Sun Server CLI Tools and IPMItool 2.0 User's Guide

Part No: 821–1600–13
November 2010

101116@25097

Contents

Preface

The Sun Server Management documentation provides detailed information about how to install and use Hardware Management Pack and its components.

This preface describes related documentation, submitting feedback to Sun, and a document change history.

Related Books

The following is a list of documents related to single server management for your Sun server. These and additional support documents are available on the web at:

`http://docs.sun.com/app/docs/prod/svrmgmt.pack2`

Document	Description
Sun Server Hardware Management Pack User's Guide	Overview of Sun Server Hardware Management Pack and how to install components
Sun Server Management Agent User's Guide	How to install, configure, and work with Sun Server Management Agents
Sun Server CLI Tools and IPMItool User's Guide	How to install, configure, and work with Sun Server CLI Tools and IPMItool

About This Documentation (PDF and HTML)

This documentation set is available in both PDF and HTML. The information is presented in topic-based format (similar to online help) and therefore does not include chapters, appendixes or section numbering.

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun is not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Feedback.

Change History

The following changes have been made to the documentation set.

- December 2009, initial publication.
- April 2010, updated to reflect changes in version 2.0.
- July 2010, updated to reflect changes in version 2.0.1.

Sun Server CLI Tools Overview

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

For more information, see the Sun Server Hardware Management Pack User's Guide and the Sun Server Hardware Management Agents 2.0 User's Guide.

The downloaded package includes Sun Server Component Manager, which is a cross platform installer that can be used to manage the hardware components.

Hardware Management Pack consists of the following installable components:

- Sun Server Management Agents
  - Hardware Management Agent manages system hardware and storage components.
  - Storage Management Agent sends monitoring information for Storage components to ILOM.
- Sun Server CLI Tools
  - `biosconfig` is a cross operating system CLI tool that enables the user to configure their server's BIOS CMOS settings and host boot order. See "Using the`biosconfig` Tool" on page 29.
  - `fwupdate` is a cross OS tool that enables you to upgrade firmware of any server component, and supports only the firmware update of SAS storage components. For more information, see "Using the `fwupdate` Tool" on page 59.
  - `raidconfig` is a cross OS and cross-vendor CLI tool that enables you to configure RAID volumes. For more information, see "Using the `raidconfig` Tool" on page 67.
  - `ilomconfig` is a cross OS tool that enables the user to manipulate ILOM configurations by way of XML input. For more information, see "Using the `ilomconfig` Tool" on page 81.

For late-breaking issues and information about the CLI Tools, refer to the Release Notes document included with the software download.

See also:

- "CLI Tools Command Syntax and Conventions" on page 25

Installing Components

This section describes how to install and uninstall Hardware Management Pack components on a Sun x86 server using the supplied Sun Server Component Manager. This section contains the following:

Getting Started

The following methods are available for installing the Hardware Management Pack components:

- Use the Sun Server Component Manager in interactive mode.
- Use the Sun Server Component Manager in unattended mode.

Regardless of the installation method you choose, you must carry out the installation as a user with administrative privileges, such as root on Linux or Solaris and Administrator on Windows.

Prerequisites

Different components are supported by different servers and operating systems, so ensure that your target platform is supported by all of the components you intend to install. Before proceeding make sure that you have consulted the supported platforms information available at the following web site:

`http://www.sun.com/systemmanagement/managementpack_supportmatrix.jsp`

Depending on the target server's operating system, you should note the following:

- Oracle Solaris operating system - For the Sun Server Hardware SNMP Plugins to function correctly, you must have System Management Agent (SMA). SMA is installed by default on Solaris. For more information about SMA, see `snmpd`(1M). When installing Hardware Management Pack components, you must be in the global zone. The device `/dev/bmc` must be present on your system for the Hardware Management Agent to function correctly.
- Linux operating system - For the Sun Server Hardware SNMP Plugins to function correctly, you must have Net-SNMP installed. For more information about Net-SNMP, see the `snmpd` documentation. You must also make sure that the KCS IPMI interface between the Sun x86 Server service processor and host operating system is enabled. When using the Hardware Management Agent, you must ensure the root user has read/write access to the IPMI device in order for the Hardware Management Agent to function correctly.
- Windows operating system - For the Sun Server Hardware SNMP Plugins to function correctly, you must have an IPMI device installed and the SNMP service enabled. For more information about the IPMI devices available for your version of Windows, see your Windows product documentation.

Getting the Software

Before you start, make sure that you have downloaded the latest Hardware Management Pack compatible with the operating system on your target Sun server from:

`http://www.sun.com/system-management/os-hw-mgmt`

This file contains the files necessary to install Hardware Management Pack components.

The Hardware Management Pack download file name for the operating systems supported by Hardware Management Pack is as follows:

`sun-ssm-mgmt-pack-`*version-OSVersionNumber*

where *version* is the version of the Hardware Management Pack, and *OSVersionNumber* is the operating system that this Hardware Management Pack is designed for.

Once you download the Hardware Management Pack, you need to uncompress it to a local directory on the Sun x86 server that you want to manage.

Note – On the Solaris operating system, due to the restrictions of `pkgadd`(1M), the path that you uncompress the Hardware Management Pack to must not contain any spaces for the installation process to proceed.

Sun SSM Component Manager Overview

Sun SSM Component Manager (Component Manager) is supplied as part of the Hardware Management Pack. Component Manager enables you to install and uninstall the Hardware Management Pack components, as well as inspect the currently installed and available components. Depending on the operating system you are using there are different methods of working with Component Manager.

On Linux and Solaris operating systems Component Manager is installed, then you can install components. Follow this procedure: "(Linux and Solaris) Using Component Manager" on page 14.

On Windows operating systems Component Manager is run from the download folder and copied to the install directory when at least one component is installed. Follow this procedure: "(Windows) Using Component Manager" on page 19.

Upgrading from Previous Versions

If you have installed a version of Hardware Management Pack prior to version 1.3 on your system, you must manually uninstall the earlier version before installing the latest version. If Component Manager detects these older versions of Hardware Management Pack during the install procedure it does not upgrade the system due to changes in the packaging of releases using Component Manager.

Management Agents 2.0 is not compatible with CLI Tools 1.0, available as part of Hardware Management Pack version 1.3. If Component Manager detects this conflict, you have to approve the upgrade of the CLI Tools component. CLI Tools 2.0 is not compatible with Management Agents 1.3 available as part of Hardware Management Pack version 1.3. When you are using Component Manager in interactive mode and a conflict is detected, you have to approve the upgrade of the Management Agents component. When you are using Component Manager in unattended mode and a conflict is detected, the Management Agents component is automatically upgraded.

In either case, check the supported server matrix at the following web site to ensure that your system is supported by the upgraded component:

`http://www.sun.com/systemmanagement/managementpack_supportmatrix.jsp`

(Linux and Solaris) Using Component Manager

To use the Component Manager on Linux and Solaris operating systems, you must first install the Component Manager. Once the Component Manager is installed, you can choose to install components either interactively using a command-line interface or automatically using command-line switches, which enables unattended installs.

▼ (Linux and Solaris) How to Install Component Manager

Before You Begin You must download and uncompress the Hardware Management Pack on the target server before proceeding. You must carry out the following procedure as a user with root privileges.

1. **Open a terminal.**
2. **Navigate to the directory where you uncompressed the Hardware Management Pack package, and then navigate to the `SOFTWARE` subdirectory .**
3. **Type the following:**

   ```
   ./setup.sh
   ```

   The Component Manager installer starts.
4. **To confirm that you want to install Component Manager, type Y at the following message:**

   ```
   Install the Sun SSM Component Manager? [Y]es, [N]o>
   ```

   Component Manager is installed on to the server at the following path:

   ```
   /usr/sbin/sunssmcompmgr
   ```

   When the installation of Component Manager is finished, the installer asks if you want to automatically start Component Manager in interactive mode.

Next Steps Once you have installed Component Manager, you can choose to use either the interactive mode or unattended mode. For more information see:

- "(Linux and Solaris) Using Component Manager in Interactive Mode" on page 14
- "(Linux and Solaris) Using Component Manager in Unattended Mode" on page 17

(Linux and Solaris) Using Component Manager in Interactive Mode

When you are using Component Manager in interactive mode, you can work with components from a interactive command-line interface.

Note – On Windows operating systems command-line interactive mode is not available, use the Windows graphical installer. See "(Windows) Using Component Manager Command-line Interface" on page 22.

The following table shows the functions available when using the Component Manager interactively.

Option	Functionality
`[L]ist`	Displays the list of currently available components.
`[D]etailed list`	Displays detailed information about the list of currently available components.
`[I]nstall`	Enables you to install some or all of the available components.
`[U]ninstall`	Enables you to uninstall some or all of the currently installed components.
`[H]elp`	Displays information about how to use Component Manager.
`[Q]uit`	Exits Component Manager.

Choose options in the Component Manager by typing the letter shown between the [] characters.

Once you have installed the Component Manager, you can use either the interactive or unattended modes of the Component Manager to work with Hardware Management Pack components. For more information, see:

- "(Linux and Solaris) Using Component Manager in Interactive Mode" on page 14
- "(Linux and Solaris) Using Component Manager in Unattended Mode" on page 17

▼ (Linux and Solaris) How to Install Interactively Using Component Manager

Before You Begin You must install the Component Manager before proceeding. You must carry out the following procedure as a user with root privileges.

Note – If you are upgrading from a previous version, see "Upgrading from Previous Versions" on page 13.

1. **Open a terminal.**
2. **Within the directory where you uncompressed the Hardware Management Pack download, navigate to the `Packages` subdirectory in the `SOFTWARE` subdirectory.**

3 **Start the Component Manager in interactive mode by typing the following command:**

```
/usr/sbin/sunssmcompmgr
```

Component Manager starts and displays a list of currently installed components and available components in the `Packages` subdirectory.

Tip – The option `-d` *directory* specifies a directory that Component Manager searches for available components.

---

4 **To install the components shown in the list of available components, type `I` at the following message:**

```
[L]ist, [D]etailed list, [I]nstall, [U]ninstall, [H]elp or [Q]uit >
```

A numbered list of the available components is displayed.

5 **Choose one of the following options:**

- **To install a single specific component, type the number listed to the right of the component name.**
- **To install all of the listed components, type `A`.**
- **To return to the previous menu, type `R`.**

6 **Depending on the component you chose to install in step 5, you might need to specify further options, such as these:**

- **`Do you wish to start the hwmgmtd service ? [Y]es, [N]o >`**

  To start or restart the named service, type `Y`.

- **`Do you wish to enable the hwmgmtd service on startup by default ? [Y]es, [N]o >`**

  To start the named service each time the server starts, type `Y`.

▼ (Linux and Solaris) How to Uninstall Interactively Using Component Manager

1 **Open a terminal.**

2 **Start the Component Manager in interactive mode by typing the following command:**

```
/usr/sbin/sunssmcompmgr
```

Component Manager starts and displays a list of currently installed components.

3. **To uninstall the components shown in the list of available components, type `U` at the following message:**

   ```
   [L]ist, [D]etailed list, [I]nstall, [U]ninstall, [H]elp or [Q]uit >
   ```

   A numbered list of the available components is displayed.

4. **Choose one of the following options:**

   - **To uninstall a single specific component, type the number listed to the right of the component name.**
   - **To uninstall all of the listed components, type `A`.**
   - **To return to the previous menu, type `R`.**

(Linux and Solaris) Using Component Manager in Unattended Mode

Component Manager provides an unattended mode that enables you to work with Hardware Management Pack components from the command line.

