFIG>
        <PASSWORD></PASSWORD>
    </PASSWORD_CONFIG>
    <BOOT_ORDER_OVERRIDE>
        <HELP_STRING>FIRST=Choose one of: pxe, cdrom, disk,
floppy, bios, none</HELP_STRING>
<FIRST></FIRST>
        <HELP_STRING>CLEAR_CMOS=Choose Yes, No or leave it
empty, em.....</HELP_STRING>
        <CLEAR_CMOS></CLEAR_CMOS>
    </BOOT_ORDER_OVERRIDE>
    <BOOT_DEVICE_PRIORITY>
        <B0>
            <DEVICE_NAME></DEVICE_NAME>
            <PCI-B-D-F></PCI-B-D-F>
        </B0>
    </BOOT_DEVICE_PRIORITY>
</BIOSCONFIG>
```

See Also

- "How to View biosconfig Command Options" on page 28

Configuring the Device Boot Order

During BIOS power-on self-test (POST), BIOS scans the hardware and accumulates a list of bootable devices. That list is then ordered into categories and presented as a boot list, which is the ordered list of bootable devices on which a boot is attempted.

`biosconfig` enables you to configure the first device to boot at the next reboot or to configure the entire boot order. `biosconfig` does this by reading the boot-related tables that the BIOS stores in NVRAM, and then by manipulating the contents of CMOS where the boot order is stored.

This section covers the following topics:

Methods for Changing the Boot List

The boot list can be changed in any of the following ways:

- Change the order in BIOS setup utility.
- Reorder the categories using the IPMI bootflags that the SP offers to the compatible BIOS during POST. The default priority order for categories is CD/DVD, disk, removable, and network. See "Configuring Boot Order Using ipmitool " on page 116.
- Change the boot order using `biosconfig`. This manipulates the contents of CMOS and the BIOS boot block structures stored in NVRAM, which is a dedicated part of the BIOS ROM.

  This chapter contains instructions for changing the boot order using `biosconfig`.

Note – This boot list changes dynamically when devices such as disk drives, USB devices, and PCIe cards are installed and removed. The boot list also changes when javaConsole floppy and CD redirection is started and stopped.

See also:

- "Device Terminology Used by `biosconfig`" on page 17

▼ How to Set the First Boot Device for the Next Boot

This procedure shows how to set the first boot device for the next boot only. To change the boot device for successive boots, see "How to Make a Persistent Change to Boot Order" on page 32.

Here is an example of using the `-set_boot_override` command that specifies the first boot device as the PXE server on only the next boot:

1. **Create an XML file containing the current boot order of your system with the following command:**

   `biosconfig -get_boot_order` *filename*`.xml`

2. **Edit XML text so that the device that you want to boot first is between the <FIRST> tags.**

   The following is an example of the resulting XML file.

```
<?xml version="1.0" encoding="UTF-8"?>
<BIOSCONFIG>
   <BIOSCONFIG_VERSION>2.1</BIOSCONFIG_VERSION>
   <SPEC_VERSION>2.4</SPEC_VERSION>
   <SP_NETWORK_CONFIG>
      <DISCOVERY></DISCOVERY>
      <IP></IP>
      <NETMASK></NETMASK>
      <GATEWAY></GATEWAY>
   </SP_NETWORK_CONFIG>
   <PASSWORD_CONFIG>
      <PASSWORD></PASSWORD>
   </PASSWORD_CONFIG>
   <BOOT_ORDER_OVERRIDE>
      <HELP_STRING>FIRST=Choose one of: pxe, cdrom, disk,
floppy, bios, none</HELP_STRING>
      <FIRST>pxe</FIRST>
      <HELP_STRING>CLEAR_CMOS=Choose Yes, No or leave it
empty, .....</HELP_STRING>
      <CLEAR_CMOS></CLEAR_CMOS>
   </BOOT_ORDER_OVERRIDE>
   <BOOT_DEVICE_PRIORITY>
   <Boot_Device_01>
      <DEVICE_NAME>USB:Port1:Memorex DVD+-RAM 510L v1</DEVICE_NAME>
   </Boot_Device_01>
   <Boot_Device_02>
      <DEVICE_NAME>SATA:3M-MRVLRD 200254-01SUN24G 0801</DEVICE_NAME>
   </Boot_Device_02>
   <Boot_Device_03>
      <DEVICE_NAME>USB:Port0:SanDisk Cruzer Contour</DEVICE_NAME>
   </Boot_Device_03>
   <Boot_Device_04>
      <DEVICE_NAME>IB:Slot2.F0:PXE:MLNX HCA IB 1.9.972 (PCI 07:00.
</DEVICE_NAME>
      <PCI-B-D-F>07,00,00</PCI-B-D-F>
   </Boot_Device_04>
   <Boot_Device_05>
      <DEVICE_NAME>PXE:IBA GE Slot 00C8 v1324</DEVICE_NAME>
      <PCI-B-D-F>00,19,00</PCI-B-D-F>
   </Boot_Device_05>
   </BOOT_DEVICE_PRIORITY>
</BIOSCONFIG>
```

3. **Set the boot order with the following command:**

   `biosconfig -set_boot_override` *filename*`.xml`

See Also
- "How to Make a Persistent Change to Boot Order" on page 32
- "How to Change Boot Order Based on the PCI Bus, Device, or Function" on page 33

▼ How to Make a Persistent Change to Boot Order

To make a persistent change to the boot order, modify the order of devices in the BOOT_DEVICE_PRIORITY section of the XML file.

The following example shows an XML file from a Sun Blade X6275 server module (which has a built-in bootable InfiniBand interface) set to optimal defaults with a 1-GByte USB flash, a USB CD, and a dual Gig-Ethernet Express Module plugged in.

1 **Create an XML file containing the current boot order of your system with the following command:**

`biosconfig -get_boot_order` *filename***`.xml`**

The following is an example of the output of the XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<BIOSCONFIG>
   <BIOSCONFIG_VERSION>2.1</BIOSCONFIG_VERSION>
   <SPEC_VERSION>2.4</SPEC_VERSION>
   <SP_NETWORK_CONFIG>
      <DISCOVERY></DISCOVERY>
      <IP></IP>
      <NETMASK></NETMASK>
      <GATEWAY></GATEWAY>
   </SP_NETWORK_CONFIG>
   <PASSWORD_CONFIG>
      <PASSWORD></PASSWORD>
   </PASSWORD_CONFIG>
   <BOOT_ORDER_OVERRIDE>
      <HELP_STRING>FIRST=Choose one of: pxe, cdrom, disk,
floppy, bios, none</HELP_STRING>
      <FIRST></FIRST>
      <HELP_STRING>CLEAR_CMOS=Choose Yes, No or leave it
empty, .....</HELP_STRING>
      <CLEAR_CMOS></CLEAR_CMOS>
   </BOOT_ORDER_OVERRIDE>
   <BOOT_DEVICE_PRIORITY>
   <Boot_Device_01>
      <DEVICE_NAME>USB:Port1:Memorex DVD+-RAM 510L v1</DEVICE_NAME>
   </Boot_Device_01>
   <Boot_Device_02>
      <DEVICE_NAME>SATA:3M-MRVLRD 200254-01SUN24G 0801</DEVICE_NAME>
   </Boot_Device_02>
   <Boot_Device_03>
      <DEVICE_NAME>USB:Port0:SanDisk Cruzer Contour</DEVICE_NAME>
   </Boot_Device_03>
   <Boot_Device_04>
      <DEVICE_NAME>IB:Slot2.F0:PXE:MLNX HCA IB 1.9.972 (PCI 07:00.
</DEVICE_NAME>
      <PCI-B-D-F>07,00,00</PCI-B-D-F>
```

```
   </Boot_Device_04>
   <Boot_Device_05>
      <DEVICE_NAME>PXE:IBA GE Slot 00C8 v1324</DEVICE_NAME>
      <PCI-B-D-F>00,19,00</PCI-B-D-F>
   </Boot_Device_05>
   </BOOT_DEVICE_PRIORITY>
</BIOSCONFIG>
```

2 **Edit the device names displayed between the <DEVICE_NAME> tags so that the devices are listed in the desired boot order.**

3 **Set the boot order with the following command:**

`biosconfig -set_boot_order` *filename*.**`xml`**

See Also

- "How to Set the First Boot Device for the Next Boot" on page 30
- "How to Change Boot Order Based on the PCI Bus, Device, or Function" on page 33

▼ How to Change Boot Order Based on the PCI Bus, Device, or Function

The `biosconfig` command can alter the boot order based on the PCI bus, device, or function if the boot order list contains that information.

1 **Create an XML file containing the current boot order of your system with the following command:**

`biosconfig -get_boot_order` *filename*.**`xml`**

2 **Edit the devices listed between the <PCI-B-D-F> tags so that they are in the desired order.**

For example:

```
<BOOT_DEVICE_PRIORITY>
<Boot_Device_01>
<DEVICE_NAME>PXE:IBA GE Slot 00C8 v1324</DEVICE_NAME>
<PCI-B-D-F>00,19,00</PCI-B-D-F>
</Boot_Device_01>
<Boot_Device_02>
<DEVICE_NAME>IB:Slot2.F0:PXE:MLNX HCA IB 1.9.972 (PCI 07:00.
</DEVICE_NAME>
<PCI-B-D-F>07,00,00</PCI-B-D-F>
</Boot_Device_02>
<Boot_Device_03>
<DEVICE_NAME>USB:Port1:Memorex DVD+-RAM 510L v1</DEVICE_NAME>
</Boot_Device_03>
<Boot_Device_04>
<DEVICE_NAME>USB:Port0:SanDisk Cruzer Contour</DEVICE_NAME>
</Boot_Device_04>
<Boot_Device_05>
<DEVICE_NAME>SATA:3M-MRVLRD 200254-01SUN24G 0801</DEVICE_NAME>
```

```
</Boot_Device_05>
</BOOT_DEVICE_PRIORITY>
```

3 **Set the boot order with the following command:**

`biosconfig -set_boot_order` *filename*.**`xml`**

See Also
- "How to Set the First Boot Device for the Next Boot" on page 30
- "How to Make a Persistent Change to Boot Order" on page 32

Configuring the BIOS CMOS

The BIOS configuration information is stored in the CMOS memory in the host's chipset. Through the BIOS setup interface at BIOS POST, you can configure many of the CMOS settings. `biosconfig` is an alternative interface to modify these settings with a program on the host OS. `biosconfig` configures the BIOS CMOS settings using two methods:

- Copying and using a golden (known reliable) image
- Controlling each setting individually

This section covers the following topics:

- "How to Capture the BIOS CMOS Golden Image" on page 34
- "How to Apply the BIOS CMOS Golden Image" on page 36
- "Configuring Individual CMOS Settings" on page 36

▼ How to Capture the BIOS CMOS Golden Image

The BIOS configuration consists of the contents of the CMOS and the boot tables in the NVRAM. The command `biosconfig -get_CMOS_dump` captures the 256 bytes of CMOS, but it does not gather the boot table information from NVRAM.

So this command might not capture the boot-order information, unless the bootable I/O configurations for the source and destination machines are the same.

1 **To generate a golden (known reliable) CMOS image, use the BIOS Setup Utility to configure the BIOS settings.**

2 **To capture the 256 bytes of CMOS containing the configuration information, use the following command:**

`biosconfig -get_CMOS_dump` *filename*.**`xml`**

The following is example output for the command.

```
Copyright (C) SUN Microsystems 2009.
BIOSconfig Utility Version 2.1
Build Date: Jul 16 2009
```

```
Build Time: 15:55:12
BIOSconfig Specification Version 2.4
Success
```

The following is an example of the .xml file containing the CMOS configuration information:


```
<BIOSCONFIG>
   <BIOSCONFIG_VERSION>2.1</BIOSCONFIG_VERSION>
   <SPEC_VERSION>2.4</SPEC_VERSION>
   <SP_NETWORK_CONFIG>
      <DISCOVERY></DISCOVERY>
      <IP></IP>
      <NETMASK></NETMASK>
      <GATEWAY></GATEWAY>
   </SP_NETWORK_CONFIG>
   <PASSWORD_CONFIG>
      <PASSWORD></PASSWORD>
   </PASSWORD_CONFIG>
   <BOOT_ORDER_OVERRIDE>
      <HELP_STRING>FIRST=Choose one of: pxe, cdrom, disk, floppy,
bios, none</HELP_STRING>
      <FIRST></FIRST>
      <HELP_STRING>CLEAR_CMOS=Choose Yes, No or leave it empty,
</HELP_STRING>
      <CLEAR_CMOS></CLEAR_CMOS>
   </BOOT_ORDER_OVERRIDE>
   <BOOT_DEVICE_PRIORITY>
      <B0>
          <DEVICE_NAME></DEVICE_NAME>
         <PCI-B-D-F></PCI-B-D-F>
      </B0>
   </BOOT_DEVICE_PRIORITY>
   <CMOS_DUMP>
   <OFFSET_00>00.00.00.00.00.00.00.00.00.00.00.00.00.00.00.00.</OFFSET_00>
   <OFFSET_10>00.30.00.30.0E.80.02.FF.FF.00.00.00.00.00.00.00.</OFFSET_10>
   <OFFSET_20>00.00.00.00.00.00.00.00.00.30.47.47.47.47.04.3A.</OFFSET_20>
   <OFFSET_30>FF.FF.20.85.90.F7.07.00.00.03.00.17.00.00.1F.3A.</OFFSET_30>
   <OFFSET_40>00.00.00.00.00.00.00.00.00.00.00.00.00.00.00.00.</OFFSET_40>
   <OFFSET_50>00.00.FF.00.13.00.00.01.80.30.30.30.30.30.00.00.</OFFSET_50>
   <OFFSET_60>EF.40.41.42.43.44.45.46.47.08.09.0A.18.00.00.0B.</OFFSET_60>
   <OFFSET_70>00.03.0C.0D.0E.0F.10.11.00.00.00.00.12.13.14.15.</OFFSET_70>
   <OFFSET_80>11.24.26.06.46.14.00.16.02.00.F8.23.C8.17.20.07.</OFFSET_80>
   <OFFSET_90>18.20.19.1A.1B.1C.1D.9E.DF.9E.DE.21.02.03.04.05.</OFFSET_90>
   <OFFSET_A0>06.07.08.09.EA.2B.0B.0B.0B.4B.00.01.0F.00.0C.00.</OFFSET_A0>
   <OFFSET_B0>00.00.00.00.10.32.54.76.10.32.54.76.14.00.00.00.</OFFSET_B0>
   <OFFSET_C0>00.46.BC.00.00.00.00.00.00.80.C0.10.42.F9.FF.FF.</OFFSET_C0>
   <OFFSET_D0>83.00.80.9C.DE.1F.40.02.FA.52.55.E0.F1.F3.E7.FF.</OFFSET_D0>
   <OFFSET_E0>7C.00.01.04.00.00.05.04.03.04.00.02.07.02.17.00.</OFFSET_E0>
   <OFFSET_F0>17.03.01.05.08.01.03.04.00.03.00.09.01.00.05.00.</OFFSET_F0>
   </CMOS_DUMP>
</BIOSCONFIG>
```

Note – The data between the `<CMOS_DUMP>` element tags contains raw CMOS data.

See Also

- "How to Apply the BIOS CMOS Golden Image" on page 36

▼ How to Apply the BIOS CMOS Golden Image

You can apply the golden image to identical hardware by copying the golden image from your system to a second system with the same BIOS revision, as shown using `-set_CMOS_dump`.

1. **Copy the *filename*.xml image from the system where you saved the CMOS configuration to another system.**

2. **Use the following command on the system that you want to copy the golden image to:**

   `biosconfig -set_CMOS_dump` *filename*`.xml`

```
Copyright (C) SUN Microsystems 2009.
BIOSconfig Utility Version 2.1
Build Date: Jul 16 2009
Build Time: 15:55:12

BIOSconfig Specification Version 2.4

Processing Input BIOS Data....

Success
```

See Also

- "How to Capture the BIOS CMOS Golden Image" on page 34

Configuring Individual CMOS Settings

`biosconfig` provides two commands to manage individual CMOS settings:

- `biosconfig -get_bios_settings`

  Gets CMOS settings from the platform.

- `biosconfig -set_bios_settings`

  Sets CMOS settings on the platform.

