Sun Server CLI Tools and IPMItool User's Guide

Part No: 821–0997–10
January 2010

100224@23474

Contents

Preface

The Sun Server Management documentation provides detailed information to 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.pack`

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 will 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.

Installing Components

This section describes how to install and uninstall Hardware Management Pack components on a Sun x64 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. This enables you to choose the installation options from a command-line interface.
- Use the Sun Server Component Manager in unattended mode. This enables you to perform an unattended installation.

Regardless of the installation method you choose, the installation must be carried out as a user with administrative privileges, such as root.

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:

- 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 x64 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.

Note – If you have installed an earlier version of Hardware Management Pack on a Linux or Solaris operating system, you must manually uninstall the earlier version before installing the latest version.

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 x64 server that you want to manage.

Note – On the Solaris operating system, due to the restrictions of `pkgadd`(1M) the path which you uncompress the Hardware Management Pack to must not contain any white 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 is a command-line application that 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. You should follow these procedures:

- "(Linux and Solaris Operating Systems) How to Install Component Manager" on page 9
- Install components by either "(Linux and Solaris Operating Systems) Using Component Manager in Interactive Mode" on page 10 or "(Linux and Solaris Operating Systems) Using Component Manager in Unattended Mode" on page 13

On Windows operating systems Component Manager is an executable which is copied to the install directory when at least one component is installed. You should follow this procedure "(Windows Operating Systems) Using Component Manager" on page 15.

(Linux and Solaris Operating Systems) 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 either install components interactively using a command-line interface or automatically using command-line switches, which enables unattended installs.

▼ (Linux and Solaris Operating Systems) How to Install Component Manager

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

1. **Open a terminal.**
2. **Navigate to the directory where you uncompressed the Hardware Management Pack package, 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 either use the interactive mode or unattended mode. For more information see:

- "(Linux and Solaris Operating Systems) Using Component Manager in Interactive Mode" on page 10
- "(Linux and Solaris Operating Systems) Using Component Manager in Unattended Mode" on page 13

(Linux and Solaris Operating Systems) Using Component Manager in Interactive Mode

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

Note – On Windows operating systems interactive mode is not available. See "(Windows Operating Systems) Using Component Manager" on page 15

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

Option	Functionality
[L]ist	Displays the list of currently available components

Option	Functionality
[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

Options are chosen in the Component Manager by typing the letter shown between the [] characters.

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

Before You Begin You must install the Component Manager before proceeding. The following procedure must be carried out as a user with root privileges. If you have installed a previous version of Hardware Management Pack, you must manually uninstall the earlier version before installing the latest version. Component Manager detects older versions of Hardware Management Pack during the install procedure, but does not upgrade the system due to changes in the packaging of this release.

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 Operating Systems) 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 Operating Systems) 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 on both the already installed and available components.
`-D`	Print detailed information on 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.
`-U` *COMPONENT1:COMPONENT2*	Uninstall components. Component names are separated by a colon (:). If "ALL" is given as the component list, all available components are installed.

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 using the `-C` or `-D` options and are listed in square brackets.

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

When using the Component Manager in unattended mode, components can be installed separately or all components can be installed. 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.

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 Operating Systems) 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 on 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 Operating Systems) Using Component Manager

When using Component Manager on Windows operating systems only the unattended mode is available. When passing options and actions to Component Manager in unattended mode, observe the following conventions:

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

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

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 on both the already installed and available components.
`/D`	Print detailed information on 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 installed.

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 using the `/C` or `/D` options.

▼ (Windows Operating Systems) How to Install Using Component Manager

When using the Component Manager in unattended mode, components can be installed separately or all components can be installed. 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.

1. **Open a terminal.**

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 terminal.

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 Operating Systems) How to Uninstall Using Component Manager

1 **Open a terminal.**

2 **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 on the currently installed components.

---

3 **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 install, separated by space characters.

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

  ```
  sunssmcompmgr /U ALL
  ```

  The selected components are uninstalled.

BIOS Configuration Tool

The BIOS configuration 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 the user 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 the user 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. If the machines differ too much, then a customized XML file must be used.

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 your desired 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 7

This section covers the following topics:

Refer to the user instruction sections for details on using BIOSconfig: "Using BIOSconfig" on page 21

Refer to the installation section on Component Manager for more details on running BIOSconfig: "Getting Started" on page 7.