On Linux and Solaris operating systems Component Manager provides the following command line options.

Options and Actions	Functionality
`-h`	Display help on using the Component Manager.
`-v`	Display the Component Manager's version information.
`-d` *directory*	Specify a custom directory for the component packages. The default option is to search for component packages in the current directory.
`-l` *log*	Specify a custom file for logging.
`-s`	Disable service manipulation (start, restart, or stop) during component installation and uninstallation.
`-C`	Print information about both the already installed and available components.
`-D`	Print detailed information about both the already installed and available components.
`-I` *COMPONENT1:COMPONENT2*	Install components. Component names are separated by a colon (:). If "ALL" is given as the component list, all available components are installed.

Options and Actions	Functionality
`-U` *COMPONENT1:COMPONENT2*	Uninstall components. Component names are separated by a colon (:). If "ALL" is given as the component list, all available components are uninstalled.

When using the `-I` or `-U` options to list components to install or uninstall, you should separate the component names using the colon (:) character. Component names are shown when you use the `-C` or `-D` options and are listed in square brackets.

▼ (Linux and Solaris) How to Install Using Component Manager in Unattended Mode

When using the Component Manager in unattended mode, you can install components separately, or you can install all components. Component Manager can provide a list of available components found in the `Packages` subdirectory. You can also configure whether Component Manager automatically starts the services associated with components.

Note – If you are upgrading from a previous version, see "Upgrading from Previous Versions" on page 13.

---

1 Open a terminal.

2 Within the directory where you uncompressed the Hardware Management Pack download, navigate to the `Packages` subdirectory in the `SOFTWARE` subdirectory.

Tip – Use the `-d` *directory* option to pass Component Manager an alternative directory to use for the component packages instead of navigating to the `Packages` subdirectory.

---

3 List the available components by typing the following command:

```
/usr/sbin/sunssmcompmgr -C
```

The list of available components is displayed in the terminal. The exact name of the component to use in the next step is shown in square brackets, for example [*component name*].

4 Choose one of the following options:

- **To install selected components in unattended mode, type the following command:**

  `sunssmcompmgr -I` *COMPONENT1:COMPONENT2*

  where *COMPONENT1:COMPONENT2* is the list of components to install, separated by colons (:).

- **To install all available components in unattended mode, type the following command:**

  ```
  sunssmcompmgr -I ALL
  ```

  The selected components are installed.

▼ (Linux and Solaris) How to Uninstall Using Component Manager in Unattended Mode

1. **Open a terminal.**
2. **List the currently installed components by typing the following command:**

   ```
   /usr/sbin/sunssmcompmgr -C
   ```

   Tip – Use the `-D` option to get detailed information about the currently installed components.

   The currently installed components are listed. The exact name of the component to use in the next step is shown in square brackets, for example [*component name*].
3. **Choose one of the following options:**

   - **To uninstall selected components in unattended mode, type the following command:**

     ```
     /usr/sbin/sunssmcompmgr -U COMPONENT1:COMPONENT2
     ```

     Where *COMPONENT1:COMPONENT2* is the list of components to install, separated by a colon (:) character.

   - **To uninstall all installed components in unattended mode, type the following command:**

     ```
     /usr/sbin/sunssmcompmgr -U ALL
     ```

     The selected components are uninstalled.

(Windows) Using Component Manager

When using the Component Manager on Windows operating systems you can choose between a command-line interface and a graphic user interface. The Component Manager graphic user interface provides an easy-to-use wizard to manage SSM components. The Component Manager command-line interface provides a text-based interface that can be used for unattended deployments.

This section provides the following information:

- "(Windows) Using the Component Manager Graphic Interface" on page 20
- "(Windows) Using Component Manager Command-line Interface" on page 22

(Windows) Using the Component Manager Graphic Interface

The Component Manager graphic interface provides a graphical wizard for working with components. Component Manager provides a list of available components found in the `Packages` subdirectory, as well as any previously installed components. You can also control how Component Manager configures the services associated with components.

▼ (Windows) How to Install Using Component Manager Graphic Interface

When using the graphic interface Component Manager, components can be installed separately or all components can be installed . You can choose whether Component Manager automatically starts or restarts the services associated with components or not.

Note – If you are upgrading from a previous version, see "Upgrading from Previous Versions" on page 13.

---

1. **Navigate to the folder where you downloaded and unpacked the Hardware Management Pack and open the `SOFTWARE` folder.**

2. **Double-click `sunssmcompmgr_gui`.**

   The Component Manager graphic interface opens.

3. **Click Deploy component from the local installation source.**

   Information about the components available to deploy opens.

   On the right side of the screen there is a list of components available to deploy, as well as any components currently installed on the system. The left side of the screen provides information about the currently selected component.

4. **From the Available components to deploy list, select the check box of one or more components you want to install on this server. When you have finished, click Next.**

   The list of pre-deployment tasks opens. Depending on the components you have chosen to install, the list of pre-deployment tasks might be empty or contain optional tasks Component Manager can carry out before you deploy the SSM components.

5. **(Optional) If there are any tasks you want Component Manager to carry out before deploying SSM components, click one or more of the tasks from the list of pre-deployment tasks. Once you have chosen the pre-deployment tasks, click Next.**

   The list of post-deployment tasks opens. Depending on the components you have chosen to install, the list of post-deployment tasks may be empty or contain optional tasks Component Manager can carry out after deploying the SSM components.

6 **(Optional) If there are any tasks you want Component Manager to carry out after deploying SSM components, click one or more of the tasks from the list of post-deployment tasks. Once you have chosen the post-deployment tasks, click Next.**

The Deployment configuration summary opens.

7 **Once you have reviewed the Deployment configuration summary, click Deploy to install the chosen SSM components.**

Component Manager installs and configures the components you selected. Once the installation has finished, a log of the actions taken is displayed.

▼ (Windows) How to Uninstall using Component Manager Graphic Interface

When using the Component Manager graphic interface, you can uninstall components using a graphical wizard. You can also configure whether Component Manager automatically stops the services associated with components or not.

1 **Navigate to the Control Panel and open Add or Remove Programs.**

Tip – You can also open the Component Manager graphic interface by running the `sunssmcompmgr_gui` from the directory where you installed the Hardware Management Pack components.

---

2 **In the list of Currently installed programs, click the component you want to remove.**

The clicked component becomes selected and the Change / Remove button appears.

3 **Click Change / Remove.**

The Component Manager graphical interface opens.

4 **In the Component Manager graphical interface, click Remove deployed component.**

The list of Available components to remove opens.

5 **Select the component you want to remove, and then click Next.**

The list of pre-removal tasks opens. Depending on the components you have chosen to install, the list of pre-removal tasks may be empty or contain optional tasks Component Manager can carry out before removing the SSM components.

6 **(Optional) If there are any tasks you want Component Manager to carry out before removing SSM components, click one or more of the tasks from the list of pre-removal tasks. Once you have chosen the pre-removal tasks, click Next.**

The list of post-removal tasks opens. Depending on the components you have chosen to install, the list of post-removal tasks may be empty or contain optional tasks Component Manager can carry out after removing the SSM components.

7 **(Optional) If there are any tasks you want Component Manager to carry out after removing SSM components, click one or more of the tasks from the list of post-removal tasks. Once you have chosen the post-removal tasks, click Next.**

The Removal configuration summary opens.

8 **Once you have reviewed the Removal configuration summary, click Remove to uninstall the chosen SSM components.**

Component Manager uninstalls and configures the components you selected. Once the uninstallation has finished, a log of the actions taken is displayed.

(Windows) Using Component Manager Command-line Interface

The command-line Component Manager on Windows operating systems provides the same functionality as the unattended mode available on Linux and Solaris operating systems. When passing options and actions to Component Manager at the command line, observe the following conventions:

```
sunssmcompmgr.exe [/h /v /s] [/r dir] [/d dir] [/l log] [ACTION]
```

The following table lists the functionality of the options and actions.

Options and Actions	Functionality
`/h`	Display help about using the Component Manager.
`/v`	Display the Component Manager's version information.
`/d` *directory*	Specify a custom directory for the component packages. The default option is to search for component packages in the current directory.
/l *log*	Specify a custom file for logging.
`/s`	Disable service manipulation (start, restart, or stop) during component installation and uninstallation.
`/C`	Print information about both the already installed and available components.

Options and Actions	Functionality
`/D`	Print detailed information about both the already installed and available components.
`/I` *COMPONENT1 COMPONENT2*	Install components. Component names are separated by a space character. If "ALL" is given as the component list, all available components are installed.
`/U` *COMPONENT1 COMPONENT2*	Uninstall components. Component names are separated by a space character. If "ALL" is given as the component list, all available components are uninstalled.

When using the `/I` or `/U` options to list components to install or uninstall, you should separate the component names using the space character. Component names are shown when you use the `/C` or `/D` options.

▼ (Windows) How to Install Using Component Manager Command-line Interface

When using the command-line Component Manager, you can install components separately or you can install all components. Component Manager can provide a list of available components found in the `Packages` subdirectory. You can also configure whether Component Manager automatically starts the services associated with components or not.

Note – If you are upgrading from a previous version, see "Upgrading from Previous Versions" on page 13.

---

1. **Open the Command Prompt window.**

2. **Within the directory where you uncompressed the Hardware Management Pack download, navigate to the `SOFTWARE` subdirectory.**

Tip – you can use the `/d` *Directory* option to pass component manager an alternative directory to use for the component packages instead of navigating to the `Packages` subdirectory.

---

3. **List the available components by typing the following command:**

   ```
   sunssmcompmgr /C
   ```

   The list of available components is displayed in the Command Prompt window.

4. **Choose one of the following options:**

   - **To install selected components, start the Component Manager in unattended mode by typing the following command:**

     `sunssmcompmgr /I` *COMPONENT1 COMPONENT2*

where *COMPONENT1 COMPONENT2* is the list of components to install, separated by spaces.

Note – if any components are already installed, you must use this method rather than using the ALL flag.

- **To install all available components, start the Component Manager in unattended mode by typing the following command:**

  ```
  sunssmcompmgr /I ALL
  ```

  The selected components are installed.

(Windows) How to Uninstall Using the Component Manager Command-line Interface

When using the Component Manager command-line interface, you can uninstall components separately, or you can uninstall all components. Component Manager provides a list of currently installed components. You can also configure whether Component Manager automatically stops the services associated with components or not.

1. **Open a Command Prompt window.**
2. **Navigate to the directory where you installed the Hardware Management Pack components.**
3. **List the currently installed components by typing the following command:**

   ```
   sunssmcompmgr /C
   ```

   The available and currently installed components are listed.

   Tip – You can also use the `/D` option to get detailed information about the currently installed components.

   ---

4. **Choose one of the following options:**

   - **To uninstall selected components in unattended mode, type the following command:**

     `sunssmcompmgr /U` *COMPONENT1 COMPONENT2*

     where *COMPONENT1 COMPONENT2* is the list of components to uninstall, separated by space characters.

   - **To uninstall all available components in unattended mode, type the following command:**

     ```
     sunssmcompmgr /U ALL
     ```

     The selected components are uninstalled.

CLI Tools Command Syntax and Conventions

This section describes the common meta syntax to be used by all storage management Command-Line Interface (CLI) tools.