To use these commands you:

1. Use `-get_bios_settings` *filename*`.xml` to generate an XML file that describes the current settings.
2. Edit that XML file so that it specifies the settings.
3. Use `set_bios_settings` *filename*`.xml` to change settings in CMOS.

You can provide a subset of the XML file to include only the settings that you want to change with the `-get_bios_settings` command. The XML file must be valid, so you must remove entire option sets from the XML file.

Note – Values for the settings vary depending on your server type. `biosconfig` reads the host's BIOS image and the platform's CMOS to find the setup questions (the strings displayed in BIOS setup), the optimal default values, the current settings, and the permitted settings. The XML file structure matches the menu hierarchy in BIOS setup.

The names in the output XML file match the names in the setup menus; the only difference is that the spaces are replaced with underscores (_). For example, the Quick Boot entry in the Boot Settings Configuration submenu in the Boot menu of BIOS setup is specified like this:

```
<BIOSCONFIG>
<SETUP_CONFIG>
<Boot>
<Boot_Settings_Configuration>
<Quick_Boot>
```

This section includes the following topics:

- "Static and Dynamic CMOS Settings" on page 37
- "How to Configure a Static CMOS Setting" on page 37
- "How to Configure a Dynamic Setting" on page 39

Static and Dynamic CMOS Settings

There are two types of CMOS settings: static and dynamic. Static settings are human readable and dynamic settings are numerical.The following settings are determined at runtime by the BIOS:

- The value in CMOS
- The behavior determined by that value
- The BIOS setup strings displayed

▼ How to Configure a Static CMOS Setting

The following procedure describes how to set static CMOS settings. The XML samples shown are subsets of the output XML file.

1. **Get the CMOS settings from the platform with the following command:**

   `biosconfig -get_bios_settings` *filename*`.xml`

2. **View the following XML code examples:**

```
<BIOSCONFIG>
   <SETUP_CONFIG>
<Boot>
<Boot_Settings_Configuration>
```

```
<Quick_Boot>
<HELP_STRING>Allows BIOS to skip certain....
</HELP_STRING>
<DEFAULT_OPTION>Enabled</DEFAULT_OPTION>
<SELECTED_OPTION>Enabled</SELECTED_OPTION>
<OPTION-0>Disabled</OPTION-0>
<OPTION-1>ENabled</OPTION-1>
</Quick_Boot>
<Onboard_IB_gPXE_boot_first_>
<HELP_STRING>Set Onboard Infiniband  gPXE ....
</HELP_STRING>
<DEFAULT_OPTION>Disabled</DEFAULT_OPTION>
<SELECTED_OPTION>Disabled</DEFAULT_OPTION>
<OPTION-0>Disabled</OPTION-0>
<OPTION-1>Enabled</OPTION-2>
</Onboard_IB_gPXE_boot_first_>
</Boot_Settings_Configuration>
</Boot>
   </SETUP_CONFIG>
</BIOSCONFIG>
```

or

```
<BIOSCONFIG>
   <SETUP_CONFIG>
      <Chipset>
         <South_Bridge_Configuration>
            <Restore_on_AC_Power_Loss>
               <HELP_STRING></HELP_STRING>
               <DEFAULT_OPTION>Power On</DEFAULT_OPTION>
               <SELECTED_OPTION>Power On</SELECTED_OPTION>
               <OPTION-0>Power Off</OPTION-0>
               <OPTION-1>Power On</OPTION-1>
               <OPTION-2>Last State</OPTION-2>
            </Restore_on_AC_Power_Loss>
         </South_Bridge_Configuration>
      </Chipset>
   </SETUP_CONFIG>
</BIOSCONFIG>
```

3 **Modify the value in the <SELECTED_OPTION> tags, as needed.**

The options listed below the <SELECTED_OPTION> tags display the available values.

For example, the options for the Quick Boot setting are Disabled and Enabled

4 **Use the following command to set the static CMOS values.**

`biosconfig -set_bios_settings` *filename***`.xml`**

See Also

- "How to Configure a Dynamic Setting" on page 39

How to Configure a Dynamic Setting

`biosconfig` cannot retrieve the strings and the mapping between the values in CMOS. This behavior is BIOS dependent; the ability to retrieve this information depends on the BIOS revision and the platform type.

To configure or export dynamic settings, you need to discover the setting that you wish to use by following these steps:

1. **Enter the BIOS Setup utility.**

2. **Configure the settings manually and save the configuration.**

3. **To examine the resulting XML output to find the value that the BIOS is using for the setting you wish to specify, type the following command:**

   `biosconfig -get_bios_settings` *filename*`.xml`

   This is an example of a dynamic CMOS setting as displayed in the XML file:

```
<BIOSCONFIG>
<SETUP_CONFIG>
<Boot>
<Option_ROM_Enable>
<NET0_Option_ROM_>
<HELP_STRING>This Option enables execut....
</HELP_STRING>
<DEFAULT_OPTION> 0000 </DEFAULT_OPTION>
<SELECTED_OPTION> 0000 </SELECTED_OPTION>
<OPTION_RANGE> 0000 - 0001 </OPTION_RANGE>
<OPTION-0>Not Available</OPTION-0>
</NET0_Option_ROM_>
</Option_ROM_Enable>
</Boot>
</SETUP_CONFIG>
</BIOSCONFIG>
```

   In the preceding code, there are no string-to-value mappings offered by the `biosconfig` output.

4. **Use the following command to set the BIOS configuration:**

   `biosconfig -set_bios_settings` *filename*`.xml`

5. **Use this XML file to configure dynamic CMOS settings on machines of the same model.**

See Also

- "How to Configure a Static CMOS Setting" on page 37

Commands That Produce Unrelated, Innocuous, Extra Output

The following is a known issue with `biosconfig`.

Some commands have extraneous output in the XML file. For example, the following is the extra output from `-get_cmos_dump`.

```
<SP_NETWORK_CONFIG>
<DISCOVERY></DISCOVERY>
<IP></IP>
<NETMASK></NETMASK>
<GATEWAY></GATEWAY>
</SP_NETWORK_CONFIG>
<PASSWORD_CONFIG>
<PASSWORD></PASSWORD>
</PASSWORD_CONFIG>
<BOOT_ORDER_OVERRIDE>
<HELP_STRING>FIRST=Choose one of: pxe, cdrom, disk,
floppy, bios, none</HELP_STRING>
<FIRST></FIRST>
<HELP_STRING>CLEAR_CMOS=Choose Yes, No or leave it
empty, empty means No</HELP_STRING>
<CLEAR_CMOS></CLEAR_CMOS>
</BOOT_ORDER_OVERRIDE>
<BOOT_DEVICE_PRIORITY>
<B0>
<DEVICE_NAME></DEVICE_NAME>
<PCI-B-D-F></PCI-B-D-F>
</B0>
</BOOT_DEVICE_PRIORITY>
```

See also:

- "How to Configure a Static CMOS Setting" on page 37
- "How to Configure a Dynamic Setting" on page 39

Using the `ubiosconfig` Tool

`ubiosconfig` provides a CLI tool for configuring BIOS on Oracle x86 servers that support UEFI BIOS. For other x86 systems, use the `biosconfig` tool. See "Using the `biosconfig` Tool" on page 15

For information on the systems supported for each tool, refer to the support matrix at:

`http://www.oracle.com/goto/system-management`

For more information on your server's UEFI BIOS, see your server documentation. `ubiosconfig` enables you to save the settings from a server's UEFI BIOS to an XML file, then load the settings from the XML file to configure another server's UEFI BIOS.

The following topics are covered in this section:

`ubiosconfig` Command Overview

The `ubiosconfig` commands adhere to the following command syntax:

`ubiosconfig` *subcommand* **[***option***]**

The options listed in the following table apply to all CLI Tools commands including `ubiosconfig`.

Short Option	Long Option	Description
`-?`	`--help`	Displays help information.
`-V`	`--version`	Displays the tool version.

If you use the `--help` or `--version` options, the `ubiosconfig` command does not require subcommands, otherwise one or more subcommands are mandatory.

The options listed in the following table are required to use `ubiosconfig` on a remote server's UEFI BIOS.

Short Option	Long Options	Description
`-H`	`--remote-hostname`	This option is followed by the IP address of the target server.
`-U`	`--remote-username`	This option is followed by the user name used to log in to remote server.

For example:

`ubiosconfig export all --remote-hostname=`*address* **`--remote-username=`***username*

where *address* is the remote server's hostname or IP address in xx.xx.xx.xx format, and *username* is the user name with access rights to the server.

`ubiosconfig` supports the subcommands listed in the following table.

Command	Function
`import`	Import a configuration XML file that will be applied to the server's UEFI BIOS at next boot.
`export`	Export the server's UEFI BIOS configuration to a local XML file.
`cancel`	Cancel pending UEFI BIOS configuration changes.
`list`	List status information regarding pending UEFI BIOS import or export operations.
`reset`	Reset the server's UEFI BIOS configuration to factory default at next boot.

`export` Subcommand

The `export` subcommand exports a server's UEFI BIOS settings to an XML file. The format of the `export` subcommand is:

`ubiosconfig export` *type* **`-x`** *filename***`.xml`** *option*

where *type* is one of the options described below, *filename* is the optional path, and *option* is one of the options described below.

The supported type for export is listed in the following table.

Type	Description
`all`	Export all options from the server's BIOS.

The supported options for export are listed in the following table.

Short Option	Long Option	Description
`-x`	`--xml_file`	The path to the XML file. Without this option, settings are displayed on the screen.
`-f`	`--force`	Ignore safeguards, and import BIOS XML file regardless of current system state.

Note – There is no guarantee of accuracy in the data when using the `--force` option.

`import` Subcommand

The `import` subcommand imports UEFI BIOS settings stored in an XML file to the server at next boot. The format of the `import` subcommand is:

`ubiosconfig import` *type* **`-x`** *filename*`.xml` *option*

where *type* is one of the options described below, *filename* is the path to the XML file you want to import settings from and *option* is one of the options described below.

The supported types of import are listed in the following table.

Type	Description
`all`	Import all options from the XML file to the server's BIOS at next boot.
`boot`	Import only boot options from the XML file to the server's BIOS at next boot.
`config`	Import only configuration options from the XML file to the server's BIOS at next boot.

The possible option for the import is listed in the following table.

Short Option	Long Option	Description
`-f`	`--force`	Ignore safeguards, and import BIOS XML file regardless of current system state. Potentially dangerous and not recommended.

Note – There is no guarantee of accuracy in the data when using the `--force` option.

`list` Subcommand

The `list` subcommand displays information on import or export changes to UEFI BIOS settings at the next server boot. The format of the `list` subcommand is:

```
ubiosconfig list all
```

`cancel` Subcommand

The `cancel` subcommand cancels any pending changes to UEFI BIOS settings. The format of the `cancel` subcommand is:

```
ubiosconfig cancel config
```

`reset` Subcommand

The `reset` subcommand resets the UEFI BIOS settings to factory default at the next server boot. The format of the `reset` subcommand is:

`ubiosconfig reset` *type*

where *type* is one of the options listed in the following table.

Type	Description
`config`	Reset the server's UEFI BIOS to factory defaults at next power cycle. Any pending UEFI BIOS changes from `ubiosconfig` are added on top of the factory defaults.
`cancel`	Cancel any pending reset change to the server's UEFI BIOS settings.

Using the `fwupdate` Tool

`fwupdate` is a cross-OS utility that enables you to update, query, and validate the firmware of Oracle server devices such as host bus adapters (HBAs), Oracle Integrated Lights Out Manager (ILOM) service processor, BIOS, SAS expanders, SAS controllers, and different types of disk drives.

This section includes the following topics:

`fwupdate` Overview

`fwupdate` enables you to work with firmware files for the following targets:

- Disk drives (spinning media and flash drives)
- Oracle ILOM service processor and BIOS
- HBA and embedded storage controllers, SAS1 and SAS2
- LSI SAS expander devices, SAS1 and SAS2
- Emulex and QLogic Fiber Channel controllers
- Mellanox InfiniBand controllers

You can use `fwupdate` to do the following:

- Examine firmware information for devices in a server
- Check firmware file compatibility
- Update device firmware using an automated XML metadata file
- Update firmware manually using a raw firmware file
- Control how devices are reset after updating the firmware

fwupdate Command Prerequsite

Before using the fwupdate command to update device firmware, you must quiesce the device.

Caution – System hang or data loss. Before updating device firmware, make sure that the device is quiesced.

For example, when updating the firmware on a hard drive:

- Make sure that the operating system is not accessing the disk (for example, the system boot disk).
- Make sure that an application is not accessing the disk (for example, a database application).
- If hardware RAID is being used on the system, make sure that the RAID controller is not accessing the disk (for example, if it is rebuilding an array or is in a degraded state). You can use `raidconfig` to check the state of the arrays.

Downloading Firmware Files

Download firmware files from `http://support.oracle.com`.

Search for the product that you want to update and download the latest firmware package available for that product.

Automatic and Manual Update Modes

With Oracle Hardware Management Pack 2.1 and later, the `fwupdate` tool supports automatic mode and manual mode.

- *Automatic mode* uses information from the XML metadata file that is packaged with the platform firmware downloads to update the device firmware. This is the most accurate method to use.
- *Manual mode* allows you to update the firmware directly. This mode should only be used if the XML metadata file is not available for the device that you want to update.

  Check the documentation and release notes for your product to determine whether or not an XML metadata file is available. The product release notes also contain upgrade information specific to the device you are upgrading.

Host-to-ILOM Interconnect

With Oracle Hardware Management Pack 2.2 and later, `fwupdate` supports the Host-to-ILOM Interconnect, which can significantly speed up the firmware update process. For more information, refer to:"Enabling the Host-to-ILOM Interconnect" in *Oracle Hardware Management Pack Installation Guide*.

fwupdate Command Overview

The options listed in the following table apply to all CLI Tools commands, including fwupdate.

Short Option	Long Option	Description
-?	--help	Displays help information.
-V	--version	Displays the tool version.

One subcommand is mandatory unless you use the --help or --version options.

fwupdate supports the subcommands listed in the following table.

Subcommand	Description
list	List mode displays system data and helps select components for upgrade.
update	Update mode enables a single component to be updated based on command-line directives.
reset	Reset mode enables individual components to be reset.

The subcommands are described in the following sections.

Device naming of target devices is shared with other CLI Tools based on the storage library.

For a full description of the naming convention, see: "CLI Tools Device-Naming Convention" on page 12.

See also:

- "CLI Tools Command Syntax and Conventions" on page 11

fwupdate Command-Line Interface

This section covers the following topics:

Automatic Mode fwupdate Command-Line Interface

Automatic command-line mode uses an XML metadata file that is included in the platform firmware downloads.

The following prerequisites must be met before using the `fwupdate` command in automatic mode:

- You must have root permission to run `fwupdate` commands on Unix-based platforms, or administrator permission for Windows platforms.
- An XML metadata file containing information on platform firmware must be available. Check the firmware release notes to see if this file is available.
- For Oracle Solaris systems, after hot-plugging a device, run the `devfsadm -C` command to reenumerate all of the system device nodes before running the `fwupdate` command.

When a command fails, it returns one of several failure codes listed in "`fwupdate` Error Codes" on page 125.

Note – Running the command without subcommands displays help.

Automatic `fwupdate` mode uses the following syntax:

`fwupdate` *subcommand target* **`-x`** *filename*`.xml` *options*

where *target* is the type of device that is being listed or updated, *filename* is the XML file that contains the firmware update metadata, and *subcommand* is one of the following.

Subcommand	Description
`list`	Provides firmware information about a device or a file.
`update`	Updates a single component based on command-line directives.

See also:

- "`list` Subcommand" on page 49
- "`update` Subcommand" on page 54

Manual Mode `fwupdate` Command-Line Interface

Manual command-line mode enables you to update a single component with a specified firmware file. In addition, you can list information about a component's current firmware, the firmware in a file, and which components a firmware file is compatible with. You can also reset a component as part of the firmware upgrade process.

The following requirements apply when using the manual mode `fwupdate` command:

- You must have root permission to run `fwupdate` commands on Unix-based platforms or administrator permission for Windows platforms.

- Only one target device can be upgraded per command-line execution.
- Only one file type and file can be specified by the command line.
- Components with multiple and different firmware files require a separate command-line execution.
- For servers running Oracle Solaris OS: After hot-plugging any device, run the `devfsadm -C` command to reenumerate all of the system device nodes before running the `fwupdate` command.