BIOSconfig Dependencies

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

For Solaris, see: "How to Obtain the BMC Driver" on page 48

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. 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`).

See also:

"BIOSconfig Terminology" on page 20

"Using BIOSconfig" on page 21

"BIOSconfig Command Overview" on page 22

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, see the following link:

  `http://www.intel.com/design/servers/ipmi`
- IPMItool is an open source tool `http://ipmitool.sourceforge.net/` 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 (`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 Dependencies" on page 20

"Using BIOSconfig" on page 21

"BIOSconfig Command Overview" on page 22

Using BIOSconfig

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

To use BIOSconfig, the user must have a working knowledge of editing XML files. The process of editing the BIOS includes using BIOSconfig to do the following tasks:

1. Runs `BIOSconfig` to create an XML file.
2. Reviews the XML file and modifies it, if necessary, to configure the system to their specific needs.
3. Writes 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 is optional (for example, indentation is optional).

See also:

"BIOSconfig Dependencies" on page 20

"BIOSconfig Terminology" on page 20

"BIOSconfig Command Overview" on page 22

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:

TABLE 1 Examples of BIOSconfig Commands

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

When it fails, it returns one of several failure codes listed at the end of this document.

See also: "How to View BIOSconfig Command Options" on page 22

How to View BIOSconfig Command Options

If you run `biosconfig` without arguments, you get the following help output:

- **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 23

▼ 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 22

What Changes the Boot List

The boot list can be changed by any of the following:

- Changing the order in BIOS setup.
- Changing the order using a tool (such as BIOSconfig) that 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 24

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 "What Changes the Boot List" on page 24

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 will be attempted.

This boot list changes as devices are installed and removed, for example:

- 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 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 (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:

● **Input 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>
```

Next Steps "How to Switch Boot Devices" on page 26

▼ How to Switch Boot Devices

Switching boot device 1 and 2 can be accomplished 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 above output, the specified hardware are 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)

Next Steps "How to Specify a Subset of Strings and a Subset of the Boot List" on page 28

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 one.

- **Input 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>
```

Next Steps "How to Move Boot List Entries" on page 28

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:

- **Input 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>
```

Next Steps "How to Change Boot Order Based on the PCI Bus, Device, or Function" on page 29

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

BIOSconfig also alters the boot order based on the PCI bus/device/function if the boot order list contains that information. (For example, 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:

- **Input 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>
```

Next Steps "How to Configure for PXE to Boot First" on page 29

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 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 just have 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 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 30

"How to Configure for Any CD/DVD to Boot First" on page 31

"How to Configure for Any Floppy or Removable Media to Boot First" on page 31

▼ 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 just have 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 will boot first. The BIOS setup reflects the change in the boot order.

See Also "How to Configure for PXE to Boot First" on page 29

"How to Configure for Any CD/DVD to Boot First" on page 31

"How to Configure for Any Floppy or Removable Media to Boot First" on page 31

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 just have 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 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 29

"How to Configure for the Hard Drive to Boot First" on page 30

"How to Configure for Any Floppy or Removable Media to Boot First" on page 31

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 just have 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 Java Console redirected floppy. The BIOS setup reflects the change in the boot order.

See Also "How to Configure for PXE to Boot First" on page 29

"How to Configure for the Hard Drive to Boot First" on page 30

"How to Configure for Any CD/DVD to Boot First" on page 31

BIOS CMOS Configuration

The BIOS configuration information is stored in the CMOS, which is battery-backed-up memory in the hosts' chipset. Through the BIOS-POST-time BIOS setup interface you can manipulate many of these settings. BIOSconfig provides an alternative interface to modify these with a program on the host OS. BIOSconfig supports two methods for configuring BIOS CMOS settings: copying a golden (known reliable) image and controlling each setting individually.

See also:

"How to Configure the BIOS CMOS Using a Golden CMOS Image" on page 32

"How to Apply the Golden Image" on page 34

▼ 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 desired.

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 34

▼ 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 below 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 32

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 desired 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 35.