CLI Tools Command Syntax

The CLI tools must 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 29 for more information.

The following table describes the command fields:

Command Field	Description	Examples
command	The action that you want to perform. 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`, `sas_bridgefirmware`
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 command options 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 26

CLI Tools Device-Naming Convention

User-friendly, fully qualified device names are used with the CLI 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 a logical ID name, assigned by the tool at initialization. The disks are sorted by expander, and slot ID, to come up with a unique numerical identifier. The numbering is sequential.

Here 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`.

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

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

The following 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 25

Using the `biosconfig` Tool

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

`biosconfig` allows you to manipulate BIOS configurations from the OS command line. Due to the nature of this utility, it must be run with administrative access. You should also close all other applications and quiesce your system when running `biosconfig`. The configuration files and command-line interfaces are compatible with the Solaris, Windows, and Linux versions. Solaris `biosconfig` can be used to configure the BIOS settings.

`biosconfig` enables you to configure settings across multiple like systems where distributions of `biosconfig` spans multiple like systems. `biosconfig` enables you to configure BIOS CMOS settings on like machines using the same XML file. However, if the configuration that is being modified refers to a peripheral or component that is not on both systems, then you need to customize the XML file.

This section shows sample XML configurations and specific configurations needed to change BIOS or BIOS CMOS settings. These XML configurations can be used to make changes to your configuration using `biosconfig`. The XML files are edited by the editor of your choice, such as `vi`.

`biosconfig` is supported on several operating systems on various platforms with common functionality. For more information, see: "Installing Components" on page 11.

This section covers the following topics:

biosconfig Dependencies

You must run biosconfig as root (Linux, Solaris) or Administrator (Windows) because it needs to use drivers that are in read— and write-protected physical address space.

For more on biosconfig for Solaris, see: "biosconfig for Solaris OS" on page 31.

Linux versions of biosconfig also depend on access to /dev/nvram to guarantee serialized access to the CMOS. RHEL4 distributions do not seem to include this device by default, RHEL5 and SLES do. 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).

For information on biosconfig for Windows, see: "biosconfig for Windows" on page 32.

See also:

- "biosconfig Terminology" on page 30
- "Using biosconfig" on page 31
- "biosconfig Command Overview" on page 40

biosconfig Terminology

- BIOS is the software that initializes the computer hardware and then boots the operating system.
- CMOS in this context means the 128 or 256 bytes of battery-backed-up RAM that holds the state that was configured through the BIOS setup menus (or biosconfig).
- IPMI is a standard interface used to manage servers. For more information, go to: http://www.intel.com/design/servers/ipmi
- ipmitool is an open-source tool used to manage a system. ipmitool is distributed on the Tools and Drivers CD for each Sun platform. You can find documentation (for example, on the ipmitool chassis bootdev command) at: http://ipmitool.sourceforge.net/manpage.html.
- NVRAM in this context means the portion of the BIOS ROM that holds the BIOS's boot information.

See also:

- "biosconfig Terminology" on page 30
- "biosconfig Dependencies" on page 30
- "biosconfig Command Overview" on page 40

Using biosconfig

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 editing XML files. The process of editing the BIOS includes using biosconfig to do the following tasks:

1. Run BIOSconfig to create an XML file.
2. Review the XML file and modify it, if necessary, to configure the system to your specific needs.
3. Write the XML file back to the system (or another system) using BIOSconfig.

The following sections describe how to execute a detailed configuration.

Note – In the output examples in this section, 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 44.

For installation instructions on Component Manager, see: "Getting Started" on page 11.

See also:

- "biosconfig Dependencies" on page 30
- "biosconfig Command Overview" on page 40
- "biosconfig Error Messages" on page 98

biosconfig for Solaris OS

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

Solaris biosconfig consists of a Solaris biosdrv driver and the BIOSconfig application.

This section covers the following:

- "How to Obtain the BMC Driver" on page 32

▼ How to Obtain the BMC Driver

Baseboard management controller firmware enables communication over the Intelligent Platform Management Interface (IPMI) controller on the alarm card.

Note – Legal considerations prevent the unrestricted redistribution of the Baseboard Management Controller (BMC) driver on the OpenSolaris OS.

---

1. **Access the `SUNWckr` package from your Solaris builds:**
2. **Copy the following files:**

```
/kernel/drv/amd64/bmc f none 0755 root sys 35984 42138 1225816778 SUNWckr
/kernel/drv/bmc f none 0755 root sys 23684 5173 1225816787 SUNWckr
/kernel/drv/bmc.conf f none 0644 root sys 177 14089 1225787326 SUNWckr
```

3. **Use the network interface.**

   Instead of using `ipmitool-I bmc fru list`, use `ipmitool -I lan -H` *SP's_IP_address* `-U root`.

See Also

- "Using the`biosconfig` Tool" on page 29

`biosconfig` for Windows

`biosconfig` for Windows is a BIOS configuration tool that runs on the host's OS and configures that host's BIOS CMOS settings and host boot order. This tool is supported on several operating systems on various platforms.

With `biosconfig` version 2.2.1 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 required in order to run `biosconfig` on Windows Server 2008 SP2 64–bit or Windows Server 2008 R2 systems. See " Installing and Removing the Sun System Management Driver on Windows 2008 R2 and Windows 2008 64–bit" on page 33.

For a full description of `biosconfig`'s functionality, see "Using the`biosconfig` Tool" on page 29.

Installing and Removing the Sun System Management Driver on Windows 2008 R2 and Windows 2008 64–bit

You must install the Sun System Management Driver on your Windows Server 2008 SP2 64–bit or Windows Server 2008 R2 server so that `biosconfig` can operate. This driver is not required for Windows 2008 32–bit.

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

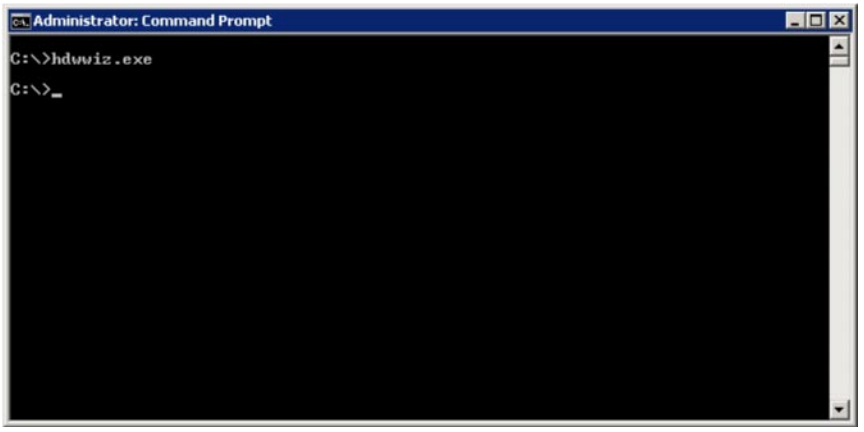
You must install the system management driver by using the Hardware Wizard executable. See "How to Install the `biosconfig` Sun System Management Driver on Windows 2008 R2 and Windows 2008 64–bit" on page 34.

The Sun System Management driver is not needed for other applications. The driver should be uninstalled to free system resources if `biosconfig` is not going be used, but deinstallation is not mandatory. When `biosconfig` is uninstalled, the driver must be uninstalled manually.

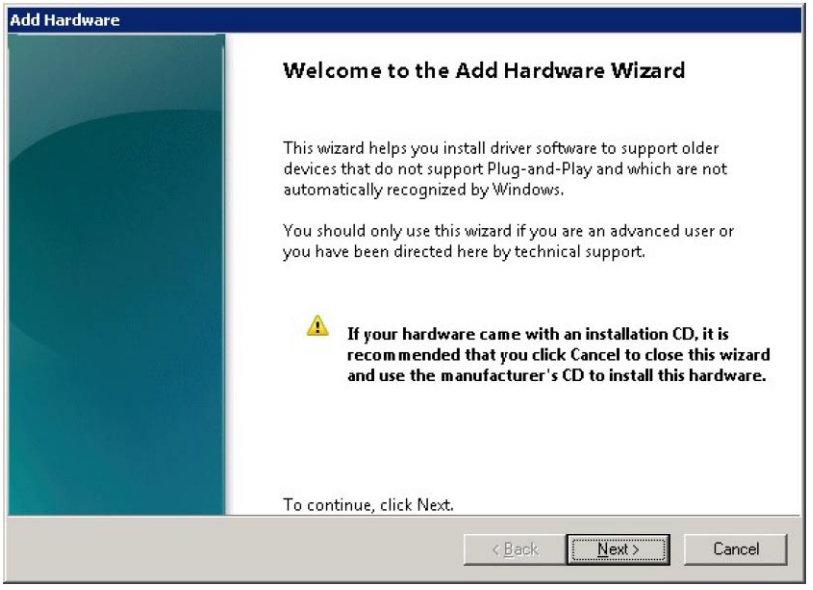
Use the Windows hardware uninstall procedure in the Windows Control Panel to remove the System Management Driver. See "How to Uninstall the `biosconfig` Sun System Management Driver on Windows 2008 R2 and Windows 2008 64–bit" on page 38.

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

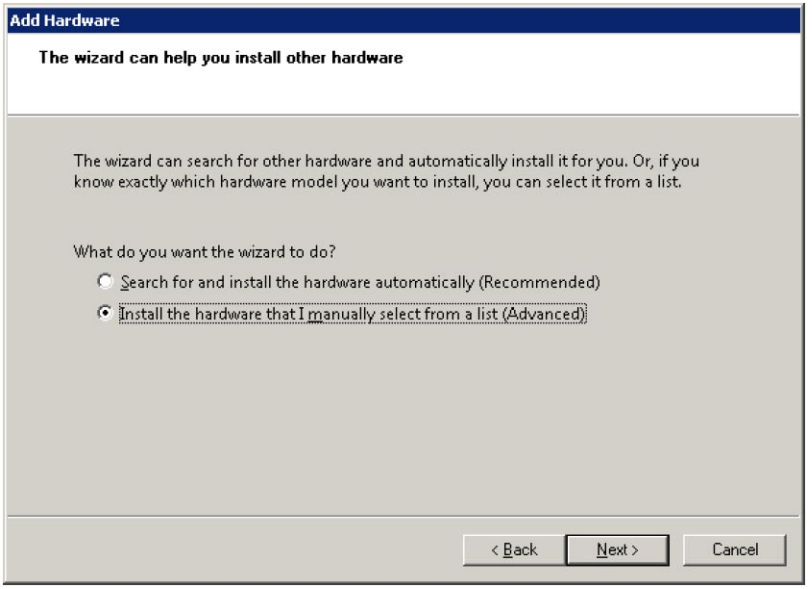
1 **In an Administrator terminal window (`cmd.exe`), run the Hardware Wizard executable file, `hdwiz.exe`.**