When a command fails, it returns one of several failure codes listed in "`fwupdate` Error Codes" on page 125.

Note – Running the command without subcommands displays help.

The tool uses the following syntax:

`fwupdate` *subcommand target options*

where *target* is the type of device that is being listed or updated, *options* are options specific to the subcommand, and *subcommand* is one of the following:

Subcommand	Description
`list`	Displays system data and helps select components for upgrade.
`update`	Updates a single component based on command-line directives.
`reset`	Resets individual components.

See also:

- "`list` Subcommand" on page 49
- "`update` Subcommand" on page 54
- "`reset` Subcommand" on page 57

`list` Subcommand

The `list` command does the following:

- Displays the version of firmware for all components
- Tells you whether the target device can be updated with the XML metadata file
- Saves the configuration information to a specified XML file

This information can be used to check the state of a device before executing a firmware upgrade and can be used to verify that a firmware update has been successful.

The list options are listed in the following table.

Short Option	Long Option	Description
-n	--device_name	Allows a mandatory parameter to designate a single device to list. The --device_name option is the common-mapped device name.
-v	--verbose	Displays detailed information about each component listed. Verbose is off by default.
-x	--xml=file	Uses the provided XML metadata file to determine which components are supported.
-o	--output_xml=file	Prints the configuration information in XML format to the given file.

The list command has two types of targets: the first lists the configuration of devices on the system or supported by a file, and the second lists the features supported by fwupdate.

The following supported targets for the list subcommand represent all of the supported component types that can be upgraded by fwupdate:

- all
- disk
- expander
- controller
- bridge
- sp_bios

For example, use the all option to view all of the devices that can be updated using the XML metadata file.

The following supported targets for the list command represent the devices that can be upgraded by fwupdate:

- supported-targets
- supported-images
- error-codes

For example, use the supported-targets option to view all of the types of target device that can be updated using fwupdate.

The following information is displayed with the list command for the following targets. Items marked with an asterisk (*) are displayed in verbose listing.

- SP BIOS
  - ID
  - Product Name
  - ILOM Version

  - BIOS/OBP Version
  - XML Support
- Controllers
  - ID
  - Type
  - Manufacturer
  - Mode
  - Product Name
  - Firmware (F/W) Version
  - BIOS version
  - EFI Version
  - FCODE Version
  - Package Version
  - NVDATA Version
  - XML Support
  - NODE ID*
  - Part Number*
  - PCI Address*
  - PCI Vendor ID*
  - WWN*
- Disk
  - ID
  - Manufacturer
  - Model
  - Chassis
  - Slot
  - Type
  - Media
  - Size
  - Firmware (FW) Version
  - XML Support
  - NODE ID*
  - WWN*
- Expander
  - ID
  - Chassis
  - Slot
  - Manufacturer
  - Model
  - Expander Name
  - Firmware (F/W) Version
  - XML Support

  - NODE ID*
  - Product Revision*
  - WWN*
- Bridge
  - ID
  - Chassis
  - Slot
  - Manufacturer
  - Model
  - Firmware (F/W) Version
  - Att FW Version
  - XML Support
  - NODE ID*
  - WWN*

The following are some examples of the output for `fwupdate list` commands:

`fwupdate list disk -v`

```
==================================================
CONTROLLER
==================================================
ID: c0
Node ID: mptir2:40:00.0
Type: SAS
Manufacturer: LSI Logic
Model: 0x0072
Product Name: SGX-SAS6-REM-Z
FW Version: 11.05.02.00
BIOS Version: 07.21.04.00
EFI Version: 07.18.02.13
FCODE Version: 01.00.60.00
PCI Address: 40:00.0
PCI Vendor ID: 0x1000
WWN: 0x500605b005243000
NVDATA Version: 10.03.00.26 (default) 10.03.00.27 (persistent)
XML Support: N/A


DISKS
===============
ID: c0d0
Manufacturer: HITACHI
Model: H106030SDSUN300G
Slot: 2
Node ID: PDS:5000cca02515b089
Type: sas
Media: HDD
Size (GB): 300
FW Version: A2B0
XML Support: N/A
```

```
ID: c0d1
Manufacturer: HITACHI
Model: H106030SDSUN300G
Slot: 3
Node ID: PDS:5000cca025143f79
Type: sas
Media: HDD
Size (GB): 300
FW Version: A2B0
XML Support: N/A
```

fwupdate list sp_bios -x metadata_3.1.2.10.b.xml

```
SP + BIOS
==================================================
ID        Product Name        ILOM Version         BIOS/OBP Version   XML Support
-------------------------------------------------------------------------------------
sp_bios   SUN FIRE X4170 M3   v3.1.2.10.a r75921   17030100           Yes
```

fwupdate list controller -n c0 -v

```
CONTROLLER
=============
ID: c0
Node ID: mptmega:41:00.0
Type: SAS
Manufacturer: LSI Logic
Model: 0x0079
Product Name: LSI MegaRAID SAS 9261-8i
FW Version: 2.130.353-1803
BIOS Version: 3.24.00
EFI Version: 4.12.05.00
FCODE Version:
PCI Address: 41:00.0
PCI Vendor ID: 0x1000
XML Support: N/A
```

fwupdate list disk -n c2d0

```
DISK
=============
ID     Manufacturer   Model               Chassis Slot   Type   Media   Size (GB) FW Version XML Support
-----------------------------------------------------------------------------------------------------------
c2d0   ATA            3E128-TS2-550B01    -       -      sata   SSD     100       TI35       N/A
```

fwupdate list disk -n c2d0 -v

```
DISK
=============
ID: c2d0
Manufacturer: ATA
Model: 3E128-TS2-550B01
Node ID: PDD:/dev/sg3
Type: sata
Media: SSD
Size (GB): 100
```

```
FW Version: TI35
XML Support: N/A
```

fwupdate list expander -n c1x0

```
EXPANDER
=============
ID     Chassis Slot Manufacturer  Model     Expander Name  FW Version   XML Support
-------------------------------------------------------------------------------------
c1x0   0       -    ORACLE        DE2-24P   Primary        0010         N/A
```

fwupdate list expander -n c1x0 -v

```
EXPANDER
=============
ID: c1x0
Chassis: 0
Manufacturer: ORACLE
Model: DE2-24P
Expander Name: Primary
FW Version: 0010
Product Revision: 0010
Node ID: EC:mpt2sas:30:00.0:5080020001431f3e
XML Support: N/A
```

See also:

- "update Subcommand" on page 54
- "reset Subcommand" on page 57

update Subcommand

The following topics are covered in this section:

- "Automatic Mode update Subcommand" on page 54
- "Manual Mode update Subcommand" on page 56

Automatic Mode update Subcommand

The automatic mode update subcommand updates the specified target device by using the firmware update information contained in the XML metadata file provided in the platform firmware download.

When using an XML metadata file, the update subcommand supports the following targets:

- all
- expander
- disk
- bridge

- controller
- sp_bios

Note – Emulex Fiber Channel cards cannot be updated using fwupdate on a server running Oracle Solaris 11 or SUSE Linux Enterprise Server 11.

Options for the update subcommand in automatic mode are listed in the following table.

Short Option	Long Option	Descriptions
-n	--device_name	Precedes name of the device to update. The name is the mapped name, which you can retrieve by using the fwupdate list command. This option is mandatory for single component mode, but optional when used with an XML file.
-d	--dry-run	Optional. Checks all input, executes an available dry-run check command on the firmware and component, but makes no permanent changes.
-x	--xml=*filename*	If the firmware package contains a metadata XML file, this command provides the path to the *filename*.
-o	--output=*filename*	Logs all actions in the specified file.
-p	--priority=*value*	Starts processing input metadata from an XML file at a given priority level, skipping all lower levels.
-q	--quiet	Suppresses informational message output and only returns error codes.
n/a	-- silent-reboot	Enables a reboot to update the firmware without a prompt. Reboot happens automatically.
n/a	--silent-no-reboot	Enables no-reboot option without a prompt. User is not prompted, and a reboot will not take place. **Note –** The system might need to be rebooted to complete the firmware update.

The following are automatic mode update command examples:

- **fwupdate update all -x** *filename***.xml**
- **fwupdate update disk -x** *filename***.xml -n c0d1**

See also:

- "Manual Mode update Subcommand" on page 56
- "list Subcommand" on page 49

Manual Mode update Subcommand

The manual mode update command enables you to update a single device using a firmware image file. Only one firmware image and one component can be specified per execution of this command.

The manual mode update subcommand supports the following targets:

- sp-bios-firmware
- disk-firmware
- expander-firmware
- expander-manufacturing_image
- fc-controller-firmware
- ib-controller-firmware
- sas-bridge-firmware
- sas-controller-firmware
- sas-controller-bios
- sas-controller-fcode
- sas-controller-efi

Note – Emulex Fiber Channel cards can not be updated using fwupdate on a server running Oracle Solaris 11 or SUSE Linux Enterprise Server 11.

Note – You cannot update the disks on the Flash Accelerator F40 PCIe Card directly. The Flash Accelerator F40 PCIe Card SSDs are updated when the general firmware package is applied to the PCIe card. You encounter an error if you try to apply an update directly to the Flash Accelerator F40 PCIe Card SSD devices.

Options for the update subcommand in manual mode are listed in the following table.

Short Option	Long Option	Descriptions
-n	--device_name	Name of the device to update. The name is the mapped name, which you can retrieve by using the fwupdate list command. This option is mandatory for single component mode, but optional when used with an XML file.
-f	--filename=*filename*	A mandatory option, with a mandatory parameter, designating the name of the firmware image file that is to be applied.
-r	--reset	Reset the component after the update has finished.

Short Option	Long Option	Descriptions
-d	--dry-run	Optional. Checks all input, executes an available dry-run check command on the firmware and component, but makes no permanent changes.
-o	--output=*filename*	Log all actions in the specified file.
-q	--quiet	Suppresses informational message output and only returns error codes.

The following is a manual mode update command example:

```
fwupdate update disk-firmware -n c1d1 -f diskfirmware.file
```

Note – Only one device can be specified per fwupdate execution. Every device that needs to be updated must be run as a separate fwupdate command.

See also:

- "Automatic Mode update Subcommand" on page 54
- "list Subcommand" on page 49
- "reset Subcommand" on page 57

reset Subcommand

After firmware for a device has been updated using the manual firmware update process, the device might need to be reset. This requirement is different with each device; therefore, the reset functionality might be part of the update procedure or a separate function. To determine if your device requires a reset after a firmware upgrade, consult the release notes of your firmware.

The reset subcommand supports the following targets:

- expander
- controller
- sp_bios

Options for the reset subcommand are listed in the following table.

Short Option	Long Option	Description
-n	--device_name	A mandatory option with a mandatory parameter, which designates a single device to show. *device_name* is the common-mapped device name.

The following example shows the fwupdate reset used to reset the controller mapped to c2.

```
fwupdate reset controller -n c2
```

See also:

- "update Subcommand" on page 54
- "list Subcommand" on page 49

fwupdate Network Based Service Processor Options

fwupdate can be used over a network connection. The connection can be an external network to either the local or a remote service processor or the Host-to-ILOM connection to the local system.

The options listed in the following table are supported for fwupdate over a network connection.

Short Option	Long Options	Description
-H	--remote_sp_ip	Specifies the SP IP address. This option is followed by the network connection target service processor IP address.
-U	--remote_username	Specifies the user name used to log in to remote service processor. This option is followed by the remote service processor's login user name.

When using these options to access a service processor over a network connection, you must use both options together.

Note – The password required by the network connection can be piped in on stdin for scripting use.

See also:

- "How to Use fwupdate to Update an Oracle ILOM Service Processor" on page 58

▼ How to Use fwupdate to Update an Oracle ILOM Service Processor

The following example shows how to use fwupdate to update an Oracle ILOM service processor and system BIOS or OBP with new firmware. You can use fwupdate to update either the local service processor or a remote service processor. When using local update, fwupdate uses the fastest local interface available. If a Host-to-ILOM connection is available, then this fast connection is used, otherwise the KCS interface is used.

Caution – Loss of server functionality. Updating the service processor firmware incorrectly might damage the service processor.

Before You Begin

- Ensure the LAN-over-USB connection or the KCS interface is correctly configured to communicate with the target Oracle ILOM service processor.
- Download the service processor updates from `http://support.oracle.com`.

  These include the metadata or firmware file for the target Oracle ILOM service processor.
- To make sure that the firmware is compatible with the target service processor, read all documentation and release notes included with the firmware file before proceeding,

1. **To display information about the service processor and system BIOS firmware, do one of the following:**

   - **To list information about the local Oracle ILOM service processor and the system BIOS firmware, type the following command:**

     `fwupdate list sp_bios`

   - **To list information about an Oracle ILOM service processor and system BIOS firmware using a network connection, type the following command:**

     `fwupdate list sp_bios -H` *remote_ip* **`-U`** *ilom_user*

     where *remote_ip* is the IP address of the service processor and *ilom_user* is the user name for logging in to the service processor.

     Enter the Oracle ILOM password when prompted.

   Output related to the target is displayed. For example the output from this command is similar to:

```
================================================
SP + BIOS
================================================
ID      Product Name           ILOM Version     BIOS/OBP Version  XML Support
-----------------------------------------------------------------------------
sp_bios SUN FIRE X4270 SERVER  v3.0.12.0 r64525 07060223          N/A
```

Note – When the ILOM and OBP are updated on a SPARC system, the system reboots the host automatically. Shut down cannot be disabled with the `silent-no-reboot` option.

2. **To update the Oracle ILOM service processor, choose the local or remote option:**

   - *Local*: **To update the local Oracle ILOM service processor, choose one of these steps:**

     - **If a `metadata.xml` file is available, type this command to use automatic mode:**

       `fwupdate update sp_bios -x` *metadata*`.xml`

where *metadata*`.xml` is the path to the metadata file.

Note – For x86 systems, you can also use the `--silent-reboot` or `--silent-no-reboot` commands after the .xml file to either automatically reboot or not reboot the server after the firmware update.

- **If `metadata.xml` file is not available, type this command to use manual mode:**

  `fwupdate update sp-bios-firmware -n sp_bios -f` *sp-bios-fimware-package-file***`.pkg`**

  where *sp-bios-fimware-package-file*`.pkg` is the path to the firmware file for your server's Oracle ILOM service processor.

- *Remote*: **To update an Oracle ILOM service processor using a network connection, do one of the following:**

  - **If a `metadata.xml` file is available, type this command to use automatic mode:**

    `fwupdate update sp_bios -x` *metadata***`.xml -H`** *remote_ip* **`-U`** *ilom_user*

    where *metadata*`.xml` is the path to the metadata file for your server's Oracle ILOM service processor.

    Note – For x86 systems, you can also use the `--silent-reboot` or `--silent-no-reboot` commands after the .xml file to either automatically reboot or not reboot the server after the firmware update.

  - **If a `metadata.xml` file is not available, type this command to use manual mode:**

    `fwupdate update sp-bios-firmware -n sp_bios -f` *sp-bios-fimware-package-file***`.pkg -H`** *remote_ip* **`-U`** *ilom_user*

    where *sp-bios-fimware-package-file*`.pkg` is the path to the firmware file for your server's Oracle ILOM service processor.

The Oracle ILOM service processor is updated.

3 Reboot the host server to initialize the BIOS update.

- **For SPARC systems: the host is automatically rebooted.**
- **For x86 systems:**
  - **If the metadata file is available and you used the `--silent-reboot` option with the `fwupdate` command, the system automatically reboots.**

- **If the metadata file is available and the system does not automatically reboot, type y at the prompt:**

  Do you wish to automatically reboot now? [y/n]?

- **If no metadata.xml file is available or the metadata file does not contain the reboot instructions, reboot the host server manually.**

See Also

- "`update` Subcommand" on page 54
- "`reset` Subcommand" on page 57
- "`list` Subcommand" on page 49

Execution Summary

After the `fwupdate` tool is used to upgrade firmware, an execution summary provides information on whether or not the upgrade was successful. This information is also written to the log file.

The following examples show the possible execution summary messages:

- Message printed after a successful dry-run/check function:

  `Check firmware successful for device:` *device_name*

- The upgrade was successful, but no firmware version information is available for this component:

  `Upgrade of firmware for` *device_name* `succeeded. Version information was not available.`

  Consult your product release notes for information on how to verify the upgrade.