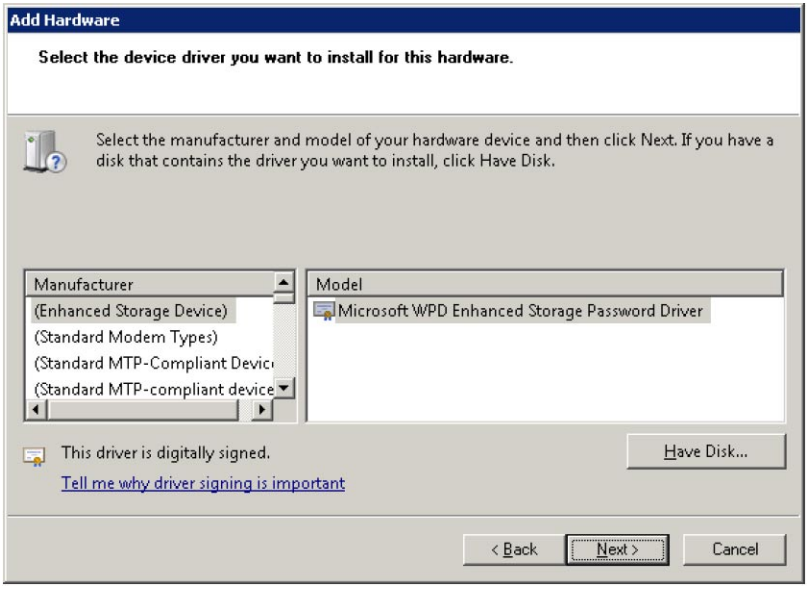
2 **Read the Welcome screen and click Next.**



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

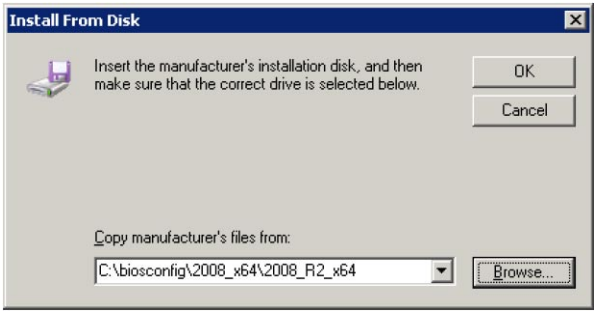


4 Click Have Disk.

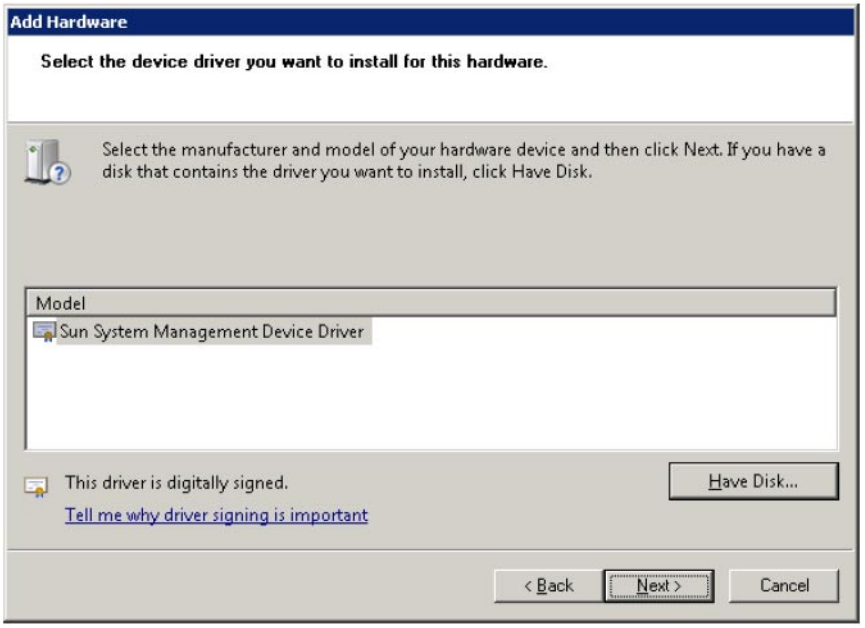


5 Select the path of the driver you want and click OK.

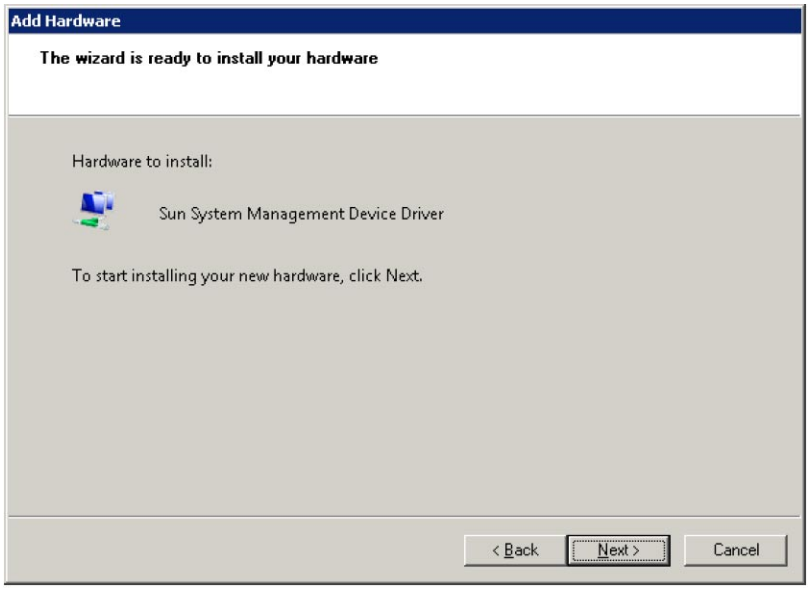
The driver is located in the `2008_`*xx*`2_x64` directory in the `bioscconfig` installation directory. Use the `2008_R2_x64` directory for Windows Server 2008 R2, and the `2008_SP2_x64` directory is for Windows Server 2008 SP2 64–bit.



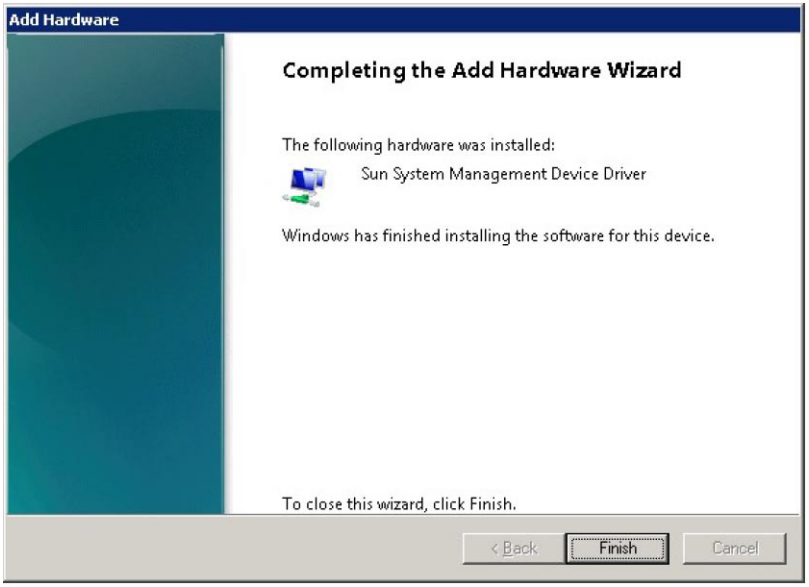
6 Select the Sun System Manager Device Driver, and click Next.



7 **Click Next to start installing.**



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

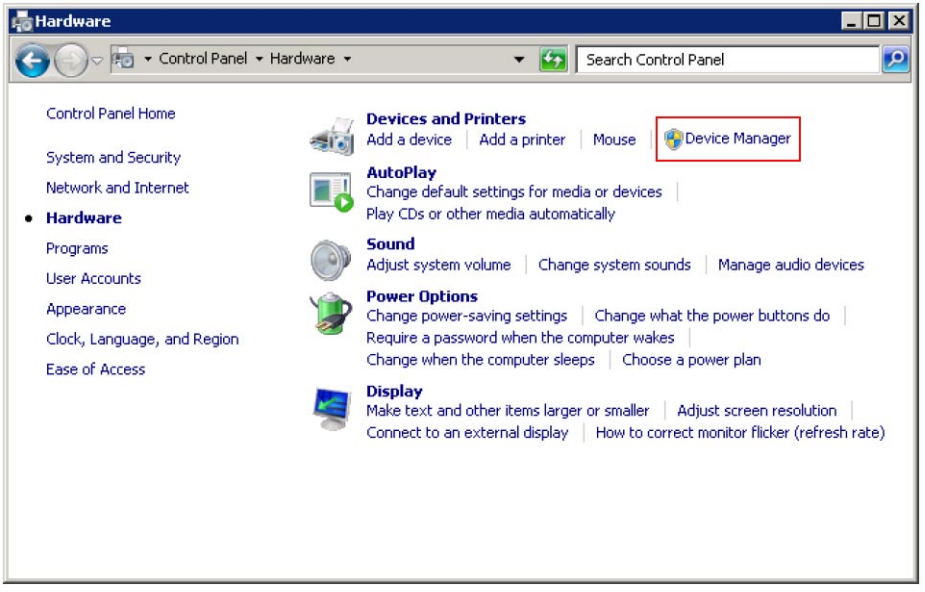


Next Steps "`biosconfig` Command Overview" on page 40

▼ 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.

biosconfig Command Overview

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 to be made.

Examples of biosconfig commands and their outputs are shown in the following table:

Command	Description
`# biosconfig -get_version`	Outputs to screen.
`# biosconfig -get_version` *file*`.xml`	Outputs to *file*`.xml`.
`# biosconfig -get_versions >` *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 several failure codes listed in "biosconfig Error Messages" on page 98.

See also:

- "How to View biosconfig Command Options" on page 40

How to View biosconfig Command Options

If you run biosconfig without arguments, you get the help output including the biosconfig command options:

- **Issue the biosconfig command without arguments:**

```
# 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
```

Note – When a command is used with a `-get` option, the information is output as standard I/O. When the `-set` option is used, the information is output to a file.

See Also

- "How to View `biosconfig` Version Information" on page 41

How to View `biosconfig` Version Information

1. **Run `biosconfig -get_version` *ver*`.xml`, to get the following output:**

```
# 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 *ver*`.xml` file, for example:**

```
<?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 40

What Changes the Boot List

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

- Changing the order in BIOS setup.
- When changing the boot order using `biosconfig` manipulates the contents of CMOS and the BIOS boot block structures stored in NVRAM, which is a dedicated part of the BIOS ROM.
- Reordering 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 also:

- "Important Notes on Devices" on page 42

Important Notes on Devices

There are a few important `biosconfig` notes on devices:

- Floppy means whatever the BIOS considers a removable device. For example, this could be a USB flash drive.
- A USB flash drive bigger than 512 MB defaults to being a disk.
- A USB/CD-ROM is classed as a CD and not a removable device.
- PXE means a bootable network device, for example, an Ethernet controller, or an InfiniBand interface that has booting support in its expansion ROM.

See also:

- "Configuring the Device Boot Order" on page 43

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.

This boot list changes as devices are installed and removed, for example, when you are doing the following:

- Changing a disk drive, which might change the string used to represent that bootable device.
- Installing and removing USB devices.
- Starting and stopping the javaConsole floppy and CD redirection.
- Adding or removing PCI cards or express modules.

This section covers the following topics:

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

`biosconfig` enables you to configure the first device to boot at the next reboot. `biosconfig` can manipulate the bootable devices individually (not by category) through specification of (subsets of) the strings that BIOS expansion ROMs use to identify their devices. `biosconfig` does this by reading the boot-related tables that the BIOS stores in NVRAM, which is a dedicated part of the BIOS ROM, and then by manipulating the contents of CMOS where the boot order is stored.