- Upgrade was successful:

  `Upgrade of` *device_name* `from` *old_fw* `to` *new_fw* `succeeded.`

- The version number of the software did not change after a successful upgrade:

  `Upgrade of` *device_name* `from` *old_fw* `succeeded, but is not yet active.`

  This might mean that the server needs to be reset, or that other instructions need to be followed. Consult your product release notes for instructions on how to update the version number.

- Upgrade failed:

  `Upgrade of` *device_name* `failed:` *error_message*

The variables in the previous output represent the following:

- *device_name* is the logical name of the device that is being upgraded.
- *old_fw* is the old firmware version.

- *new_fw* is the new firmware version.
- *error_message* is the error message that explains why the firmware update did not succeed.

See also:

- "`update` Subcommand" on page 54
- "`list` Subcommand" on page 49
- "`reset` Subcommand" on page 57

Using the raidconfig Tool

raidconfig is part of the Oracle Hardware Management Pack and uses a general-purpose cross-OS storage management library to configure attributes to RAID volumes using XML files.

This section includes the following topics:

raidconfig Overview

raidconfig allows you to explore, monitor, and configure storage resources connected to the system.

Note – raidconfig is not currently supported for the SPARC M5–32 server.

---

raidconfig provides the following functions:

- Shows, creates, deletes, and modifies RAID volumes.
- Facilitates scripting by using command-line options.
- Configures many similar and dissimilar platforms in a data center.
- Displays the current RAID configuration and writes it to an XML file so it can be edited and used to configure the same or a different platform.

- Represents a logical disk in a portable manner.

  For example, using a unique enumeration per controller, instead of a SAS address, facilitates moving the XML file to other platforms.
- Provides a super-set of all configuration options provided by the Adaptec and LSI CLI commands.
- Uses capability checking for particular adapters based on data retrieved from the API.
- Creates nested RAID volumes depending on the controller.

raidconfig Requirements

raidconfig has the following restrictions:

- You must be in root permission level to run raidconfig commands on Unix-based platforms, or Administrator permission level for Windows platforms.
- On Oracle Solaris, raidconfig is not compatible with the raidctl CLI tool. raidconfig supports SAS2, but the raidctl tool does not.
- For servers running Oracle Solaris, after hot-plugging any device, run the **devfsadm -C** command to reenumerate all of the system device nodes before running the raidconfig command.

See also "raidconfig Command Overview" on page 64.

raidconfig Command Overview

The raidconfig commands use the following command syntax:

raidconfig *subcommand device-type -option(s)*

When a command fails, it returns one of several failure codes listed in "raidconfig Error Codes" on page 121.

The options shown in the following table apply to all CLI Tools commands including raidconfig.

Short Option	Long Option	Description
-?	--help	Displays help information.
-V	--version	Displays the tool version.
-q	--quiet	Suppresses informational message output and only returns error codes.
-y	--yes	Confirms operation. Does not prompt user for confirmation on the operation when running.

The raidconfig command requires subcommands unless used with the --help or --version options.

The following table lists the raidconfig subcommands.

Command	Function
list	Lists information on controllers, RAID volumes and disks, including disks not in a RAID volume. Specific devices can be selected for display.
create	Creates a RAID volume.
delete	Deletes a RAID volume.
add	Adds a specified disk or spare.
remove	Removes a specified disk or spare.
modify	Modifies a RAID volume or a disk.
start	Starts a maintenance task.
stop	Stops a maintenance task.
restore	Finds the RAID configuration saved on a disk and restore it.
clear	Clears the RAID configuration saved on the disks of a defined controller.
export	Generates an XML file from a RAID configuration.
import	Reads in a RAID configuration and create RAID volumes and spares.

Whenever devices (controllers, RAID volumes, and disks) are used with commands, they must be uniquely identified. For information on how to do so, see the device-naming scheme at "CLI Tools Device-Naming Convention" on page 12.

Device naming is shared with other CLI Tools based on the storage library.

See also "CLI Tools Command Syntax and Conventions" on page 11.

list Subcommand

This section includes the following topics:

list Subcommand Overview

The list subcommand displays controller, RAID volume, and disk data. The device types for the raidconfig list are listed in the following table.

Subcommand	Description
all	Shows details on all controllers, physical disks, and RAID volumes.
controller	Shows details on all controllers.
disk	Shows the physical disks.
raid	Shows all RAID details.

The raidconfig list command supports options listed in the following table.

Short Option	Long Option	Description
-c	--controller	Shows details about a particular controller. This option is followed by the controller ID string.
-r	--raid	Shows details about a particular RAID volume. This option is followed by the RAID ID string.
-d	--disks	Shows details about particular disk(s). This option is followed by a comma-separated list of the disk ID strings.
-v	--verbose	Lists all fields. By default, a brief listing shows only a subset of the fields.

The following data is displayed. Items marked with an asterisk (*) show a brief listing; all other items show a verbose listing. Items that were added to the 2.2.6 version are marked with (2.2.6).

Controllers:

- Node ID
- Manufacturer*
- Model*
- Part number
- Firmware(F/W) version*
- Serial Number (2.2.6)
- RAID Volumes*
- Disks*
- Disks in use by another controller
- PCI address
- PCI vendor ID
- PCI device ID

- PCI subvendor ID
- PCI subdevice ID
- Battery backup status
- Maximum RAID volumes
- Maximum disks per RAID volume
- Supported RAID levels
- Maximum dedicated spares
- Maximum global spares
- Stripe size minimum
- Stripe size maximum
- Disable Auto Rebuild (2.2.6)

Disks:

- ID*
- Chassis ID*
- Slot ID*
- Node ID
- Mapped to host OS (true/false)
- Device
- Disabled (true/false)
- In use by another controller
- RAID ID*
- Status*
- Type*
- Media*
- Manufacturer
- Model
- Size
- Serial number
- NAC name
- Spare state (global, dedicated, or N/A)*
- Current task
- Stoppable tasks
- Startable tasks
- Task state
- Task completion percent

RAID volumes:

- Logical ID (0-based)*
- Node ID
- Device name*
- Name (user assigned)*
- Status*
- RAID level*

- Number of disks*
- Capacity*
- Mounted
- Stripe size
- Leg size
- Read cache
- Write cache
- Current task
- Task state
- Task completion percent
- Stoppable tasks
- Startable tasks
- BIOS Boot Target (2.2.6)

See also:

- "Brief Listing Example" on page 68
- "Detailed Listing Example" on page 69

Brief Listing Example

The following example shows a brief listing of all available controllers, RAID volumes, disks in use, and available disks:

```
raidconfig list all
CONTROLLER c0
=============
Manufacturer   Model     F/W Version    RAID Volumes   Disks
-------------------------------------------------------------
Adaptec        0x0285    5.2-0          4              8
RAID Volumes
============
ID     Name               Device     Status        Num Disks   Level   Size (GB)
-------------------------------------------------------------------------------
c0r0   0919XF5017-0       /dev/sda   OK            1           Simple  146
c0r1   raid1              /dev/sdb   OK            2           0       293
c0r2   raid2              /dev/sdc   OK            3           10      146
c0r3   noname             /dev/sdd   OK            2           0       293
DISKS In Use
============
ID      Chassis   Slot   RAID ID   Status    Type   Media   Spare      Size (GB)
-------------------------------------------------------------------------------
c0d0    0         0      c0r0      OK        sas    HDD     -          146
c0d1    0         1      c0r2      OK        sas    HDD     -          146
c0d2    0         2      c0r3      OK        sas    HDD     -          146
c0d3    0         3      c0r3      OK        sas    HDD     -          146
c0d4    0         4      c0r2      OK        sas    HDD     -          146
c0d5    0         5      c0r2      -         sas    HDD     Dedicated  146
c0d6    0         6      c0r1      OK        sas    HDD     -          146
c0d7    0         7      c0r1      OK        sas    HDD     -          146
```

See also:

- "list Subcommand Overview" on page 66
- "Detailed Listing Example" on page 69

Brief Listing of a Disk Example

The following example shows of a brief listing of a disk:

```
raidconfig list disk -d c0d0
```

```
DISKS Available
===============
ID      Chassis   Slot   RAID ID   Status    Type   Media   Spare      Size (GiB)
-------------------------------------------------------------------------------------
c0d0    0         0      -         -         sas    HDD     -          279
```

See also:

- "list Subcommand Overview" on page 66
- "Detailed Listing Example" on page 69

Detailed Listing Example

The following example shows a detailed listing of a disk:

```
raidconfig list disk -d=c0d0 -v
```

```
Disk c0d0
=========
ID: c0d0
Chassis: 0
Slot: 0
Node ID: PDS:5000cca0257dbac1
Mapped to Host OS: true
Device: 5000CCA0257DBAC0
Disabled: false
Type: sas
Media: HDD
Manufacturer: HITACHI
Model: H106030SDSUN300G
Size (GiB): 279
Serial Number: 001214N74K2B        PQJ74K2B
NAC Name: /SYS/SASBP/HDD0
Current Task: none
```

See also:

- "list Subcommand Overview" on page 66
- "Brief Listing Example" on page 68

create raid Subcommand

The create raid subcommand is used to create RAID volumes. This subcommand must take -d option followed by one or more of the options shown in the following table.

Short Option	Long Option	Description
-d	--disks	Specifies a list of disks with a comma separating the disk ID numbers.
N/A	--level	Specifies the RAID level of the volume e.g. 0, 1, 1E, 5, 10, 50, 60 etc. The levels supported for a particular controller can be seen in the controller 'Supported RAID Levels' field from the list command. If this option is not supplied, a level of '0' is used.
N/A	--stripe-size	Specifies the stripe size, in kilobytes, of the RAID volume to be created. If this option is not supplied, the controller uses a default size.
N/A	--subarrays	For nested RAID levels (10, 50), specifies the size of the RAID components in number of physical disks.
N/A	--name	Assigns the user-defined name that identifies the RAID volume. This name can be set to an empty string ("").
N/A	--subdisk-size	See "Creating RAID Volumes With Partial Disks" on page 81.

The maximum capacity of the RAID volume is not configurable. You can create RAID from partial disks if the HBA or controller support it, and all the disks must all be of the same size.

▼ How To Create a RAID Volume

- **To create a RAID volume, type the following command:**

  raidconfig create raid *options* **-d** *disks*

  For example, to create a RAID 0 volume with a stripe size of 128 Kb on controller 1, type the following command:

  ```
  raidconfig create raid --stripe-size 128 -d c1d0,c1d1
  ```

See Also "create raid Subcommand" on page 70

delete raid Subcommand

The delete raid subcommand is used to delete RAID volumes. This subcommand requires one of the options shown in the following table.

Short Option	Long Option	Description
-r	--raid	Deletes the volume listed by ID number.
N/A	--all	Deletes all RAID volumes on all controllers. RAIDconfig queries the storage management library to determine if the RAID disks have been mounted. If so, it generates a warning message to the user and queries the user to delete the RAID volume.

How To Delete a RAID Volume

- **To delete a RAID volume, type the following command:**

  raidconfig delete raid *option*

  For example:

  - To delete RAID volume 1 created on controller 1, type the following command:

    ```
    raidconfig delete raid -r c1r1
    ```

  - To delete all RAID volumes, type the following command:

    ```
    raidconfig delete raid --all
    ```

See Also

- "delete raid Subcommand" on page 71

add disk Subcommand

The add disk subcommand adds a specified disk to a RAID configuration.

Only certain RAID levels, such as RAID 5 or 6, allow for disks to be added to their configuration when in a non-degraded (healthy) state. Only RAID levels that support redundancy allow for disks to be added.

The add disk subcommand requires the options shown in the following table.

Short Option	Long Option	Description
-d	--disks	Specifies the list of disks that you want to add to the RAID volume.
-r	--raid	Specifies the RAID volume ID number that you want to add the disk to.

▼ How to Add a Disk

- **To add a specific disk to a RAID volume, type the following command:**

  raidconfig add disk -d *disk* **-r** *raidvolume*

  For example:

  ```
  raidconfig add disk -d c0d2 -r c0r1
  ```

Note – If you list the disk's properties after adding a disk, the RAID ID is not updated to reflect that it has been added to a RAID volume until the add process is complete.

---

remove disk Subcommand

The `remove disk` subcommand removes a disk from a RAID volume. Only RAID levels that support redundancy allow for disks to be removed. This subcommand requires the options shown in the following table.

Short Option	Long Option	Description
`-d`	`--disks`	Specifies the disk that you want to remove from the RAID volume.
`-r`	`--raid`	Specifies the RAID volume ID you want to remove the disk from.

▼ How to Remove a Disk from a RAID Volume

- **To remove a specific disk from a RAID volume, type the following command:**

  raidconfig remove disk -d *disk* **-r** *raidvolume*

  For example:

  ```
  raidconfig remove disk -d c0d0 -r c0r1
  ```

add spare Subcommand

The `add spare` subcommand adds global or dedicated spare disks:

```
raidconfig add spare
```

The `add spare` subcommand requires one of the options shown in the following table.

Short Option	Long Option	Description
-d	--disks	This mandatory option specifies a list of disk ID numbers, separated by commas. If the -r option is not used, the disks are added as global spares.
-r	--raid	Only used when working with dedicated spares. If a RAID Volume ID is specified, the spares should be added as dedicated spares for this RAID Volume. Note that some controllers do not support dedicated spares and the command might fail.

▼ How to Add a Spare

1. **To create two global spares using the specified disks, type the following command:**

   raidconfig add spare -d *disk*,*disk*

   For example:

   ```
   raidconfig add spare -d c1d0,c1d1
   ```

2. **To create two dedicated spares on a RAID volume using the specified disks, type the following command:**

   raidconfig add spare -d *disk*,*disk* **-r** *raidvolume*

   For example:

   ```
   raidconfig add spare -d c1d0,c1d1 -r c1r0
   ```

See Also

- "add spare Subcommand" on page 72
- "remove spare Subcommand" on page 73

remove spare Subcommand

The remove spare subcommand removes disks as spares or RAID volumes. This subcommand requires the options shown in the following table.

Short Option	Long Option	Description
-d	--disks	Specifies disks to remove. Disk ID numbers are separated by commas. If the -r option is not defined, the disks are removed as global spares.
-r	--raid	If a RAID volume ID is specified, the disks should be removed as dedicated spares from this RAID volume.

See also:

"How to Remove a Spare Disk or a RAID Volume" on page 74

▼ How to Remove a Spare Disk or a RAID Volume

1. **To remove two disks as global spares, type the following command:**

   raidconfig remove spare -d *disk*,*disk*

   For example:

   ```
   raidconfig remove spare -d c1d0,c1d1
   ```

2. **To remove two disks as dedicated spares on a RAID volume, type the following command:**

   raidconfig remove spare -d *disk*,*disk* **-r** *raidvolume*

   For example:

   ```
   raidconfig remove spare -d c1d0,c1d1 -r c1r0
   ```

See Also

- "remove spare Subcommand" on page 73

modify Subcommand

The modify subcommand modifies the attributes of a RAID volume and certain controller attributes. The modify subcommand requires one of the options shown in the following table.

Short Option	Long Option	Description
-r	--raid	Specifies the RAID volume to modify. This is required for the modify raid subcommand.
-c	--controller	Specifies the controller to modify. This is required for the modify controller subcommand.

The following table lists additional options for the modify subcommand.

Option	Description
--raid	Specifies the RAID volume to modify.
--name	Specifies the user-defined name to identify the RAID volume. Can be set to an empty string (").

Option	Description
`--read-cache`	Write cache can be one of the following: `disabled` – Disables RAID read caching `enabled` – Enables RAID read caching
`--write-cache`	Write cache can be one of the following: `disabled` – Disables RAID write caching. `enabled` – Enables RAID write caching. `enabled_protect` – Enables caching only if the battery is available.
`--bios-boot-target=true`	Sets the boot target. When this option is set to true for a specific RAID volume, that RAID volume becomes the BIOS boot target.
`--disable-auto-rebuild=true|false`	Disables auto rebuild. When this option is set to true for a specific controller, auto rebuild will be disabled. If the option is set to false, a hot spare can automatically replace a faulty disk, in which case a long running background task is started.

Note – Not all controllers support modifications of `--read-cache`, `--write-cache`, `--–bios-boot-target` and `--disable-auto-rebuild`.

▼ How to Modify the BIOS Boot Target

The RAID volume with ID 0 is the default boot target. If you want to change the boot target, use the `--bios-boot-target` option.

- **To change the bios boot target, type the following command:**

  `raidconfig modify raid -r` *raidvolume* **`--bios-boot-target=true`**

  For example:

  ```
  raidconfig modify raid -r c0r0 --bios-boot-target=true
  ```

▼ How to Disable Auto Rebuild

When a hotspare disk replaces a faulty disk, it will start autobuilding the volume to use the hotspare disk if auto rebuild is enabled. If you do not want to start the long-running background task automatically, you can disable this feature.

- **To disable auto rebuild, type the following command:**

  raidconfig modify controller -c *controller id* **--disable-auto-rebuild=true**

  For example:

  raidconfig modify controller -c c0 --disable-auto-rebuild=true

▼ How to Modify a RAID Volume Name

To modify the user-specified name of a RAID volume:

- **To change the user-specified name of a RAID volume, type the following command:**

  raidconfig modify raid -r *raidvolume* **--name** *name*

  For example:

  raidconfig modify raid -r c0r0 --name engineering

See Also

- "create raid Subcommand" on page 70
- "delete raid Subcommand" on page 71

start task and stop task Subcommands

The start task and stop task subcommands control the execution of maintenance tasks on a disk or RAID volume. The syntax of the commands are as follows:

raidconfig start task -t *taskname* **[-d|-r]**

raidconfig stop task -t *taskname* **[-d|-r]**

The available background tasks are shown in the following table.

Task	Description
verify	Checks the validity of the RAID volume redundant data.
init	Initializes the RAID volume to write out the initial parity values. The initialization goes over the entire volume and initializes the parity data.
copy	Copies and moves an online physical disk onto a hotspare or unconfigured good drive. The copy is performed while the volume is online. Once completed, the destination disk is added to the logical volume configuration while the original source disk is removed from it.

Task	Description
`rebuild`	Regenerates the data of a single physical disk that is part of a logical volume with data redundancy. The physical disk is reconstructed from another physical disk and/or parity disks. A disk rebuild typically occurs after a disk replacement or repair.
`clear`	Clears a physical disk by writing zeroes over the entire disk.

Note – Not all devices support all tasks. To check the tasks a device supports, use the list subcommand and check the output under Startable tasks. If this field is blank, the device does not support any tasks.

The `start task` and `stop task` subcommands accept the options shown in the following table.

Short Option	Long Option	Description
`-t`	`--task`	Specifies the type of task to execute. Possible options are `verify`, `init`, `rebuild`, `clear`, or `copy`.
`-d`	`--disk`	Specifies the disk to execute the task on. Required by the `rebuild` and `clear` tasks.
`-r`	`--raid`	Specifies the RAID volume to execute the task on. Required by the `verify` and `init` tasks.
n/a	`--src-disk`	Specifies the source disk to use in a `copy` task.
n/a	`--dst-disk`	Specifies the destination disk to use in `copy` task.

Start and Stop Task Examples

The following are command examples for the `start task` and `stop task` subcommands:

- A RAID ID must be provided for the verify check (`verify`) and initialization task (`init`).
  - To start the `verify` task on a specified RAID volume, type the following command:

    `raidconfig start task -t verify -r=`*raidvolume*

    For example:

    `raidconfig start task -t verify -r=c0r1`
  - To stop the `init` task on a specified RAID volume, type the following command:

    `raidconfig stop task -t init -r=`*raidvolume*

    For example:

    `raidconfig stop task -t init -r=c0r1`

- A disk must be provided for the rebuild and clear tasks.
  - To start the `rebuild` task on a specified disk, type the following command:

    `raidconfig start task -t rebuild -d=`*disk*

    For example:

    `raidconfig start task -t rebuild -d=c0d1`

    Note – This can only be run on a disk that is in a RAID.

  - To start the `clear` task on a specified disk, type the following command:

    `raidconfig start task -t clear -d=`*disk*

    For example:

    `raidconfig start task -t clear -d=c0d1`

    Note – This can only be run on a disk that is not in a RAID.

- Source and destination disks must be provided for the copy task.

  To start the `copy` task from one disk to another, type the following command:

  `raidconfig start -task -t copy --src-disk=`*source_disk*
  `--dst-disk=`*destination_disk*

  For example:

  `raidconfig start -task -t copy --src-disk=c0d2 --dst-disk=c0d3`

  Note – The source disk must be in a RAID volume, the destination disk cannot be in a RAID volume.

`restore config` and `clear config` Subcommands

When you make changes to the configuration of a controller, the controller saves its configuration on one of the disks. The `restore config` subcommand enables you to load this configuration from a disk. The `clear config` subcommand enables you to reset a controller's configuration. Both commands only work if the configuration has been saved to a disk.

The `restore config` and `clear config` subcommands require the options shown in the following table.

Short Option	Long Option	Description
-c	--controller	Specifies the controller to use in the operation.

This section covers the following topics:

- "How to Restore a RAID Controller Configuration" on page 79
- "How to Clear a RAID Controller Configuration" on page 80

How to Check a Controller Configuration Exists

1. **To determine if an old configuration exists on the disks, type the following command to view the controller's verbose properties:**

   raidconfig list controller -v

   The controller's properties are listed.

2. **View the property Disks In Use by Another Controller.**

   a. **If the Disks In Use by Another Controller property is set to True, then an old configuration exists. This can be either restored or cleared.**

   b. **If the Disks In Use by Another Controller property is set to False, then an old configuration does not exist.**

Note – If an old configuration does not exist and you attempt to run the restore config or clear config subcommands, raidconfig displays an error.

See Also

- "How to Restore a RAID Controller Configuration" on page 79
- "How to Clear a RAID Controller Configuration" on page 80

How to Restore a RAID Controller Configuration

The restore config subcommand finds a RAID configuration stored on disks and restores this configuration to the destination controller.

1. **To restore a RAID configuration saved on disks to a defined controller, type the following command:**

   raidconfig restore config -c=*controller_id*

   where *controller_id* is the controller the RAID configuration is restored to.

2 **If you have a Linux system, reboot to complete the restore operation.**

See Also

- "restore config and clear config Subcommands" on page 78

How to Clear a RAID Controller Configuration

The clear command finds a RAID configuration stored on disks and removes this configuration.

- **To clear a RAID configuration saved on disks, type the following command:**

  raidconfig clear config -c=*controller_id*

  where *controller_id* is the controller the RAID configuration is cleared from.

See Also

- "restore config and clear config Subcommands" on page 78

export Subcommand

The export subcommand writes XML-formatted configuration or inventory data to a file. Inventory data is a snapshot of all the fields for the controllers, RAID volumes, and disks. Configuration data contains only attributes that can be set and imported onto another system to configure that system's RAID volumes in the same manner.

The export subcommand requires a file name as a modifier. If a file by that name exists, the tool prompts to overwrite the file (unless the -y option is used). If the hyphen ("-") is given for the filename, then the XML-formatted configuration is written to the screen.

The export subcommand can be used to write the inventory or configuration to an XML file.

This subcommand requires at least one of the types shown in the following table.

Option	Description
inventory	Exports and writes all controller, RAID volume, and physical disk information to an XML file.
config	Exports and writes only configuration fields to an XML file in a format that can be imported.

▼ How to Export Inventory Data to a File

- **To export the inventory or a configuration and write it to a file:**
  - **To export the inventory data and write it to a file, type the following command:**
    `raidconfig export inventory` *filename*`.xml`
  - **To export a configuration and write it to a file, type the following command:**
    `raidconfig export config` *filename*`.xml`

See Also
- "`import` Subcommand" on page 81

`import` Subcommand

The `import` subcommand reads an XML-formatted configuration file and configures RAID volumes based on the file. If the creation of a specific RAID volume fails, the error is logged and the next RAID volume in the file is created. The `import` subcommand requires the `config` type and a file name as the operand (modifier).

▼ How to Configure RAID Volumes from a File

- **To configure the RAID volumes according to a configuration file, type the following command:**
  `raidconfig import config` *filename*`.xml`

See Also
"`export` Subcommand" on page 80

Creating RAID Volumes With Partial Disks

Starting with Oracle HMP 2.2.6, a new `--subdisk-size` option is available for the `raidconfig create` command to define the size of RAID volumes. This option is used to define the size of the partial disks to be used in a RAID volume.

The syntax of the `raidconfig create` command with the `--subdisk-size` option is as follows:

`raidconfig create raid --disk=`*disks* **`--subdisk-size=`*sizes*

For example, the following command will create three RAID volumes with subdisks within disks c0d0 and c0d2 sized at 50, 100, and 200 GB :

```
# raidconfig create raid --disk=c0d0,c0d1 --subdisk-size=50,75,100
Create RAID level 0 volumes using disk sizes 50, 75, 100 from the
following disk(s):
```

```
Disk c0d0 (controller 0 slot 0)
Disk c0d1 (controller 0 slot 1)  [y/n]? y
RAID created successfully
```

If the `--subdisk-size` option is not used, the `raidconfig create` command will create a single RAID volume from the defined disks.

Guidelines for Using the RAID Volume Size Option

Keep the following guidelines in mind when using the RAID volume `--subdisk-size`:

- The total sizes for the RAID volumes designated in the `--subdisk-size` option cannot exceed the available size for any of the disks. The total size can be less than or equal to the disk size, but it cannot be larger.
- You cannot create a RAID volume using a partial disk on a disk that is configured as part of a RAID volume. Once a disk has been included in a RAID volume, the disk is marked as "In Use" and cannot be used to create another RAID volume, even if just a part of the disk is used.

  For example, the following sequence of commands is not allowed:

  ```
  # raidconfig create raid --disk=c0d0,c0d2 --subdisk-size=50

  # raidconfig create raid --disk=c0d0,c0d2 --subdisk-size=100
  ```

  The second command will result in an error.
- When creating multiple RAID volumes at the same time using the `--subdisk-size` option, all of the RAID volumes are configured with the same name if the `--name` option is used.

  If this occurs, you can rename the volumes using the `raidconfig modify` command.

Related Information

- "`modify` Subcommand" on page 74

Disk Display

The `raidconfig list all` command output has changed to indicate that a disk is part of more than one RAID volume. A row is added for each disk/raid combination under the `DISKS In Use` list.

The `Size` column shows the size of the subdisk used to create the RAID volume.

The following is an example of the `Disks In Use` output:

```
DISKS In Use
============
ID      Chassis   Slot   RAID ID   Status    Type   Media   Spare      Size (GiB)
```

```
--------------------------------------------------------------------------------
c0d0    0         17     c0r0      OK        sas    HDD     -          50
c0d0    0         17     c0r1      OK        sas    HDD     -          100
c0d0    0         17     c0r2      OK        sas    HDD     -          200
c0d2    0         18     c0r0      OK        sas    HDD     -          50
c0d2    0         18     c0r1      OK        sas    HDD     -          100
c0d2    0         18     c0r2      OK        sas    HDD     -          200
```

Deletion of RAID Volumes With Partial Disks

You can delete a RAID volume on a partial disk, but if the partial disk is used in another RAID volume, the disk will be marked as "In Use". You will not be able to create another RAID volume using that disk.

Exporting a RAID Configuration Including a RAID Volume With Partial Disks

If a RAID volume was created using partial disks, `raidconfig` stores the size of the sub-disk in the XML output generated by the export command. An example of a disk property is shown below:

```
   <disk>
<chassis_id>0</chassis_id>
<slot_id>1</slot_id>
<subdisk_size>100</subdisk_size>
</disk>
```

Adding or Removing a Partial Disk

The `raidconfig add` and `raidconfig remove` features are supported for partial disks. If a disk contains multiple RAID volumes, it can be added and removed.

Note – When the disk supports multiple RAID volumes, only use the first RAID volume in the `add` and `remove` commands.

An example of removing a disk is shown below:

```
raidconfig remove disk -r=c0r4 -d=c0d0

Removing the following disk(s) from RAID c0r4:
Disk c0d0 (controller 0 slot 0)  [y/n]? y
Successfully removed disk from RAID

raidconfig list all
```

```
CONTROLLER c0
=============
Manufacturer   Model     F/W Version    RAID Volumes   Disks
-------------------------------------------------------------
LSI Logic      0x0079    2.130.353-1803 6              7

RAID Volumes
============
ID     Name            Device            Status    Num Disks Level   Size (GiB)
-------------------------------------------------------------------------------
c0r0   0               c3t0d0p0          OK        1         0       558
c0r1                   c3t1d0p0          OK        1         0       278
c0r2                   c3t2d0p0          OK        1         0       136
c0r3                   c3t3d0p0          OK        1         0       70
c0r4                   c3t4d0p0          DEGRADED  2         1       50
c0r5                   c3t5d0p0          DEGRADED  2         1       100

DISKS In Use
============
ID      Chassis   Slot   RAID ID   Status    Type   Media Spare      Size (GiB)
-------------------------------------------------------------------------------
c0d1    0         1      c0r4      OK        sas    HDD    -          50
c0d1    0         1      c0r5      OK        sas    HDD    -          100
c0d3    0         3      c0r0      OK        sas    HDD    -          558
c0d4    0         4      c0r1      OK        sas    HDD    -          278
c0d5    0         6      c0r3      OK        sas    HDD    -          70
c0d6    0         7      c0r2      OK        sas    HDD    -          136

DISKS Available
===============
ID      Chassis   Slot   RAID ID   Status    Type   Media Spare      Size (GiB)
-------------------------------------------------------------------------------
c0d0    0         0      -         OK        sas    HDD   -          279
c0d2    0         2      -         OK        sas    HDD   -          279
```

The following is an example of adding a disk:

```
raidconfig add disk -r=c0r4 -d=c0d2

Adding the following disk(s) to RAID c0r4:
Disk c0d2 (controller 0 slot 2)  [y/n]? y
Successfully added disk to RAID

raidconfig list all

CONTROLLER c0
=============
Manufacturer   Model     F/W Version    RAID Volumes   Disks
-------------------------------------------------------------
LSI Logic      0x0079    2.130.353-1803 6              7

RAID Volumes
============
ID     Name            Device            Status    Num Disks Level   Size (GiB)
-------------------------------------------------------------------------------
c0r0   0               c3t0d0p0          OK        1    0            558
c0r1                   c3t1d0p0          OK        1    0            278
```

```
c0r2                    c3t2d0p0          OK        1    0            136
c0r3                    c3t3d0p0          OK        1    0            70
c0r4                    c3t4d0p0          DEGRADED  2    1            50
c0r5                    c3t5d0p0          DEGRADED  2    1            100

DISKS In Use
============
ID      Chassis   Slot   RAID ID   Status    Type   Media Spare      Size (GiB)
--------------------------------------------------------------------------------
c0d1    0         1      c0r4      OK        sas    HDD   -          50
c0d1    0         1      c0r5      OK        sas    HDD   -          100
c0d2    0         2      c0r4      INIT      sas    HDD   -          50
c0d2    0         2      c0r5      INIT      sas    HDD   -          100
c0d3    0         3      c0r0      OK        sas    HDD   -          558
c0d4    0         4      c0r1      OK        sas    HDD   -          278
c0d5    0         6      c0r3      OK        sas    HDD   -          70
c0d6    0         7      c0r2      OK        sas    HDD   -          136

DISKS Available
===============
ID      Chassis   Slot   RAID ID   Status    Type   Media Spare      Size (GiB)
--------------------------------------------------------------------------------
c0d0    0         0      -         OK        sas    HDD   -          279
```

Using the `ilomconfig` Tool

`ilomconfig` allows you to configure Oracle ILOM service processors from the host OS without having to connect to the management network. You can target `ilomconfig` changes to either the local or a remote Oracle ILOM service processor.

`ilomconfig` also functions as an *XML builder* by either exporting the configuration of an Oracle ILOM service processor to an existing XML file, or creating a new XML file. These XML files can then be used for a subsequent restore operation on compatible Oracle ILOM service processors.

You can also use `ilomconfig` to configure a Host-to–ILOM Interconnect on platforms that support this configuration.

Note – There are some limitations to using `ilomconfig` tool for SPARC M5–32 servers. Refer to the Release Notes for more information.