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. **Determine the current boot order of your system by using the `biosconfig -get_boot_order` command option.**

2. **Modify the boot order by using the `biosconfig -set_boot_override` command. Set XML text similar to the following:**

```
<?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>
```

See Also

- "How to Make a Persistent Change to Boot Order" on page 44
- "How to Switch Boot Devices" on page 45
- "How to Specify a Subset of Strings and a Subset of the Boot List" on page 47
- "How to Move Boot List Entries" on page 47
- "How to Change Boot Order Based on the PCI Bus, Device, or Function" on page 48

▼ How to Make a Persistent Change to Boot Order

`biosconfig` can manipulate the bootable devices individually (not by category) through specification of (subsets of) the strings that BIOS expansion ROMs use to identify their devices. `biosconfig` does this by reading the boot-related tables that the BIOS stores in NVRAM, which is a dedicated part of the BIOS ROM, and then by manipulating the contents of CMOS where the boot order is stored.

Here is an example output of the `-get_boot_order` command option 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:

- **Set XML text similar to the following:**

```
<?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>
```

See Also

- "How to Set the First Boot Device for the Next Boot" on page 43
- "How to Switch Boot Devices" on page 45
- "How to Specify a Subset of Strings and a Subset of the Boot List" on page 47
- "How to Move Boot List Entries" on page 47
- "How to Change Boot Order Based on the PCI Bus, Device, or Function" on page 48

▼ How to Switch Boot Devices

You can switching boot device 1 and 2 by using the `-set_boot_order` command option with this XML input.

Note – The boot order is sent by the boot device tag number and not in the order in which the devices appear in this file, for example, `Boot_Device_01` boots before `Boot_Device_02`).

- **View the following XML code:**

```
<BIOSCONFIG>
   <BOOT_DEVICE_PRIORITY>
      <Boot_Device_01>
         <DEVICE_NAME>SATA:3M-MRVLRD 200254-01SUN24G 0801 </DEVICE_NAME>
      </Boot_Device_01>
      <Boot_Device_02>
         <DEVICE_NAME>USB:Port1:Memorex DVD+-RAM 510L </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>
```

- **In the previous output, the specified hardware is described as follows:**

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)

See Also

- "How to Set the First Boot Device for the Next Boot" on page 43
- "How to Make a Persistent Change to Boot Order" on page 44
- "How to Specify a Subset of Strings and a Subset of the Boot List" on page 47
- "How to Move Boot List Entries" on page 47
- "How to Change Boot Order Based on the PCI Bus, Device, or Function" on page 48

How to Specify a Subset of Strings and a Subset of the Boot List

`biosconfig` permits specifying a subset of the strings and a subset of the boot list. For example, using `biosconfig -set_boot_order` with this XML input results in moving the InfiniBand device boot list entry number to number 1.

- **Set XML code similar to the following:**

```
<BIOSCONFIG>
   <BOOT_DEVICE_PRIORITY>
      <Boot_Device_01>
         <DEVICE_NAME>MLNX HCA IB</DEVICE_NAME>
      </Boot_Device_01>
   </BOOT_DEVICE_PRIORITY>
</BIOSCONFIG>
```

See Also

- "How to Set the First Boot Device for the Next Boot" on page 43
- "How to Make a Persistent Change to Boot Order" on page 44
- "How to Switch Boot Devices" on page 45
- "How to Move Boot List Entries" on page 47
- "How to Change Boot Order Based on the PCI Bus, Device, or Function" on page 48

How to Move Boot List Entries

The `biosconfig -set_boot_order` command also moves down the other boot list entries so that the boot list order becomes as follows:

- **Set similar XML code:**

```
<BOOT_DEVICE_PRIORITY>
   <Boot_Device_01>
      <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_01>
   <Boot_Device_02>
      <DEVICE_NAME>PXE:IBA GE Slot 00C8 v1324</DEVICE_NAME>
      <PCI-B-D-F>00,19,00</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>
```

See Also

- "How to Set the First Boot Device for the Next Boot" on page 43
- "How to Make a Persistent Change to Boot Order" on page 44

- "How to Switch Boot Devices" on page 45
- "How to Specify a Subset of Strings and a Subset of the Boot List" on page 47
- "How to Change Boot Order Based on the PCI Bus, Device, or Function" on page 48

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

`biosconfig` alters the boot order based on the PCI bus, device, or function if the boot order list contains that information. (USB devices do not have this information, but PCI devices do.)

For example, using `biosconfig -set_boot_order` with this XML input moves the specified Ethernet NIC to the top of the boot list:

- **Set similar XML code:**

```
<BIOSCONFIG>
   <BOOT_DEVICE_PRIORITY>
   <Boot_Device_01>
      <PCI-B-D-F>00,19,00</PCI-B-D-F>
   </Boot_Device_01>
   </BOOT_DEVICE_PRIORITY>
</BIOSCONFIG>
```

  - **As a result, the boot list now becomes:**

```
<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>
```

See Also

- "How to Set the First Boot Device for the Next Boot" on page 43
- "How to Make a Persistent Change to Boot Order" on page 44
- "How to Switch Boot Devices" on page 45
- "How to Specify a Subset of Strings and a Subset of the Boot List" on page 47
- "How to Move Boot List Entries" on page 47

BIOS CMOS Configuration

The BIOS configuration information is stored in the CMOS memory, battery back-up, 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

See also:

- "How to Configure the BIOS CMOS Using a Golden CMOS Image" on page 49
- "How to Apply the Golden Image" on page 50

▼ How to Configure the BIOS CMOS Using a Golden CMOS 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. To generate a golden (known reliable) CMOS image, use BIOS setup to configure the host as you want.

1. **Use the `biosconfig -get_CMOS_dump` to capture the 256 bytes of CMOS holding the configuration information:**

```
# biosconfig -get_CMOS_dump golden.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. **To view the XML file, type:**

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

```
<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>
```

See Also ■ "How to Apply the Golden Image" on page 50

How to Apply the Golden Image

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

- **Use the following command:**

```
# biosconfig -set_CMOS_dump golden.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 Configure the BIOS CMOS Using a Golden CMOS Image" on page 49

Configuring Individual CMOS Settings

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

- `-get_bios_settings`

  Gets CMOS settings from the platform.
- `-set_bios_settings`

  Sets CMOS settings on the platform.

To use these commands you can:

1. Use `-get_bios_settings` 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` to change settings in CMOS.

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. When using `-set_bios_settings`, you can provide a subset of the XML file output by using `-get_bios_settings` so that it includes only the settings that you wish to make.

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>
```

See also:

- "How to Retrieve Static CMOS Settings" on page 52
- "How to Configure a Dynamic Setting" on page 53
- "How to Configure NET0_Option_ROM" on page 54
- "How to View Chipset-Related Settings" on page 54
- "How to Configure System Powered Off" on page 55
- "How to Turn Off Quick Boot and Power Off Options" on page 55

How to Retrieve Static CMOS Settings

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

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

For example, for the BIOS setup question in the Quick Boot, in the Boot Settings Configuration menu, under the Boot menu, there are two choices, Enabled and Disabled, and the optimal default is Enabled.

The following are subsets of the output XML file:

- **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>
```

See Also

- "How to Configure a Dynamic Setting" on page 53
- "How to Configure NET0_Option_ROM" on page 54
- "How to View Chipset-Related Settings" on page 54
- "How to Configure System Powered Off" on page 55
- "How to Turn Off Quick Boot and Power Off Options" on page 55

How to Configure a Dynamic Setting

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

- Value in CMOS
- Behavior determined by that value
- BIOS setup strings displayed

`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 dynamic settings, you need to discover the setting that you wish to use by following these steps:

1. **Enter the BIOS setup.**
2. **Configure this setting manually and save.**
3. **Run `biosconfig-get_bios_settings` and examine the resulting XML output to find the value that the BIOS is using for the setting you wish to specify.**

   This is an example of a dynamic CMOS setting:

```
<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 this setting with `-set_bios_settings` to configure other machines.**

See Also
- "How to Configure NET0_Option_ROM" on page 54
- "How to Retrieve Static CMOS Settings" on page 52
- "How to View Chipset-Related Settings" on page 54
- "How to Configure System Powered Off" on page 55
- "How to Turn Off Quick Boot and Power Off Options" on page 55

How to Configure NET0_Option_ROM

If you do this, for this particular entry, the optimal default is Enabled which has the value 0. Disabled has the value 1.

- **Set the following XML code:**

```
<BIOSCONFIG>
   <SETUP_CONFIG>
      <Boot>
         <Option_ROM_Enable>
            <NET0_Option_ROM_>
               <SELECTED_OPTION> 1 </SELECTED_OPTION>
            </NET0_Option_ROM_>
         </Option_ROM_Enable>
      </Boot>
   </SETUP_CONFIG>
</BIOSCONFIG>
```

See Also
- "How to View Chipset-Related Settings" on page 54
- "How to Retrieve Static CMOS Settings" on page 52
- "How to Configure a Dynamic Setting" on page 53
- "How to Configure System Powered Off" on page 55
- "How to Turn Off Quick Boot and Power Off Options" on page 55

How to View Chipset-Related Settings

During BIOS development, many chipset-related settings that are not relevant to the platform under development are hidden in the BIOS setup; however, some of those appear in the `biosconfig -get_setup_config` output.

- **To view chipset-related settings, use the `biosconfig -get_setup_config` command:**

```
<BIOSCONFIG>
   <NET1_Option_ROM_>
      <HELP_STRING>This Option enables execution of the ......
</HELP_STRING>
      <DEFAULT_OPTION>Enabled</DEFAULT_OPTION>
      <SELECTED_OPTION>Enabled</SELECTED_OPTION>
```

```
      <OPTION-0>Disabled</OPTION-0>
      <OPTION-1>Enabled</OPTION-1>
   </NET1_Option_ROM_>
</BIOSCONFIG>
```

This example describes an on-board network interface card (NIC) that is not used. To avoid confusion, look in the BIOS setup to determine the name of options that you can control.

See Also

- "How to Configure System Powered Off" on page 55
- "How to Retrieve Static CMOS Settings" on page 52
- "How to Configure a Dynamic Setting" on page 53
- "How to Configure NET0_Option_ROM" on page 54
- "How to Turn Off Quick Boot and Power Off Options" on page 55

▼ How to Configure System Powered Off

To change the behavior of the system so that it remains off after AC power is restored, you can use this minimal XML file to change the setting from the default of Power On to Power Off:

- **Set the following XML code:**

```
<BIOSCONFIG>
   <SETUP_CONFIG>
      <Chipset>
         <South_Bridge_Configuration>
            <Restore_on_AC_Power_Loss>
               <SELECTED_OPTION>Power Off</SELECTED_OPTION>
            </Restore_on_AC_Power_Loss>
         </South_Bridge_Configuration>
      </Chipset>
   </SETUP_CONFIG>
</BIOSCONFIG>
```

See Also

- "How to Turn Off Quick Boot and Power Off Options" on page 55
- "How to Retrieve Static CMOS Settings" on page 52
- "How to Configure a Dynamic Setting" on page 53
- "How to Configure NET0_Option_ROM" on page 54
- "How to View Chipset-Related Settings" on page 54

▼ How to Turn Off Quick Boot and Power Off Options

You can combine changes into a single file. For example, to turn off Quick Boot and Power Off after AC power is restored you could use this:

- **Set the following XML code:**

```
<BIOSCONFIG>
   <SETUP_CONFIG>
      <Chipset>
         <South_Bridge_Configuration>
```

```
            <Restore_on_AC_Power_Loss>
               <SELECTED_OPTION>Power Off</SELECTED_OPTION>
            </Restore_on_AC_Power_Loss>
         </South_Bridge_Configuration>
      </Chipset>
      <Boot>
         <Boot_Settings_Configuration>
            <Quick_Boot>
               <SELECTED_OPTION>Disabled</SELECTED_OPTION>
            </Quick_Boot>
         </Boot_Settings_Configuration>
      </Boot>
   </SETUP_CONFIG>
</BIOSCONFIG>
```

See Also

- "How to Retrieve Static CMOS Settings" on page 52
- "How to Configure a Dynamic Setting" on page 53
- "How to Configure NET0_Option_ROM" on page 54
- "How to View Chipset-Related Settings" on page 54
- "How to Configure System Powered Off" on page 55

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:

Using the `fwupdate` Tool

The `fwupdate` tool is one of the CLI components of the Sun Server Hardware Management Pack. `fwupdate` is a cross-OS utility that enables you to query, update, and validate the firmware of storage devices such as HBA, expanders, and disks on Sun servers. `fwupdate` is supported on Linux, the Solaris OS, and Windows for x86 servers and Solaris OS for SPARC servers.

The `fwupdate` tool updates a storage component's (SAS controller, SAS expander, disk drive, SAS bridge) firmware images when an update is made to that firmware. The tool displays the inventory of storage components that can be seen by a host and can show all of the available firmware versions.

The `fwupdate` tool uses a general-purpose cross-OS storage management library to access specific hardware information and provides exploration, monitoring, and configuration of on-board (local disks) and external storage resources (JBODs) connected to the host system.

The `fwupdate` CLI commands are run on the host machine and supports the target devices. `fwupdate` supports the following storage subsystems:

- Disk drives (spinning media and Flash drives)
- HBA and embedded storage controllers, SAS1 and SAS2
- LSI SAS expander devices, SAS1 and SAS2

This section covers the following topics:

fwupdate Command-Line Interface

Command-line mode is designed to update a single component with a user-specified firmware file.

The following restrictions apply when using the `fwupdate` command:

- You must be in root permission level to run `fwupdate` commands on Unix-based platforms, or administrator permission level for Windows platforms.
- Only one device type can be upgraded per command-line execution.
- Only one file type and file may be specified by the command line.
- Components with multiple and different firmware files require a separate command-line execution to be upgraded.

Note – For Solaris, 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 101.

Note – Default (noarg) behavior of the tool lists the help options. Incomplete command-line arguments result in an error, and context-sensitive help is displayed.

The following table shows the options that apply to all CLI tools commands including `fwupdate`.

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.

The tool uses the following syntax:

`fwupdate` *subcommand device_type option*

where *subcommand* is one of the options shown in the following table.

Subcommand	Description
list	Listing mode displays system data and helps select components for upgrade.
update	Update mode allows updating a single component based on command-line directives.
reset	Reset mode allows resetting of individual components.

The preceding subcommands have the options shown in the following table.

Subcommand	Description
--?	The help option provides context sensitive help for that subcommand. After every command or subcommand, can you type **--?** or **--help** to discover the additional available subcommands or options.
--q	The quiet option outputs the mode with no prompting or stdout reporting. All output goes to a log file.
--v	The version option prints version information of the tool.

See also:

- "list Subcommand" on page 61
- "reset Subcommand" on page 62
- "update Subcommand" on page 63

list Subcommand

The list command displays the version of firmware for all components. You can compare the listed version to the latest release to determine whether your device needs a firmware upgrade.

The list options are shown in the following table.

Short Option	Long Option	Description
-n	--device_name	These two subcommands are identical in function. These options must have a mandatory parameter to designate a single device to list. The --device_name option is the common-mapped device name.
-v	--verbose	These two subcommands are identical in function. Displays much more information about each component listed. Verbose is off by default.

The supported *device_type* for the list command are:

- all

- disk
- expander
- controller
- bridge

These targets represent all of the supported component types that can be upgraded by this tool. A master list can be created, stored, and printed to inform you about the available targets. Use the `all` option to discover all of the supported targets.

The following are `fwupdate list` command examples:

- **`fwupdate list disk`**

  Executes a listing of all the disks on the system.

- **`fwupdate list expander -n c1x0 -v`**

  Shows verbose information about the expander mapped to c1x0.

See also:

- "`reset` Subcommand" on page 62
- "`update` Subcommand" on page 63

`reset` Subcommand

After firmware for a device has been updated, 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 supported *device_types* for the `fwupdate reset` command are:

- `all`
- `disk`
- `expander`
- `bridge`
- `controller`

Options for the `update` subcommand are shown in the following table.

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

The following is a `fwupdate reset` subcommand example. This example resets the disk mapped to c2d2.

```
# fwupdate reset disk -n c2d2
```

See also:

- "list Subcommand" on page 61
- "update Subcommand" on page 63

update Subcommand

The target set for the update command maps one target to one supported firmware image type that is supported by the tool.

The update subcommand supports the following targets:

- expander-firmware
- expander-boot-record
- expander-manufacturing_image
- expander-fpga
- expander-cpld
- disk-firmware
- sas-bridge-firmware
- controller-firmware
- controller-bios

Only one firmware image and one component can be specified per execution of this command.

Options for the update subcommand are shown in the following table.

Short Options	Long Option	Descriptions
-n	--device_name	A mandatory option, with a mandatory parameter, to designate the device to update. The name is the mapped name, which you can retrieve by using the list command.
-o	--filename *filename*	A mandatory option, with a mandatory parameter, designating the name of the firmware image file that is to be applied.
-r	--reset	Do not use this option. If you need to reset, use the reset subcommand. See "reset Subcommand" on page 62.
-d	--dry-run	Optional. Checks all input, executes an available dry-run check command on the firmware and component, but makes no permanent changes.

The following are update command examples:

- **fwupdate update disk-firmware -n c1d1 -f diskfirmware.file**
- **fwupdate update expander-fpga -n c1x2 -f expander.fpga -d -r**

See also:

- "`list` Subcommand" on page 61
- "`reset` Subcommand" on page 62

Device-Naming Convention

Device naming 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 26.

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 are examples of the possible execution summary messages:

- Message printed after successful a 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 machine needs to be reset, or other instructions need to be followed. Consult your product release notes for information on what needs to be done to update the version number.
- Upgrade failed:

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

Where:

- *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: "`fwupdate` Command-Line Interface" on page 60

Using the `raidconfig` Tool

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

This section covers the following topics:

raidconfig Overview

raidconfig provides exploration, monitoring, and configuration of on-board (local disks) and external storage resources (JBODs) connected to the system. It uses a set of storage libraries to access specific information about the host system. These libraries and packages are distributed and installed automatically with raidconfig. Supported components of raidconfig include those devices supported by the library. raidconfig is supported on Linux, the Solaris OS, and Windows platforms. raidconfig is capable of 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. This is accomplished by the ability to read from an XML file based on a command-line option. The XML file can be edited to fit various platform configurations. The tool allows the configuration to be easily written to a file in XML format.
- 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 example, on supported RAID types, for particular adapters based on data retrieved from the API.

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.
- raidconfig cannot create nested RAID volumes (a RAID volume as part of another RAID volume).
- On Oracle Solaris, raidconfig is not compatible with the raidctl CLI tool. (raidconfig supports SAS2, but the raidctl tool does not.)

See also "raidconfig Command Overview" on page 68.

raidconfig Command Overview

The raidconfig commands adhere to the following command syntax:

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

For Solaris, after hot-plugging any device, run the **devfsadm -C** command to reenumerate all of the system device nodes before running the raidconfig command.

When a command fails, it returns one of several failure codes listed in "raidconfig Error Codes" on page 99.

The following options apply to all CLI tools commands including `raidconfig`:

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

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

The following are subcommands for `raidconfig`:

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`	Create a RAID volume.
`delete`	Delete a RAID volume.
`modify`	Modify a RAID volume or a disk.
`add`	Add a spare disk.
`remove`	Remove a spare disk.
`export`	Generate an XML file from a RAID configuration.
`import`	Read in a RAID configuration and create RAID volumes and spares.

The subcommands are discussed below. 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 26.

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

See also "CLI Tools Command Syntax and Conventions" on page 25.

list Subcommand

The list subcommand displays controller, RAID volume, and disk data. The following subcommands are available to the list subcommand:

The following are command options to the raidconfig list command:

Subcommand	Description
all	All — Shows details on all controllers, physical disks, and RAID volumes.
controller	Controller — Shows details on all controllers. Use the -c option for details on a specific controller.
disk	Disks — Shows the physical disks.
raid	RAID — Shows all RAID details.

The following data is displayed. Items marked with an asterisk (*) show a brief listing; all other items show a verbose listing:

Controllers:

- Logical ID (0-based)
- Node ID
- Manufacturer*
- Model*
- Firmware version*
- PCI address
- PCI vendor ID
- RAID levels supported
- Current number of RAID volumes*
- Current number of disks*
- Max disks
- Max RAID volumes
- Dedicated spares (per RAID volume)
- Max global spares
- Stripe size minimum
- Stripe size minimum
- PCI device ID
- PCI subvendor ID
- PCI subdevice ID
- Battery backup status

Disks:

- Logical ID (0-based)*

- Node ID
- Chassis ID*
- Slot ID*
- Device name
- Disable
- Disk type*
- Mapped (true/false)
- Status*
- Capacity*
- Manufacturer
- Model
- Serial number
- RAID volumes that the `disk*` is part of.
- Spare state (global, dedicated, or N/A)*
- Media*

RAID volumes:

- Logical ID (0-based)*
- Node ID
- Device name*
- Name (user assigned)*
- Status*
- In-resync (true/false)
- RAID level*
- Number of disks*
- Capacity*
- Mounted
- Stripe size
- Leg size
- Read-cache
- Write-cache

See also:

- "`list` Options" on page 71

`list` Options

The following options apply to the `raidconfig list` command:

Short Option	Long Option	Description
`-c`	`--controller`	Controller ID — This option is followed by the controller ID string. Shows details about a particular controller.

Short Option	Long Option	Description
-r	--raid	RAID ID — This option is followed by the RAID ID string. Shows details about a particular RAID volume.
-d	--disks	Disk ID Numbers – This option is followed by the disk ID string(s). Comma-separated list of disks using the disk's ID. Shows details about particular disk(s). Valid only when using the raidconfig list disk data command.
-v	--verbose	Verbose — Lists all fields. By default, a brief listing shows only a subset of the fields.

See also:

- "list Subcommand" on page 70

How To Show a Brief Listing

To show a brief listing of all available controllers, RAID volumes, disks in use, and available disks:

- **Issue the following command:**

```
# ./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" on page 70
- "list Options" on page 71

▼ How To Show a Detailed Listing

To show a detailed listing of RAID volume 1 in controller 1, along with the disks that this volume is using:

- **Issue the following command:**

```
raidconfig list raid -v -r c1r1
```

See Also
- "list Subcommand" on page 70
- "list Options" on page 71

▼ How To Show a Brief Listing of a Disk

To show a brief listing of disk 2 on controller 1:

- **Issue the following command:**

```
raidconfig list disks -d c1d2
```

See Also
- "list Subcommand" on page 70
- "list Options" on page 71

create raid Subcommand

The create raid subcommand can be used to create RAID volumes. This subcommand must take at least one of the following required options:

Short Option	Long Option	Description
-d	--disks	List of disks — A list of disks with a comma separating the disk ID numbers.
-c	--controller	The controller ID number — When specific disks are not supplied, this option indicates which controller to use.
-n	--number-disks	Number of disks — The number of disks in the array. The tool chooses from the available disks. If there are not enough disks available to match the number, the command fails.
N/A	--level	Level — The RAID level supported by the controller. For example, 0, 1, 1E, 5, 10, 50, 60, etc. The levels supported for a particular controller can be seen in the raid levels supported field of the show command. Not all controllers support all RAID levels. For example, Adaptec supports 0, 1, 5, 10, and 50. Since any RAID level can be set, it is possible that the command results in an error if the RAID level is not supported. If no RAID level is supplied, level 0 is assumed.