This section includes the following topics:

`ilomconfig` Overview

This section includes the following topics:

`ilomconfig` Features

The `ilomconfig` commands can be directed at a local or remote Oracle ILOM service processor, or an XML configuration file. This file can then be used as a "golden image" to make changes to multiple Oracle ILOM service processors. You can either export the configuration of an Oracle ILOM service processor or create a new XML configuration file.

`ilomconfig` provides the following functions:

- Back up and restores from an Oracle ILOM XML file
- Modify the XML file using sub-commands
- Set the network, including DHCP and sideband
- List and configure identification information, including hostname, contact, location, and description
- List and configure DNS
- List and configure clock including time zone
- List and configure user management
- List and configure SNMP community

Restoring and Modifying Oracle ILOM XML Configuration Files

Starting with Hardware Management Pack 2.1, `ilomconfig` can generate a backup of an Oracle ILOM service processor's configuration to an XML file with the `export config` command. The `create` or `modify` subcommands can be used to create or modify XML files.

By default, `ilomconfig` commands are executed on the local Oracle ILOM service processor. When you use the `--xmlfile=`*config*.xml option, the `ilomconfig` commands operate on the specified XML file.

The `ilomconfig` subcommands can modify already existing settings in the XML file or create new settings.

Note – Ensure that when you create a new setting in an XML file, your target Oracle ILOM service processor supports the setting.

Oracle ILOM settings can be restored from an XML file starting with Oracle ILOM 3.0.12. Oracle ILOM settings that can be restored include:

- SSH private keys
- User SSH keys
- SSL cert
- COD license
- LDAP and AD certificates
- Platform binary data (currently limited to SPARC LDOMS config)
- User passwords
- SNMP users
- LDAP/LDAPSSL/RADIUS passwords

- Servicetag passphrase

See also:

- " XML File Configuration Commands" on page 92
- "`ilomconfig` Error Codes" on page 124

Host-to-ILOM Interconnect

Host-to–ILOM Interconnect is a new feature from Management Pack 2.1 that provides an interface to the host, which allows the host to communicate with Oracle ILOM over a high-speed channel. In some versions of the Oracle ILOM interface, this feature is referred to as "Local Host Interconnect". In some versions of the Oracle Hardware Management Pack interface, this feature is referred to as the "Local ILOM Interconnect".

Note – Your service processor must be running Oracle ILOM 3.0.12.x or later to support this feature.

The assignment of the IP address for the internal USB Ethernet device (host IP address) can be automatically configured when you install the Oracle Hardware Management Pack 2.1 or later. Alternatively, you can configure the host interconnect IP address at any time using the `ilomconfig` tool.

See the following documentation for more information on enabling the Host-to–ILOM Interconnect between the Oracle ILOM service processor and the host OS:

- For further information on this feature and instructions on enabling it during the Oracle Hardware Management Pack installation, see "Enabling the Host-to-ILOM Interconnect" in *Oracle Hardware Management Pack Installation Guide* .
- Refer to the Oracle ILOM Host-to-ILOM Interconnect (Local Host Interface) documentation to enable this feature in Oracle ILOM. This feature must be enabled in Oracle ILOM in order for the Management Pack (or ilomconfig commands) to automatically configure the network connections points between the service processor and the host OS.

  If you choose not to use the Management Pack (or the ilomconfig commands) to automatically configure the Host-to-ILOM Interconnect, you can manually configure the connection points between the Oracle ILOM service processor and the host OS. For more information, see guidelines for setting up the local interconnect in the Oracle ILOM documentation.
- For instructions on manually configuring Host-to-ILOM Interconnect using `ilomconfig` commands, see "Host-to-ILOM Interconnect Configuration Commands" on page 103.

OS-Specific Notes

The following notes relate to running and installing Host-to-ILOM Interconnect on specific operating systems:

- On servers running the Oracle Solaris 11 OS with Oracle ILOM 3.0.12.x or later, the Host-to–ILOM Interconnect is automatically enabled by default during OS installation.
- Oracle VM 3.0.3 does not have the drivers needed to support the Host-to-ILOM Interconnect.

ilomconfig Commands

This section covers the following topics:

ilomconfig Command Usage

The ilomconfig commands must be run in administrator mode.

When a command fails, it returns one of several failure codes listed in "ilomconfig Error Codes" on page 124.

Options

The following table lists the options available to all CLI Tools commands, including ilomconfig.

Short Option	Long Option	Description
-?	--help	Displays help information.
-V	--version	Displays the tool version.
-q	--quiet	Suppresses informational message output and returns only error codes.
-y	--yes	Confirms operation. Does not prompt user for confirmation on the operation when running.

The options listed in the following table apply to using ilomconfig on a remote service processor.

Short Option	Long Options	Description
-H	--remote-hostname	This option is followed by the IP address of the target service processor.
-U	--remote-username	Username used to log in to remote service processor.

You use these options to make changes to a remote service processor, rather than the local service processor. When using these options to access a remote service processor, you must use both options together. For example:

```
ilomconfig list system-summary --remote-hostname=address
--remote-username=username
```

where *address* is the remote Oracle ILOM service processor's hostname or IP address in xx.xx.xx.xx format, and *username* is the user name with access rights to the Oracle ILOM service processor.

Note – The `enable/disable interconnect` and `create/delete credential` subcommands can not be used on remote service processors.

You are prompted for the password for this user name when accessing the remote Oracle ILOM service processor.

Note – The password required by the remote user name can be piped in on stdin for scripting use.

The option listed in the following table applies to using `ilomconfig` on a local XML file.

Long Options	Description
--xmlfile	This option is followed by the path to the file you want to modify.

You use this option to make changes to a local XML file, rather than a local or remote service processor.

Subcommands

The available `ilomconfig` subcommands are listed in the following table.

Subcommand	Description
list	Show Oracle ILOM settings, users, SNMP communities, and system summary.

Subcommand	Description
`create`	Create users and SNMP communities.
`delete`	Delete users and SNMP communities.
`modify`	Modify Oracle ILOM settings.
`import`	Restore Oracle ILOM settings from an XML file.
`export`	Backup Oracle ILOM settings to an XML file.
`reset`	Reset Oracle ILOM to factory defaults.
`enable`	Enable Host-to-ILOM interconnect.
`disable`	Disable Host-to-ILOM interconnect.

See also "CLI Tools Command Syntax and Conventions" on page 11.

XML File Configuration Commands

This section covers the following topics:

▼ How to Export an XML Configuration

To export an entire configuration to an XML file, use the `ilomconfig export config` command.

Note – Exit or close all active ILOM login sessions before proceeding. The `ilomconfig export` file command enables exports the current Oracle ILOM configuration. Before an export operation can be executed, all active open sessions must be closed. There must be no active ILOM sessions logged in to `/SP/console` during the export operation.

In the following examples, *filename*`.xml` represents the file to which you are exporting the ILOM configuration.

- **Do one of the following:**
  - **To export an XML configuration using a passphrase, choose one of the following commands:**
    - **To get a prompt asking whether you want to enter a passphrase:**

      `ilomconfig export config --xmlfile=`*filename*`.xml`

      For example:

      ```
      #ilomconfig export config --xmlfile=config.xml
      Do you want to enter a passphrase to back up sensitive data? [y/n]? y
      Enter passphrase: ********
      Wrote backup of ILOM configuration to 'config.xml'.
      ```

    - **To set up a passphrase to be used with an automated script, provide a passphrase or a file containing the passphrase as follows:**

      `echo` *passphrase* **`| ilomconfig export config --xmlfile=`***filename***`.xml`**

      Where *passphrase* is the passphrase that you want to use.

      or

      `cat` *file_with_passphrase* **`| ilomconfig export config --xmlfile=`***filename***`.xml`**

      Where *file_with_passphrase* is the file containing the passphrase.

      For example:

      ```
      #echo passphrase | ilomconfig export config --xmlfile=config.xml
      Enter passphrase: ********
      Wrote backup of ILOM configuration to 'config.xml'.
      ```

      The passphrase is automatically passed through the command line.

  - **To export an XML configuration without using a passphrase:**

    `ilomconfig export config --xmlfile=`*filename*`.xml -y`

    For example:

    ```
    #ilomconfig export config --xmlfile=config.xml -y
    Wrote backup of ILOM configuration to 'config.xml'.
    ```

    This option exports the ILOM configuration without using a passphrase.

See Also
- "How to Restore Oracle ILOM to Defaults " on page 95
- "How to Import an XML Configuration" on page 94

How to Import an XML Configuration

To import an XML configuration file to configure Oracle ILOM, use the `ilomconfig import config` command. You can also use this command to restore the system configuration by importing a known reliable XML file.

Note – Exit or close all active ILOM login sessions before proceeding. The `ilomconfig import` file command imports the current Oracle ILOM configuration. Before an import operation can be executed, all active open sessions must be closed. There must be no active ILOM sessions logged in to `/SP/console` during the import operation.

In the following examples, *filename*`.xml` represents the file to which you are importing the ILOM configuration.

- **Do one of the following:**
  - **To import an XML configuration using a passphrase, choose one of the following commands.**
    - **To get a prompt asking whether you want to enter a passphrase:**

      `ilomconfig import config --xmlfile=`*filename***`.xml`**

      For example:

      ```
      #ilomconfig import config --xmlfile=config.xml
      Are you sure you want to import the settings from the XML file to ILOM? [y/n]? y
      Do you want to enter a passphrase to restore sensitive data? [y/n]? y
      Enter passphrase: ********
      Preparing to restore XML file to ILOM...
      Done preparing to restore XML file ILOM.
      Restoring configuration (allow several minutes)..............................
      ..............................................................................
      .......Done.
      ```

    - **To set up a passphrase to be used with an automated script, provide a passphrase or a file containing the passphrase as follows:**

      `echo` *passphrase* **`| ilomconfig import config --xmlfile=`***config***`.xml`**

      Where *passphrase* is passphrase that you want to use.

      or

      `cat` *file_with_passphrase* **`| ilomconfig import config --xmlfile=`***config***`.xml`**

      Where *file_with_passphrase* is the file containing the passphrase.

For example:

```
#echo passphrase | ilomconfig import config --xmlfile=config.xml
Enter passphrase: ********
Preparing to restore XML file to ILOM...
Done preparing to restore XML file ILOM.
Restoring configuration (allow several minutes)..............................
.......................................................................
.......Done.
```

The passphrase is automatically passed in through the command line.

- **To import an XML configuration without using a passphrase:**

  ilomconfig export config --xmlfile=*filename***.xml -y**

  For example:

```
#ilomconfig export config --xmlfile=config.xml -y
Preparing to restore XML file to ILOM...
Done preparing to restore XML file ILOM.
Restoring configuration (allow several minutes)..............................
.......................................................................
.......Done.
```

  This option imports the ILOM configuration without using a passphrase.

See Also
- "How to Restore Oracle ILOM to Defaults " on page 95
- "How to Export an XML Configuration" on page 92

How to Restore Oracle ILOM to Defaults

To restore the Oracle ILOM configuration to the factory defaults, use the `ilomconfig reset config` command. Use the `-y` option to bypass the yes or no confirmation prompt. This results in the reboot of the Oracle ILOM.

- **Issue the following command:**

  ilomconfig reset config [*-y***]**

See Also
- "How to Import an XML Configuration" on page 94
- "How to Export an XML Configuration" on page 92

How to List System Summary Information

Use the `ilomconfig list` command to list system summary information including the product name, part number, serial number, Oracle ILOM host name, and Oracle ILOM version information, use the `ilomconfig list system-summary` command. This lists the same information as the the Summary tab in the Oracle ILOM web interface.

Use this subcommand to list information from a local Oracle ILOM, a remote Oracle ILOM, or an Oracle ILOM service processor configuration XML file.

- **Choose one of these procedures depending on where the system summary details are:**
  - **To view the system summary of the local Oracle ILOM service processor, type:**

    `ilomconfig list system-summary`
  - **To view the system summary information from a remote Oracle ILOM service processor, type:**

    `ilomconfig list system-summary --remote-hostname=`*`remote hostname ip`*
    `--remote-username=root`

    For example:

    ```
    ilomconfig list system-summary --remote-hostname=192.168.1.10
    --remote-username=root
    ```

    Oracle ILOM prompts for the root password.

See Also
- "How to Import an XML Configuration" on page 94

How to Create a User

To create a user, use the `ilomconfig create user` command. The `-y` option prevents the yes/no confirmation prompt. If you specify an XML file name, the command modifies the XML file rather than modifying Oracle ILOM.

1. **Issue the following command:**

   `ilomconfig create user` *`username`* **`[--role=`***`role`***`] [--xmlfile=`***`filename`***`.xml]`**

   where *username* is the user to modify, `-role` is the role of the Oracle ILOM user and *filename*.xml is the name of the file to modify.

2. **At the prompt, enter the password for the user.**

See Also
- "How to Delete a User" on page 96

How to Delete a User

To delete a user, use the `ilomconfig delete user` command. The `-y` option prevents the yes or no confirmation prompt. If you specify an XML file name, the command modifies the XML file rather than modifying Oracle ILOM.

- **Issue the following command:**

  `ilomconfig delete user` *`username`***`[-y] [--xmlfile=`***`filename`***`.xml]`**

  where *username* is the user to delete and *filename*.xml is the name of the file to modify.

See Also

- "How to Create a User" on page 96

How to Modify a User Password or Role

To modify a user password or role, use the `ilomconfig modify user` command. When you specify an XML file name, the command modifies the XML file rather than modifying Oracle ILOM.

- **Issue the following command:**

  `ilomconfig modify user` *`username`***`[-p] [--role=`***`role`***`] [--xmlfile=`***`filename`***`.xml]`**

  where *`username`* is the user to modify, `-p` prompts for the user's password, `-role` is the role of the Oracle ILOM user and *`filename`*.xml is the name of the file to modify.

See Also

- "How to List Users" on page 97

How to List Users

To list one or all users, use the `ilomconfig list user` *`username`* command. If *`username`* is specified then only that user is listed. If *`username`* is blank, then all users are listed.

When you specify an XML file name, this command lists users defined in the XML file rather than querying Oracle ILOM.

- **Issue the following command:**

  `ilomconfig list user [`*`username`***`] [--xmlfile=`***`filename`***`.xml]`**

  where *`username`* is the user to list and *`filename`*.xml is the name of the file to modify.

See Also

- "How to Modify a User Password or Role" on page 97

How to List an SNMP Community

To list one or all SNMP communities, use the `ilomconfig snmp-community` command. If an XML filename is specified, the command lists SNMP communities defined in the XML file rather than querying Oracle ILOM.

- **Issue the following command:**

  `ilomconfig list snmp-community [`*`communityname`***`] [--xmlfile=`***`filename`***`.xml]`**

See Also

- "How to Create an SNMP Community" on page 98

How to Create an SNMP Community

To create an SNMP community, use the `ilomconfig create snmp-community` command. When you specify an XML file name, the command modifies the XML file accordingly rather than modifying Oracle ILOM.

- **Issue the following command:**

  `ilomconfig create snmp-community` *communityname***`[--permission=`***ro*|*rw***`] [--xmlfile=`***filename***`.xml]`**

  where `--permission` is either read-only or read-write (*ro*|*rw*).

See Also

- "How to List an SNMP Community" on page 97

How to List IPv4 Network Settings

To list IPv4 network settings, use the `ilomconfig list network` command. This command lists IP address, netmask, gateway, DHCP settings, sideband, and MAC. When you specify an XML file name, this command lists users defined in the XML file rather than querying Oracle ILOM.

- **Issue the following command:**

  `ilomconfig list network [--xmlfile=`*filename***`.xml]`**

See Also

- "How to Modify IPv4 Network Settings" on page 98

How to Modify IPv4 Network Settings

To modify settings, use the `ilomconfig modify network` command. This command modifies IP address, netmask, gateway, DHCP settings, and sideband. When you specify an XML file name, the command modifies the XML file rather than modifying Oracle ILOM.