Short Option	Long Option	Description
N/A	--stripe-size	Stripe size — In kilobytes, the stripe size of the RAID volume to be created. If this option is not supplied, a stripe size of 128 K is used.
N/A	--legs	Leg size in number of disks — For nested RAID levels (10, 50), specifies the size of the RAID components in number of physical disks.
N/A	--name	Name — The user-defined name that identifies the RAID volume. This name can be set to an empty string ("").

The maximum capacity of the RAID volume is not configurable. The storage library does not currently support configuring RAID from partial disks. You can only create RAID volumes from full disks not partial and the disks must all be of the same size.

See also:

- "How To Create a RAID Volume" on page 74

▼ How To Create a RAID Volume

The following are command examples for the `create` subcommand:

1. **To create a RAID 0 volume with capacity 2 Gb and stripe size 128K on controller 1, issue the following command:**

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

2. **To create a RAID 1 volume on controller 2 using 2 available disks, issue the following command:**

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

See Also "`create raid` Subcommand" on page 73

`delete raid` Subcommand

The `delete raid` subcommand can be used to delete RAID volumes. This subcommand must take at least one of the following required options:

Short Option	Long Option	Description
-r	--raid	The RAID volume ID number — Deletes the volume listed by ID number.
N/A	-all	All Volumes — 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.

See also:

- "How To Delete a RAID Volume" on page 75

How To Delete a RAID Volume

The following are command examples for the `delete` subcommand:

1. **To delete RAID volume 1 created on controller 1, issue the following command:**

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

2. **To delete all RAID volumes, issue the following command:**

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

See Also

- "`delete raid` Subcommand" on page 74

`add spare` Subcommand

The `add spare` subcommand can be used to add global or dedicated spare disks:

```
#raidconfig add spare
```

See also:

- "`add spare` Options" on page 75
- "How to Add a Spare" on page 76

`add spare` Options

The `add spare` subcommand takes one of the following required options:

Short Option	Long Option	Description
`-d`	`--disks`	List of disks — Adds disks where a list of disk ID numbers are separated by commas. If the `-r` option is not defined, the disks are added as global spares.
`-c`	`--controller`	Controllers — Identifies which controller the disks are with so that the disk can be identified and added. Whenever the specific disks are not specified, this option specifies the controller to use.
`-n`	`--number-disks`	Number of disks — The number of disks to be added as spares. The tool chooses from the available disks. If not enough disks are available to match the number, the command fails. If the `-r` command option is not defined, the disks are added as global spares.

Short Option	Long Option	Description
-r	--raid	RAID volume ID number — 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 may fail because of this reason.

See also:

- "add spare Options" on page 75
- "How to Add a Spare" on page 76

▼ How to Add a Spare

The following are command examples for the add spare subcommand.

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

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

2. **To create two global spares from available disks on controller 1, issue the following command:**

```
raidconfig add spare -c c1 --number-disks 2
```

3. **To create two dedicated spares on RAID volume 0 using the specified disks, issue the following command:**

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

See Also

- "add spare Subcommand" on page 75
- "add spare Options" on page 75

remove spare Subcommand and Options

The remove spare subcommand can be used to remove disks as spares or RAID volumes. This subcommand takes the following required options:

Short Option	Long Option	Description
-d	--disks	List of disks — Adds disks where ID numbers are separated by commas. If the -r option is not defined, the disks are added as global spares.
-r	--raid	RAID volume ID — 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 77

▼ How to Remove a Spare Disk or a RAID Volume

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

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

2. **To remove two disks as dedicated spares on RAID volume 0, issue the following command:**

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

See Also

- "remove spare Subcommand and Options" on page 76

modify Subcommand

The modify subcommand is used to modify the attributes of a RAID volume or disk. The modify subcommand must take at least one the following required options:

Short Option	Long Option	Description
-r	--raid	RAID volume — The RAID volume to modify.
-d	--disks	Disks — The list of disks to modify.

See also:

- "modify Options" on page 77
- "How to Modify a RAID Volume Name" on page 78

modify Options

Option	Description
--raid	RAID volume — Specifies the RAID volume to modify.
--disks	Disks to modify — Comma-separated list of disks using the disk's ID. The disks must be in a RAID Volume.
--name	Name — The user-defined name to identify the RAID volume. Can be set to an empty string ("").
--read-cache	Read cache — Disabled or enabled. Enables or disables RAID read caching.

Option	Description
`--write-cache`	Write cache: `disabled` – Disables RAID write caching. `enabled` – Enables RAID write caching. `enabled_protect` — Enables caching only if the battery is available.
`--disable`	Disable — True or false. Sets a disk to be disabled (offline) or reenabled (online).

See also:

- “`modify` Subcommand” on page 77
- “How to Modify a RAID Volume Name” on page 78

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, issue the following command:**

```
raidconfig modify raid -r c0r0 --name engineering
```

See Also

- “`modify` Subcommand” on page 77
- “`modify` Options” on page 77

`export` Subcommand

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

The `export` subcommand takes a required 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.

See also:

- “`raidconfig export` Options” on page 79
- “How to Export an Inventory to a File” on page 79

raidconfig export Options

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

This subcommand must take at least one of the following required subcommands:

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

This subcommand must take one or more of the following required options:

Option	Description
--use-number-disks	Number of disks in a volume — For configuration data, instead of listing each disk in a RAID volume, the XML file can contain the number of disks in each RAID volume.

See also:

- "export Subcommand" on page 78
- "How to Export an Inventory to a File" on page 79

How to Export an Inventory to a File

To export the inventory or a configuration and write it to a file:

1. **To export the inventory and write it to a file, issue the following command:**

   ```
   raidconfig export inventory raid_inv.xml
   ```

2. **To export a configuration and write it to a file, issue the following command:**

   ```
   raidconfig export config raid_config.xml
   ```

See Also

- "export Subcommand" on page 78
- "raidconfig export Options" on page 79

import Subcommand

The import subcommand is used to read an XML-formatted configuration file and to configure RAID volumes based on the file. If creating a specific RAID volume fails, the error is logged and the next RAID volume in the file is created. The import subcommand takes the config type and a required file name as the operand (modifier).

See also:

- "How to Configure RAID Volumes from a File" on page 80

How to Configure RAID Volumes from a File

The following is an example command to read the RAID configuration from a file and configure the RAID volumes according to that file:

- **To configure the RAID volumes according to a configuration file, issue the following command:**

```
raidconfig import config raid_config.xml
```

See Also "import Subcommand" on page 80

Using the `ilomconfig` Tool

`ilomconfig` allows you to configure ILOM from the host OS without having to first connect to the management network. It also serves as an *XML builder* as it helps you construct an XML file for a subsequent restore operation.

This section covers the following topics:

`ilomconfig` Overview

`ilomconfig` can restore configuration (with some limitations) from an XML file and allows parameters to be specified using the command line. `ilomconfig` can modify an XML file that has been generated from the ILOM backup feature; however, `ilomconfig` cannot generate a backup ILOM configuration file. Use the `create` or `modify` subcommand to create or modify XML files.

You must be in root permission level to run `ilomconfig` commands on Unix-based platforms, or Administrator permission level for Windows platforms.

In summary, `ilomconfig` can perform the following functions:

- Restores from an ILOM backup XML file.
- Modifies the XML file using convenience sub-commands.
- Sets the network, including DHCP and sideband.
- Lists and configures identification information, including hostname, contact, location, and description.
- Lists and configures DNS.
- Lists and configures clock including time zone.
- Lists and configures user management.
- Lists and configures SNMP community.

You can use an XML file to perform either one of the following functions:

- Create a new XML file for functions that have an `ilomconfig` command.

- Modify an existing XML file that was generated by ILOM; however, some parts of the XML file cannot be restored.

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

The `list` commands, with the XML-file argument, can be used to view the contents of the XML file.

Note – If the XML-file option is specified, the commands operate on that XML file. If the XML file option is omitted, the changes are made directly to the ILOM.

The following ILOM settings cannot be restored from an XML file or specified using commands:

- SSH private keys
- User SSH keys
- SSL cert
- COD license
- LDAP and AD certs
- Platform binary data (currently limited to SPARC LDOMS config)

In addition, the following settings cannot be restored from an XML file if they are encrypted. However, unencrypted values in the XML file can be generated for some of these settings by the `ilomconfig` tool, or the XML file can be edited by hand and then restored:

- User passwords
- SNMP users
- LDAP/LDAPSSL/RADIUS passwords
- Servicetag passphrase

See also:

- "`ilomconfig` Command Usage" on page 82
- "`ilomconfig` Error Codes" on page 100

`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 100.

The following options are available to all CLI tools commands including `ilomconfig`:

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.

This section covers the following topics:

How to Import an XML Configuration

To import an XML configuration file to configure ILOM, use the `ilomconfig import config` command. Use the `-y` option to bypass the yes/no confirmation prompt.

You can also use this command to restore the system configuration by importing a known reliable XML file.

- **Issue the following command:**

  `ilomconfig import config [--xmlfile` *file_name*`] [-y]`

  where *file_name* is the name of the XML configuration file you want to use to configure ILOM.

See Also

- "How to Restore ILOM to Defaults by Using an XML Configuration" on page 84

How to Restore ILOM to Defaults by Using an XML Configuration

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

- **Issue the following command:**

  ```
  ilomconfig reset config [-y]
  ```

See Also

- "How to Import an XML Configuration" on page 83

How to List a System Summary

To list system summary information including the product name, part number, serial number, ILOM host name, and ILOM version information, use the `ilomconfig list system-summary` command. This functionality is equivalent to the Summary tab in the web interface.

- **Issue the following command:**

  ```
  ilomconfig list system-summary
  ```

See Also

- "How to Import an XML Configuration" on page 83

How to Create a User

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

- **Issue the following command:**

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

  where `--role` is the role of the ILOM user.

See Also

- "How to Delete a User" on page 84

How to Delete a User

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

- **Issue the following command:**

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

See Also ■ "How to Create a User" on page 84

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 accordingly rather than modifying ILOM itself.

- **Issue the following command:**

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

See Also ■ "How to List Users" on page 85

How to List Users

To list one or all users, use the `ilomconfig list user` command. When you specify an XML file name, this command lists users defined in the XML file rather than querying ILOM itself.

- **Issue the following command:**

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

See Also ■ "How to Modify a User Password or Role" on page 85

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 ILOM itself.

- **Issue the following command:**

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

See Also ■ "How to Create an SNMP Community" on page 85

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 ILOM itself.

- **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 85

How to List Network Settings

To list 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 ILOM itself.

- **Issue the following command:**

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

See Also

- "How to Modify Network Settings" on page 86

How to Modify Network Settings

To modify settings, use the ilomconfig modify network 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 accordingly rather than modifying ILOM itself.

- **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.	**ro** or **rw**
--ipaddress	ILOM IP address.	**255.255.255.0**
--netmask	ILOM netmask.	**255.255.255.0**
--gateway	ILOM gateway.	**255.255.255.0**
--state	ILOM management port path.	**/SP/SP/NET0**
--mgmtport	ILOM management port state.	**enabled** or **disabled**
--xmlfile	Modify specified XML file rather than ILOM.	**file.xml**

See Also ■ "How to List Network Settings" on page 86

How to List SP Identification Information

To list identification information for the SP, use the `ilomconfig list identification` command. This command lists SP 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 ILOM itself.

- **Issue the following command:**

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

See Also ■ "How to Modify Identification Information" on page 87

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 ILOM itself.

- **Issue the following command:**

  `ilomconfig modify identification [--hostname` *hostname*`] [--system-contact` *system_contact*`] [--system-location` *system_location*`] [--system-identifier` *system_identifier*`]`

Option	Description	Example
`--hostname`	ILOM host name.	**`ro`** or **`rw`**
`--system-contact`	ILOM system contact field.	**`user`**
`--system-location`	ILOM system location field.	**`west`**
`--system-identifier`	ILOM system identifier field.	**`x4800`**
`--xmlfile`	Modify specified XML file rather than ILOM.	**`file.xml`**

See Also ■ "How to List DNS Information" on page 87

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 ILOM itself.

- **Issue the following command:**

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

See Also

- "How to Modify DNS Information" on page 88

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 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 ILOM separated by commas.	**ro** or **rw**
--auto-dns	ILOM Auto-DNS state.	**enabled** or **disabled**
--searchpath	List of search suffixes in preferred order and separated by commas.	
--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 ILOM.	**file.xml**

See Also

- "How to List DNS Information" on page 87

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 ILOM itself.

- **Issue the following command:**

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

See Also

- "How to Modify Clock Information" on page 89

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 accordingly rather than modifying ILOM itself.

- **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`	ILOM date in *MMDDhhmmYYYY* format or*MMDDhhmmYYYY.ss* format.	**`032514272010`**
`--timezone`	ILOM clock time zone, such as GMT.	**`enabled`** or **`disabled`**
`--usentp`	ILOM NTP client state.	**`enabled`** or **`disabled`**
`--ntp-server1`	ILOM NTP server 1 IP address.	*aaa.bbb.ccc.ddd*
`--ntp-server2`	ILOM NTP server 2 IP address.	*aaa.bbb.ccc.ddd*
`--xmlfile`	Modify specified XML file rather than ILOM.	**`file.xml`**

See Also

- "How to List Clock Information" on page 88

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:

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

This section covers the following topics:

`ipmitool` Overview

IPMI configuration CLI tool (`ipmitool`), as part of the Sun Server Hardware Management Pack, 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 *Sun Server Management Agents 2.0 User's Guide*.

Sun IPMI System Management Driver 2.1

Note – See the OS support matrix in "Installing Components" on page 11 to determine if you need this driver.

Microsoft Windows Server 2003 SP1 – 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 works only with Microsoft Windows Server 2003 SP1.

For later OS versions (including Microsoft Windows Server 2003 R2 and Microsoft Windows Server 2003 SP2) the Microsoft-provided IPMI driver included in their Hardware Management Module provides the same functionality.

See also:

- "`ipmitool` Overview" on page 91
- "How to Install Sun IPMI System Management Driver 2.1 Manually" on page 92
- "How to Perform an Unattended Installation of the Sun IPMI System Management Driver 2.1" on page 93
- "How to Verify `ipmitool` Installation" on page 94

▼ How to Install Sun IPMI System Management Driver 2.1 Manually

To install the Sun IPMI System Management Driver 2.1 for Microsoft Windows Server 2003 SP1:

1 Uninstall any previous versions of this driver:

- **Right-click My Computer and select Properties.**
- **Select the Hardware tab and click Device Manager.**
- **Expand the System Devices section.**
- **Locate SUN IPMI System Management Driver and right-click this item.**

- **Select Uninstall and confirm the removal.**

2. **Open the Control Panel and select New Hardware.**
3. **Click Next.**
4. **Select Yes, I have already connected the hardware, and click Next.**
5. **Select Add a new hardware device from the displayed list, and click Next.**
6. **Select Install the hardware that I manually select from a list, and click Next.**
7. **Select System Devices, and click Next.**
8. **Click the Have Disk tab to specify the file system location where the `ism.inf` file is stored.**
9. **Click OK to uninstall the driver.**

See Also

- "`ipmitool` Overview" on page 91
- "Sun IPMI System Management Driver 2.1" on page 92
- "How to Perform an Unattended Installation of the Sun IPMI System Management Driver 2.1" on page 93
- "How to Verify `ipmitool` Installation" on page 94

How to Perform an Unattended Installation of the Sun IPMI System Management Driver 2.1

This section describes how to perform unattended (no user interaction required) installation of the Sun IPMI System Management Driver 2.1 for Microsoft Windows Server 2003 SP1.

To perform an unattended installation , you must download the `devcon` CLI utility:

1. **Download the `devcon` command-line utility from the Microsoft web site at:**
   `http://support.microsoft.com/kb/311272`
2. **Uninstall any previous versions of this driver:**

```
F:\ism> devcon remove *ISM
```

3. **From the directory containing `ism.inf` (assumed to be `F:\ism` in the following example), run the following command:**

```
F:\\ism> devcon install ism.inf *ism
```

Next Steps

- "How to Verify `ipmitool` Installation" on page 94

See Also
- "`ipmitool` Overview" on page 91
- "Sun IPMI System Management Driver 2.1" on page 92
- "How to Install Sun IPMI System Management Driver 2.1 Manually" on page 92

How to Verify `ipmitool` Installation

1. **Obtain the device string:**

   ```
   F:\ism> devcon find *ism
   ```

   The output contains the device string for this driver.

2. **Run the following command to obtain the status:**

   ```
   F:\ism> devcon status @device-string
   ```

3. **Assuming the `devcon find` command returns the device string `ROOT\SYSTEM\0003`, the following command:**

   ```
   F:\ism> devcon status @ROOT\SYSTEM\0003
   ```

   The expected output should be similar to this:

   ```
   Name: Sun IPMI System Management Driver v2.1
   Driver is running
   1 matching device(s) found.
   ```

Next Steps
- "How to Configure for Any Floppy or Removable Media to Boot First" on page 96

How to Configure for PXE to Boot First

On some platforms, the alternative to using `biosconfig` to control the boot order is IPMI commands, which can also make persistent changes to the boot order through the service processor. 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 the following command:**

  ```
  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 gPXE first followed by GE if IB fails over. The BIOS setup reflects the change in the boot order.

See Also
- "How to Configure for the Hard Drive to Boot First" on page 95
- "How to Configure for Any CD/DVD to Boot First" on page 95
- "How to Configure for Any Floppy or Removable Media to Boot First" on page 96

▼ How to Configure for the Hard Drive to Boot First

On some platforms, the alternative to using `biosconfig` to control the boot order is IPMI commands, which can also make persistent changes to the boot order through the service processor. This interface can only specify which is the highest priority category of boot devices. 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 just like the `ipmitool` chassis `bootdev` commands. During BIOS POST, the BIOS asks the SP for boot flags. The raw commands have just one 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 the host Linux system.

- **Use the following command:**

```
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.

See Also
- "How to Configure for PXE to Boot First" on page 94
- "How to Configure for Any CD/DVD to Boot First" on page 95
- "How to Configure for Any Floppy or Removable Media to Boot First" on page 96

▼ How to Configure for Any CD/DVD to Boot First

On some platforms, the alternative to using `biosconfig` to control the boot order is IPMI commands, which can also make persistent changes to the boot order through the service processor. This interface can only specify which is the highest priority category of boot devices. 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 just like the `ipmitool` chassis `bootdev` commands. During BIOS POST, the BIOS asks the SP for boot flags. The raw commands have just one 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 the host Linux system.

- **Use the following command:**

```
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.

See Also
- "How to Configure for PXE to Boot First" on page 94
- "How to Configure for the Hard Drive to Boot First" on page 95
- "How to Configure for Any Floppy or Removable Media to Boot First" on page 96

How to Configure for Any Floppy or Removable Media to Boot First

On some platforms, the alternative to using `biosconfig` to control the boot order is IPMI commands, which can also make persistent changes to the boot order through the service processor. This interface can only specify which is the highest priority category of boot devices. 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 just like the `ipmitool` chassis `bootdev` commands. During BIOS POST, the BIOS asks the SP for boot flags. The raw commands have just one 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 the host Linux system.

- **Use the following command:**

```
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.

See Also
- "How to Configure for PXE to Boot First" on page 94
- "How to Configure for the Hard Drive to Boot First" on page 95
- "How to Configure for Any CD/DVD to Boot First" on page 95

CLI Tools Error Codes

This section covers the following topics:

Common Error Codes

The following is the list of 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.
11	Option not supported.

TABLE 1 Common Error Codes *(Continued)*

Code Number	Error Description
12	Storage init failed.
13	Name too long.
14	Invalid string after subcommand.
15	XML filename required.
16	Invalid argument.
17	Failure writing XML file.
18	Device is busy, command cannot be completed.

See also:

- "biosconfig Error Messages" on page 98.
- "raidconfig Error Codes" on page 99.
- "ilomconfig Error Codes" on page 100.
- "fwupdate Error Codes" on page 101.

biosconfig Error Messages

This sections lists possible biosconfig errors and action to take when they occur.

Error Number(s)	Description
Errors 36-49	Verify that either Microsoft IPMI driver (2003 R2) or 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:

- "Using biosconfig" on page 31
- "biosconfig Command Overview" on page 40

raidconfig Error Codes

Errors might be returned if you are attempting to configure the RAID entry to 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 list of error codes and strings specific to this tool are show in the following table.

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.
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 leg size for controller.
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.

TABLE 2 raidconfig Error Codes *(Continued)*

Code Number	Error Description
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.

See also:

- "Using the raidconfig Tool" on page 67

ilomconfig Error Codes

The following are ilomconfig error codes:

TABLE 3 ilomconfig Error Codes

Code Number	Error Description
50	Cannot connect to BMC.
51	Missing -username option.
52	Missing -password option.
54	Missing -communityname option.
55	Specified community already exists.
57	Community name does not exist.

TABLE 3 ilomconfig Error Codes *(Continued)*

Code Number	Error Description
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.
72	One of the following, depending on the command used: ■ User already exists. ■ User does not exist. ■ ILOM error occurred.

See also:

- "Common Error Codes" on page 97
- "Using the ilomconfig Tool" on page 81

fwupdate Error Codes

The following is the list of fwupdate 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 4 fwupdate Error Codes

Code Number	Error Description
200	Invalid device type.

TABLE 4 fwupdate Error Codes *(Continued)*

Code Number	Error Description
201	Invalid image type.
202	Invalid device ID.
203	Reset failed.
204	Firmware check failed.
205	Firmware download failed.
206	Component mismatch.
207	No file name.
208	Invalid image file.
209	Cannot reset.
210	Reset mismatch.
211	No device specified.
212	Update canceled.

See also:

- "Common Error Codes" on page 97
- "Using the fwupdate Tool" on page 59

Index

R

S

T

U

V

W

X