- **Issue the following command:**

  `ilomconfig modify network [--ipdiscovery=`*static*|*dhcp***`] [--ipaddress=`***ipaddress***`] [--netmask=`***netmask***`] [--gateway=`***gateway***`] [--state=`***enabled*|*disabled***`] [--mgmtport=`***port***`] [--xmlfile=`***filename***`.xml]`**

Option	Description	Example
`--ipdiscovery`	Network discovery mechanism. Can be either static or DHCP.	**`static`** or **`dhcp`**
`--ipaddress`	Oracle ILOM IP address.	**`255.255.255.0`**
`--netmask`	Oracle ILOM netmask.	**`255.255.255.0`**

Option	Description	Example
`--gateway`	Oracle ILOM gateway.	**`255.255.255.0`**
`--state`	Oracle ILOM management port path.	**`/SYS/SP/NET0`** or **`SYS/MB/SP/NETMGM`**
`--mgmtport`	Oracle ILOM management port state.	**`enabled`** or **`disabled`**
`--xmlfile`	Modify specified XML file rather than local Oracle ILOM service processor. Must be followed by = and the pathname to the file.	**`file.xml`**

See Also

- "How to List IPv4 Network Settings" on page 98

How to List IPv6 Network Settings

To list IPv6 network settings, use the `ilomconfig list network-ipv6` command. This command lists IP address, gateway, autoconfig, link local IP address, dynamic IP address and interface state. When you specify an XML file name, this command lists IPv6 network settings defined in the XML file rather than querying Oracle ILOM.

- **Issue the following command:**

  `ilomconfig list network-ipv6 [--xmlfile=`*filename***`.xml]`**

See Also

- "How to Modify IPv6 Network Settings" on page 99

How to Modify IPv6 Network Settings

To modify IPv6 settings, use the `ilomconfig modify network-ipv6` command. This command lists IP address, netmask, gateway, DHCP settings, and sideband. When you specify an XML file name, the command modifies the XML file rather than modifying Oracle ILOM.

- **Issue the following command:**

  `ilomconfig modify network-ipv6 [--static-ipaddress=`*IPv6 address***`]`**
  `[--autoconfig=`*disabled***`|`***statelessdhcpv6_stateful***`|`***dhcpv6_stateless***`]`**
  `[--state=`*enableddisabled***`]|[--xmlfile=`***filename***`.xml]`**

Option	Description	Example
`--static-ipaddress`	Oracle ILOM IPv6 static address.	**`2001:0db0:0000:82a1:0000:0000:1234:abcd`**

Option	Description	Example
`--autoconfig`	Oracle ILOM IPv6 autoconfiguration state.	When using Oracle ILOM 3.0.12.x: **`disabled, stateless_only`** When using Oracle ILOM 3.0.14.x: **`disabled, stateless, dhcpv6_stateful, dhcpv6_stateless`**
`--state`	Oracle ILOM IPv6 administrative state.	**`enabled`** or **`disabled`**
`--xmlfile`	Modify specified XML file rather than local Oracle ILOM service processor. Must be followed by = and the pathname to the file.	**`file.txt`**

See Also

- "How to List IPv6 Network Settings" on page 99

How to List Service Processor Identification Information

To list identification information for the service processor, use the `ilomconfig list identification` command. This command lists service processor host name, system contact, system location, and system description, which is equivalent to the Identification tab on web interface. When you specify an XML file name, the command lists identification information defined in the XML file rather than querying Oracle ILOM.

- **Issue the following command:**

  `ilomconfig list identification [--xmlfile=`*`filename`***`.xml]`**

See Also

- "How to Modify Identification Information" on page 100

How to Modify Identification Information

To modify identification information, use the `ilomconfig modify identification` command. This command modifies the host name, system contact, system location, and system description. When you specify an XML file name, the command modifies the XML file accordingly rather than modifying Oracle ILOM.

- **Issue the following command:**

  `ilomconfig modify identification [--hostname=`*`hostname`***`]`**
  `[--system-contact=`*`system_contact`***`] -[-system-location=`***`system_location`***`]`**
  `[--system-identifier=`*`system_identifier`***`] [--xmlfile=`***`filename`***`.xml]`**

Option	Description	Example
`--hostname`	Oracle ILOM host name.	**`service-processor.domain.com`**
`--system-contact`	Oracle ILOM system contact field.	**`user`**
`--system-location`	Oracle ILOM system location field.	**`west`**
`--system-identifier`	Oracle ILOM system identifier field.	**`x4800`**
`--xmlfile`	Modify specified XML file rather than local Oracle ILOM service processor. Must be followed by = and the pathname to the file.	**`file.xml`**

See Also

- "How to List DNS Information" on page 101

How to List DNS Information

To list DNS information, use the `ilomconfig list dns` command. If an XML file name is specified, the command lists DNS information defined in the XML file rather than querying Oracle ILOM itself.

- **Issue the following command:**

  `ilomconfig list dns [--xmlfile=`*filename***`.xml]`**

See Also

- "How to Modify DNS Information" on page 101

How to Modify DNS Information

To modify DNS information, use the `ilomconfig modify dns` command. If an XML file name is specified, the command modifies the XML file accordingly rather than modifying Oracle ILOM itself.

- **Issue the following command:**

  `ilomconfig modify dns [--nameservers=`*nameserverlist***`] [--autodns=`***enabled***`|`***disabled***`] [--retries=`***retries***`] [--searchpath=`***searchpathlist***`] [--timeout=`***timeout***`] [--xmlfile=`***filename***`.xml]`**

Option	Description	Example
`--nameservers`	List of DNS nameserver IP addresses for Oracle ILOM separated by commas.	**`10.168.1.10`**
`--auto-dns`	Oracle ILOM Auto-DNS state.	**`enabled`** or **`disabled`**
`--searchpath`	List of search suffixes in preferred order and separated by commas.	

Option	Description	Example
`--retries`	Number of retry attempts for DNS.	Integer between 0 and 5.
`--timeout`	Number of seconds to wait for a DNS response. This can be used with up to six search suffixes, each separated by a comma.	**`2`**
`--xmlfile`	Modify specified XML file rather than local Oracle ILOM service processor. Must be followed by = and the pathname to the file.	**`file.xml`**

See Also

- "How to List DNS Information" on page 101

How to List Clock Information

To list clock information, use the `ilomconfig list clock` command. When you specify an XML file name, the command lists clock information defined in the XML file rather than querying Oracle ILOM.

- **Issue the following command:**

  `ilomconfig list clock [--xmlfile=`*filename***`.xml]`**

See Also

- "How to Modify Clock Information" on page 102

How to Modify Clock Information

To modify clock information, use the `ilomconfig modify clock` command. When you specify an XML file name, the command modifies the XML file rather than modifying Oracle ILOM.

- **Issue the following command:**

  `ilomconfig modify clock [--datetime=`*datetime***`] [--timezone=`***timezone***`] [--usentp=`***enabled***`|`***disabled*** **`[-ntp-server1=`***ntpserver1***`] [--ntp-server2=`***ntpserver2***`] [--xmlfile=`***filename***`.xml]`**

Option	Description	Example
`--datetime`	Oracle ILOM date in *MMDDhhmmYYYY* format or *MMDDhhmmYYYY.ss* format.	**`032514272010`**
`--timezone`	Oracle ILOM clock time zone, such as GMT.	**`enabled`** or **`disabled`**
`--usentp`	Oracle ILOM NTP client state.	**`enabled`** or **`disabled`**
`--ntp-server1`	Oracle ILOM NTP server 1 IP address.	*aaa.bbb.ccc.ddd*
`--ntp-server2`	Oracle ILOM NTP server 2 IP address.	*aaa.bbb.ccc.ddd*

Option	Description	Example
`--xmlfile`	Modify specified XML file rather than local Oracle ILOM service processor. Must be followed by = and the pathname to the file.	**`file.xml`**

See Also

- "How to List Clock Information" on page 102

Host-to-ILOM Interconnect Configuration Commands

This section includes the following procedures:

▼ How to Enable the Host-to-ILOM Interconnect

Host-to-ILOM Interconnect (known as Local ILOM Interconnect in the some versions of Installer interface) can be enabled during the Hardware Management Pack installation. Alternatively, you can use the `ilomconfig` tool to enable this feature and manage its properties. See "Enabling the Host-to-ILOM Interconnect" in *Oracle Hardware Management Pack Installation Guide* for a further description of this feature.

Note – It is recommended that you use this command without any arguments and let the command choose the settings. You can override the defaults with different IP and netmask addresses, but this is for advanced users only.

● **Issue the following command:**

`ilomconfig enable interconnect [--ipaddress=`*ipaddress*** `] [--netmask=`***netmask*** `] [--hostipaddress=`***hostipaddress***`]**

Option	Description	Example
`--ipaddress`	Oracle ILOM IP address. This address must be in the format: 169.254.x.x	**`169.254.175.72`**
`--netmask`	Oracle ILOM netmask.	**`255.255.255.0`**
`--hostipaddress`	Host IP address. This address must be in the format: 169.254.x.x	**`169.254.175.73`**

How to Disable the Host-to-ILOM Interconnect

To disable the Host-to-ILOM Interconnect between the host and Oracle ILOM, use the `ilomconfig disable interconnect` command.

- **Issue the following command:**

  `ilomconfig disable interconnect`

How to Modify the Host-to-ILOM Interconnect

To modify the Host-to-ILOM Interconnect between the host and Oracle ILOM, use the `ilomconfig modify interconnect` command. This works only when the interconnect is enabled. At least one option must be specified.

- **Issue the following command:**

  `ilomconfig modify interconnect [--ipaddress=`*ipaddress*` ] [--netmask=`*netmask*` ] [--hostipaddress=`*hostipaddress*`]`

Option	Description	Example
`--ipaddress`	Oracle ILOM IP address. This address must be in the format: 169.254.x.x	**`169.254.175.72`**
`--netmask`	Oracle ILOM netmask.	**`255.255.255.0`**
`--hostipaddress`	Host IP address. This address must be in the format: 169.254.x.x	**`169.254.175.72`**

How to List the Host-to-ILOM Interconnect Settings

To list the interconnect state and IP settings on both the Oracle ILOM and host side of the interconnect, use `ilomconfig list interconnect`.

- **Issue the following command:**

  `ilomconfig list interconnect`

How to Set Up a Credential Cache on the Host

The host local credential cache contains the username and password required to access Oracle ILOM via the Host-to-ILOM Interconnect. The Host-to-ILOM Interconnect can be used by Management Pack utilities that access Oracle ILOM via the LAN-over-USB interface. For more information on the Host-to-ILOM Interconnect, see:

"Host-to-ILOM Interconnect" on page 89

The username and password for the credential cache must match the Oracle ILOM service processor's username and password (for example as shown by `ilomconfig list user`).

Note – The credentials provided must have Administrator privileges, for example root on a Linux system.

- **Issue the following command:**

  `ilomconfig create credential --username=`*`user`*

  where *user* is the username you use to connect to Oracle ILOM.

How to Delete a Credential Cache on the Host

To remove an existing host local credential cache, for example when you have changed the username used to access Oracle ILOM via the Host-to-ILOM Interconnect, delete the host local credential cache.

- **Issue the following command:**

  `ilomconfig delete credential --username=`*`user`*

  where *user* is the username you use to connect to Oracle ILOM.

Using the hwmgmtcli Tool

hwmgmtcli provides a CLI tool to display hardware configuration information and the status of your Oracle servers.

Starting with Oracle Hardware Management Pack version 2.2.4, the open_problems subsystem has been added for hwmgmtcli. This open_problems subsystem allows you to get the current list of all diagnosed faults for the server.

Note – There are some limitations to using hwmgmtcli tool for SPARC M5–32 servers. Refer to the Release Notes for more information.

This section includes the following topics:

hwmgmtcli Command Overview

The hwmgmtcli commands use the following command syntax:

hwmgmtcli *subcommand*

The options listed in the following table apply to all CLI Tools commands, including hwmgmtcli.

Short Option	Long Option	Description
-?	--help	Help—Displays help information.
-V	--version	Version—Displays the tool version.

If you use the --help or --version options, the hwmgmtcli command does not require subcommands, otherwise one or more subcommands are mandatory.

hwmgmtcli supports the subcommands shown in the following table.

Command	Function
`list` *subsystem*	Show details of one or all subsystems.
`export` *subsystem*	Export details of all subsystems to an XML file.

You can choose to show all available information or you can choose a subsystem. The available subsystems are listed in the following table.

Subsystem	Description
all	Show all subsystems available.
server	Show details of server subsystem.
cooling	Show details of cooling subsystem.
processor	Show details of processor subsystem.
memory	Show details of memory subsystem.
power	Show details of power subsystem.
storage	Show details of storage subsystem.
network	Show details of network subsystem.
firmware	Show details of firmware subsystem.
device	Show details of the device subsystem.
bios	Show details of BIOS sub-system.
iomodule	Show details of IO module sub-system.
open_problems	Show all SP diagnosed open problems (ILOM 3.1 or newer).
dcu	Show details of dcu sub-system (only available on multi-domained systems)

The subcommands are discussed below.

The `list` *subsystem* subcommand supports the options listed in the following table.

Short Option	Long Option	Description
`-d`	`--details`	Show all of the properties and components for the subsystem in detail.

The option listed in the following table is supported for the `export` *subsystem* subcommand.

Short Option	Long Option	Description
-f	--filename	Export the subsystem information to *filename*.

See also:

- "list Subcommand" on page 109
- "export Subcommand" on page 110

list Subcommand

The following procedure describes how to use the list *subsystem* subcommand to display the current hardware configuration and status information of a server and its subsystems.

How to List Subsystem Information

- **Issue the following command:**

  hwmgmtcli list *subsystem*

  where *subsystem* is one of the subsystems listed in "hwmgmtcli Command Overview" on page 107.

  The current subsystem information is listed.

How to View Open Problems

The open_problems subsystem is available for Oracle Hardware Management 2.2.4 and later.

- **To view open server problems, type the following command:**

  hwmgmtcli list open_problems

  The following display shows sample output from this command for Oracle HMP 2.2.4 and 2.2.5:

```
Date/Time                 Subsystems          Component
------------------------  ------------------  ------------
Fri Apr 27 13:14:46 2012  Cooling             FM3 (Fan Module 3)
Fan tachometer speed is below its normal operating range. (Probability:
100, UUID: 6f6ef474-5059-c11e-98db-b52f6ffbffa6, Part Number: N/A, Seria
l Number: N/A, Reference Document: http://www.sun.com/msg/SPX86-8000-33)
Fri Apr 27 14:03:56 2012  Processors          P1 (CPU 1)
An integrated I/O (IIO) hot-plug I/O extender port error has occurred. (
Probability: 100, UUID: e2dadf98-24e0-6058-ce74-b87844c4b894, Part Numbe
r: 060D, Serial Number: N/A, Reference Document: http://www.sun.com/msg/
SPX86-8003-PD)
Mon May 28 07:08:48 2012  System              MB (Motherboard)
Power to server is not available due to a malfunctioning component detec
ted by CPLD. (Probability: 100, UUID: ee437083-990b-eb6c-8665-8d952319ab
```

```
1d, Part Number: 7024015-01, Serial Number: 489089M+1135PR00CX, Referenc
e Document: http://www.sun.com/msg/SPX86-8002-2J)
Fri Jun  8 05:27:42 2012  Processors          P0 (CPU 0)
An integrated I/O (IIO) hot-plug I/O extender port error has occurred. (
Probability: 100, UUID: ae3b229a-57f6-67f3-8d3f-fcb7f350fa0d, Part Numbe
r: 060D, Serial Number: N/A, Reference Document: http://www.sun.com/msg/
SPX86-8003-PD)
```

The following display shows sample output from this command for Oracle HMP 2.2.6 and later:

```
=== open_problems report ===
Open Problem 1
Problem time        : Thu Feb 14 22:38:19 2013
Problem subsystem   : System
Problem location    : /SYS (Host System)
Problem description : The top cover of server was opened while AC
input was still applied to the power supplies. (Probability: 100, UUID:
8bb87e70-d210-632b-d553-fc1450105bc4, Part Number: 31112054+1+1, Serial
Number: 1242FML0UV, Reference Document: http://www.sun.com/msg/SPX86-8003-8C).
Open Problem 2
Problem time        : Fri Feb 15 10:37:48 2013
Problem subsystem   : Storage
Problem location    : /SYS/DBP0/HDD2
Problem description : The disk temperature has exceeded the critical
limit. (Probability: 100, UUID: N/A, Part Number: H106030SDSUN300G, Serial
Number: 001234NTR1KD        PWGTR1KD, Reference Document: N/A)
```

`export` Subcommand

The following procedure describes how to use the `export` *subsystem* subcommand to save the current hardware configuration and status information of a server and its subsystems to an XML file.

Note – The only subsystem available for the `export` subcommand is `all`.

▼ How to Export Subsystem Information

- **Issue the following command:**

  `hwmgmtcli export all --filename` *file*`.xml`

  where *file.xml* is the file to which you want to export the system or subsystem information.

  The current information is exported as XML to the specified *file.xml*.

Using the `zoningcli` Tool

`zoningcli` provides a CLI tool for configuring zoning on Oracle SPARC T3-1 servers with the following qualifications:

- Runs Oracle Solaris OS
- Has a 16–disk backplane (SAS-2 expander)
- If your Oracle SPARC T3-1 server is using the two on-board hard disk controllers, the 16–disk backplane *must* be zoned.
- If the server is using the SGX-SAS6-R-INT-Z SAS2 internal PCIe RAID HBA, the 16 disk backplane must *not* be zoned.

`zoningcli` can split zoning into two separate zones:

- Zone A (group 10) has the first 8 disks (PHY 0 - 7) and the first controller (PHY 20 -23)
- Zone B (group 11) has the last 8 disks (PHY 8 - 15) and the second controller (PHY 16-19)

Before attempting to change the server's zoning state, read the product notes pertaining to the 16 disk backplane.

This section covers the following topics:

`zoningcli` Command Overview

The `zoningcli` commands use the following command syntax:

`zoningcli` *subcommand*

The options listed in the following table apply to all CLI Tools commands, including `zoningcli`.

Short Option	Long Option	Description
-?	--help	Help—Displays help information.
-V	--version	Version—Displays the tool version.

If you use the --help or --version options, the zoningcli command does not require subcommands, otherwise one or more subcommands are mandatory.

zoningcli supports the commands listed in the following table.

Command	Function
enable	Enable the zoning.
disable	Disable the zoning.
list	Show the current zoning information.

list expander Subcommand

The following procedure describes how to use the list subcommand to check the current zoning information. This enables you to see if zoning is currently enabled or disabled.

▼ How to List Zoning Information

- **Issue the following command:**

  ```
  zoningcli list expander
  ```

  The current zoning information is listed.

enable zoning and disable zoning Subcommands

The following procedure describes how to use the enable zoning and disable zoning subcommands to control zoning. Zoning is set up by default.

The following restriction applies when using the zoningcli command:

- You must have root permission to run zoningcli.

▼ How to Enable and Disable Zoning

1. **To enable zoning, type the following command:**

   ```
   zoningcli enable zoning
   ```

2 **To disable zoning, type the following command:**

```
zoningcli disable zoning
```

Using `ipmitool` for Windows

This section describes `ipmitool` and how to install `ipmitool` on systems running the Windows operating system.

Before beginning the procedures in this section, you must complete the following procedures:

1. Install the Microsoft Windows Server 2003 or Windows Server 2008 operating system.
2. Download `Windows.zip` and extract `InstallPack_`*x_x_x*`.exe`
3. Run `InstallPack_`*x_x_x*`.exe` to install supplemental software on the server

This section covers the following topics:

`ipmitool` Overview

The Oracle Hardware Management Pack IPMI configuration CLI tool (`ipmitool`), is a utility that reads the sensor data repository (SDR) and displays the following information:

- Sensor values
- System event log (SEL)
- Field-replaceable unit (FRU) information
- Inventory information

`ipmitool` also gets and sets LAN configuration parameters, and performs chassis power control operations through the server's service processor.

`ipmitool` is supplemental software that you can install using the server's Tools and Drivers CD or using the `Installpack_`*x_x_x*`.exe` executable file, where the *_x_x_x* number identifies the version of the package (for example, `InstallPack_1_1_4.zip`).

For information about using `ipmitool` with the Management Agents, see "Generating SNMP Traps" in *Oracle Server Management Agents User's Guide*.

Sun IPMI System Management Driver 2.1

The Sun IPMI System Management Driver 2.1 allows communication between the Microsoft Windows host operating system and the ILOM service processor over an internal Keyboard Controller Style (KCS) interface. This driver is required for Microsoft Windows Server 2003 SP2 and earlier.

For later OS versions (including Microsoft Windows Server 2003 R2) the Microsoft-provided IPMI driver included in their Hardware Management Module provides the same functionality.

This driver must be installed before installing ipmitool. Refer to "Installing the Sun IPMI System Management Driver 2.1" in *Oracle Hardware Management Pack Installation Guide* for information on installing the driver.

Configuring Boot Order Using ipmitool

On some platforms, ipmitool offers an alternative to using biosconfig to make persistent changes to the boot order. This interface can specify which is the highest-priority category of boot device. This operation is equivalent to entering BIOS setup and moving an entire category of devices to the top of the boot list (for example, moving all disks to boot before CD-ROMs).

The following `ipmitool` raw commands work like the `ipmitool` chassis `bootdev` commands. During BIOS POST, the BIOS asks the SP for boot flags. The raw commands have an extra bit set (the persistent bit), which causes the BIOS to reorder the boot list and save that order in CMOS. These `ipmitool` commands can also be issued through the host SP Keyboard Controller Style (KCS) interface if you have the IPMI drivers installed on a host Linux system.

Use one of the following commands, depending on which device you want to boot first:

- To boot from PXE first:

  ```
  ipmitool -H ... -U root -P ... raw 0x0 0x8 0x5 0xC0 0x4 0x0 0x0
  ```

  The BIOS boot order changes so that PXE attempts to boot first. Sun Blade X6275 has both IB and Gigabit Ethernet interfaces, and this moves both to the top of the boot list with the InfiniBand PXE first followed by GE if IB fails over. The BIOS setup reflects the change in the boot order.

- To boot from CD/DVD first:

  ```
  ipmitool -H ... -U root -P ... raw 0x0 0x8 0x5 0xC0 0x14 0x0 0x0
  ```

  The BIOS boot order changes so that any CD/DVD attempts to boot first. On Sun Blade X6275 this could be a USB external CD/DVD-ROM drive or a JavaConsole-redirected CD. The BIOS setup reflects the change in the boot order.

- To boot from floppy or any removable media:

  ```
  ipmitool -H ... -U root -P ... raw 0x0 0x8 0x5 0xC0 0x3C 0x0 0x0
  ```

The BIOS boot order changes so that any floppy or removable media (such as USB flash) attempts to boot first. On Sun Blade X6275 this could be a USB flash drive or a JavaConsole-redirected floppy. The BIOS setup reflects the change in the boot order.

- To boot from the hard drive:

  ```
  ipmitool -H ... -U root -P ... raw 0x0 0x8 0x5 0xC0 0x8 0x0 0x0
  ```

  The BIOS boot order changes so that the hard drive attempts to boot first. Sun Blade X6275 has a flash mini-DIMM SATA that boots first. The BIOS setup reflects the change in the boot order.

CLI Tools Error Codes

This section covers the following topics:

Common Error Codes

The following table lists the common command error codes. Each error code has a string associated with it. The error code is printed to the log file and to the `stdout` file.

TABLE 1 Common Error Codes

Code Number	Error Description
0	OK.
1	Invalid option.
2	Invalid subcommand.
3	Subcommand not supported.
4	Invalid device format.
5	Cannot create XML file.
6	Cannot read XML file.
7	Cannot retrieve application data.
8	Internal error.
9	Insufficient memory.
10	Invalid boolean argument.

TABLE 1 Common Error Codes *(Continued)*

Code Number	Error Description
11	Option not supported.
12	Storage library initialization failure.
13	Entered name is too long.
14	Invalid name after subcommand.
15	XML filename required.
16	Invalid argument.
17	Failure writing XML file.
18	Device is busy, command cannot be completed.
19	User terminated by pressing ctrl-c.
20	Insufficient privilege to execute command.
21	One or more arguments are missing.
22	Unsupported XML file. Please see errors.
23	XML parse failure.
24	Cannot find XML file.
25	XML file contains no records.
26	The current directory is not writeable.
27	Invalid type.
28	The prerequisite criteria fails priority requirement.
29	Prerequisite criteria causes forever loop.
30	IPMI timeout. Wait a few sections and try again.
31	Installation issues detected.

See also:

- "`biosconfig` Error Codes" on page 121
- "`raidconfig` Error Codes" on page 121
- "`ilomconfig` Error Codes" on page 124
- "`fwupdate` Error Codes" on page 125

biosconfig Error Codes

The following table lists possible biosconfig errors and the actions to take when they occur.

Error Number(s)	Description
Errors 36-49	Verify that either the Microsoft IPMI driver (2003 R2) or the Sun ISM driver (Pre–2003 R2) is installed correctly. For information about how to install one of these drivers onto your system, refer to your system's Windows OS documentation on updating your specific driver, or go to your systems software download site.
Errors 57-63	Verify that either Microsoft IPMI driver (2003 R2) or Sun ISM driver (Pre 2003 R2) is installed correctly. Verify that only one of these is installed on your system.
Error 64	Execute biosconfig as root on Linux/Solaris or as Administrator on Windows. Do not run more than one instance of biosconfig at the same time. There are no locks in place (for any OS) to allow for multiple simultaneous accesses.

See also:

- "biosconfig Command Overview" on page 18

raidconfig Error Codes

Errors might be returned if you attempt to configure the RAID entry for an unsupported parameter. For example, if the RAID controller does not support the configured RAID level, the CLI displays a user-friendly error string identifying the misconfiguration and returns a matching error code.

The following table lists the error codes and strings specific to this tool.

TABLE 2 raidconfig Error Codes

Code Number	Error Description
100	No controllers available.
101	Controller does not support RAID.
102	No physical disks associated with controller.
103	Invalid controller.
104	Invalid disk.
105	Invalid RAID volume.

TABLE 2 raidconfig Error Codes *(Continued)*

Code Number	Error Description
106	RAID level not supported by controller.
107	Default RAID level not supported.
108	A defined disk is in use.
109	Number of disks exceeds allowed number for this level.
110	Failure retrieving internal data.
111	Number of disks requested exceeds the number of available disks.
112	Cannot define both actual and requested number of disks.
113	Option not supported by controller.
114	Invalid stripe size for controller.
115	Invalid number of subarrays.
116	Cannot retrieve RAID data.
118	RAID creation failure.
119	RAID deletion failure.
120	Disk defined multiple times.
121	Disks must be in the same controller.
122	The maximum number of RAID Volumes has been created.
123	Invalid RAID configuration.
124	The RAID Volume is in use.
125	Incomplete RAID configuration.
126	Failure writing internal data.
127	Command requires disks to be entered.
128	Disk is not a dedicated spare.
129	Disk is not a global spare.
130	Controller does not support dedicated spares.
131	Controller does not support global spares.
132	Command requires disks or RAID volume to be entered.
133	A defined disk is not in a RAID volume.
134	Cannot set both read and write cache in same command.

TABLE 2 raidconfig Error Codes *(Continued)*

Code Number	Error Description
135	Import could not create RAID volumes or spares - disks may be in use.
136	Subarrays option is required for this RAID level.
137	Incomplete command, no options have been supplied.
138	Number of disks requested exceeds the number of available disks with the same capacity.
139	RAID configuration does not have enough disks for the requested RAID level.
140	RAID configuration has too many disks for the requested RAID level.
141	Disk detected as in use by another controller. Use raidconfig restore or clear command.
142	The number of spares exceeds the maximum allowed by controller.
143	This command does not support the number-disks option.
144	Task type is invalid.
145	Task type must be defined.
146	Task type is only valid for disks.
147	Task type is only valid for RAID Volumes.
148	For this task, disk must not be in use.
149	For this task, disk must be in a RAID Volume.
150	Command currently cannot be executed.
151	The source disk must be in a RAID Volume.
152	The destination disk must not be in a RAID Volume.
153	The source and destination cannot be the same disk.
154	No foreign configuration detected for controller.
155	Unable to add disk to RAID Volume.
156	Task cannot be started, make sure task is listed in Startable Tasks.
157	Task cannot be stopped, make sure task is listed in Stoppable Tasks.
158	Invalid command, filename must come before options.
159	All disks must be the same size.
160	Command is not valid for this RAID level.
161	Subdisk sizes must be less than disk capacity.
162	Could not restore controller configuration.

See also:

- "Using the raidconfig Tool" on page 63

ilomconfig Error Codes

The following table lists the ilomconfig error codes:

TABLE 3 ilomconfig Error Codes

Code Number	Error Description
50	Cannot connect to BMC interface.
51	Missing -username option.
52	Missing -password option.
53	User already exists.
54	Missing -communityname option.
55	Specified community already exists.
56	User does not exist.
57	Community name does not exist.
58	Delete failed.
59	Failures occurred during restore.
60	Must specify option to modify.
61	No such property.
62	Invalid user name length.
63	Invalid role value.
64	Invalid permission value.
65	Invalid password length.
66	Invalid IP discovery value.
67	Invalid IP state value.
68	Invalid IP address.
69	Invalid auto DNS value.
70	Invalid Use NTP value.
71	Product serial number does not match current system.

TABLE 3 ilomconfig Error Codes *(Continued)*

Code Number	Error Description
72	Oracle ILOM error occurred.
73	Cannot modify interconnect when disabled (use enable command).
74	ILOM not reachable over internal LAN.
75	Credential Failure.
76	Cannot manage interconnect when hostmanaged is set to false.
77	Could not connect to remote SP by LAN with supplied credentials.
78	Specified Command can not be used with a remote connection.
79	Oracle ILOM version does not support LAN over USB.
80	ILOM Interconnect required for fault forwarding.
81	SNMP timeout occured while setting up fault forwarding.
82	Failed to configure ILOM SNMP correctly.
83	Service Processor has conflicting configuration. Refer to release notes for resolution.

See also:

- "Common Error Codes" on page 119
- "Using the ilomconfig Tool" on page 87

fwupdate Error Codes

The following table lists the fwupdate command error codes.

You can also list the error codes using the fwupdate list error-codes command. See "list Subcommand" on page 49 for more information.

TABLE 4 fwupdate Error Codes

Code Number	Error Description
200	Invalid device type.
201	Invalid device target type..
202	Invalid device ID, please run \"fwupdate list all\" to verify id.
203	Reset of component failed.
204	Firmware check failed for component.

TABLE 4 fwupdate Error Codes *(Continued)*

Code Number	Error Description
205	Firmware download failed for component.
206	Specified component and specified image type do not match.
207	Must specify an image file name when doing an update.
208	Could not read specified image file.
209	Reset of this component type is not supported.
210	Specified component type does not match devices type.
211	Must specify device to update.
212	Update canceled by user.
213	Firmware version information not available. Reset necessary to activate new firmware.
214	Version verification failed.
215	Final version is being reported the same as the starting version. Update may have succeeded, please check update documentation.
216	Missing or corrupt firmware file referenced by firmware metadata file.
217	Metadata file invalid or corrupt.
218	Metadata error. Prerequisite and priority settings conflict.
219	Power control option is not supported for pre-application.
220	Power control option is not supported for post-application.
221	Power Control option is not supported.
222	Requested component not available.
223	Can't verify version information, no XML provided.
224	Metadata does not include support for this host.
225	Could not identify host type.
226	A valid subcommand required.
227	Invalid option entered.
228	Must specify device to reset.
229	Cannot open file to write XML output.
230	Metadata XML file is required.
231	Invalid priority level entered.

TABLE 4 fwupdate Error Codes *(Continued)*

Code Number	Error Description
232	Cannot read firmware metadata XML file.

See also:

- "Common Error Codes" on page 119
- "Using the fwupdate Tool" on page 45

hwmgmtcli Error Codes

The following table lists the hwmgmtcli command error codes.

TABLE 5 hwmgmtcli Error Codes

Code Number	Error Description
242	Initialize HDL library failure.
243	HDL library command failure.

See also:

- "Common Error Codes" on page 119
- "Using the hwmgmtcli Tool" on page 107

zoningcli Error Codes

The following table lists the zoningcli command error codes.

TABLE 6 zoningcli Error Codes

Code Number	Error Description
250	Storage management library failure.
251	Zoning CLI SMP command failure.
252	Zoning CLI running on a not supported platform.
253	The expander does not have zoning support.

See also:

- "Common Error Codes" on page 119

- "Using the zoningcli Tool" on page 111

Index

D

E

F

H

I

L

M

N

O

P

R

S

T

U

W

X

Z