"How to Configure a Dynamic Setting" on page 37

"How to Configure NET0_Option_ROM" on page 38

"How to View Chipset-Related Settings" on page 38

"How to Configure System Powered Off" on page 39

"How to Turn Off Quick Boot and Power Off Options" on page 40

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>
```

Next Steps "How to Configure a Dynamic Setting" on page 37

See Also "How to Configure NET0_Option_ROM" on page 38

"How to View Chipset-Related Settings" on page 38

"How to Configure System Powered Off" on page 39

"How to Turn Off Quick Boot and Power Off Options" on page 40

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 above 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.**

Next Steps "How to Configure NET0_Option_ROM" on page 38

See Also "How to Retrieve Static CMOS Settings" on page 35

"How to View Chipset-Related Settings" on page 38

"How to Configure System Powered Off" on page 39

"How to Turn Off Quick Boot and Power Off Options" on page 40

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.

- **Input 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>
```

Next Steps "How to View Chipset-Related Settings" on page 38

See Also "How to Retrieve Static CMOS Settings" on page 35

"How to Configure a Dynamic Setting" on page 37

"How to Configure System Powered Off" on page 39

"How to Turn Off Quick Boot and Power Off Options" on page 40

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.

Next Steps "How to Configure System Powered Off" on page 39

See Also "How to Retrieve Static CMOS Settings" on page 35

"How to Configure a Dynamic Setting" on page 37

"How to Configure NET0_Option_ROM" on page 38

"How to Turn Off Quick Boot and Power Off Options" on page 40

▼ 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:

- **Input 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>
```

Next Steps "How to Turn Off Quick Boot and Power Off Options" on page 40

See Also "How to Retrieve Static CMOS Settings" on page 35

"How to Configure a Dynamic Setting" on page 37

"How to Configure NET0_Option_ROM" on page 38

"How to View Chipset-Related Settings" on page 38

How to Turn Off Quick Boot and Power Off Options

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

- **Input 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 35

"How to Configure a Dynamic Setting" on page 37

"How to Configure NET0_Option_ROM" on page 38

"How to View Chipset-Related Settings" on page 38

"How to Configure System Powered Off" on page 39

Failure Return Codes

This table describes the error return codes for BIOSconfig.

TABLE 2 Error Return Codes

Error Type	Return Code	Displayed Image
SUCCESS	0	Success
FAILURE	1	Error (1): Unknown failure

TABLE 2 Error Return Codes *(Continued)*

Error Type	Return Code	Displayed Image
INVALID_ARGS	2	Error (2): Invalid arguments
MULTIPLE_OPS	3	Error (3): Multiple operations not allowed
CANT_OPEN_INPUT_FILE	4	Error (4): Cannot open input file for reading
CANT_OPEN_OUTPUT_FILE	5	Error (5): Cannot open output file for writing
WRITE_FAILED	6	Error (6): Cannot write to output file
ILOM_UNRESPONSIVE	7	Error (7): Cannot get the ILOM to respond
ILOM_TIMEOUT	8	Error (8): ILOM responsive but transaction timed out
ILOM_FAILURE	9	Error (9): ILOM indicated the operation as failed
INPUT_FILE_PARSE_FAILURE	10	Error (10): Input file parse error
NW_VERIFY_FAILURE	11	Error (11): Network configuration command issued, but verification failed
PW_VERIFY_FAILURE	12	Error (12): Password command issued, but verification failed
BOOT_VERIFY_FAILURE	13	Error (13): Boot option command issued, but verification failed
INVALID_PASSWORD	14	Error (14): Invalid password
OUT_OF_MEMORY	15	Error (15): Out of memory
READ_FAILED	16	Error (16): Input file read failed
ILOM_IFC_NOT_FOUND	17	Error (17): ILOM interface not found
BBS_NOT_FOUND	18	Error (18): Runtime BIOS Boot Settings (BBS) interface not found
BBS_FAILURE	19	Error (19): BIOS Boot Settings failed
BOOT_ DEVICE_NOT_FOUND	20	Error (20): Specified boot device not found
DMI_TYPE_NOT_FOUND	21	Error (21): SMBIOS (DMI) structure not found
DMI_UPDATE_FAILED	22	Error (22): SMBIOS (DMI) update failed
PROCESSING_BIOS_IMAGE	23	Error(23): BIOS Image process failed
PROCESSING_SETUP_DATA	24	Error(24): Setup Data process failed
FLASH_INTERFACE_ERROR	30	Error(30): Flash interface error
BIOS_READ_FAILURE	31	Error(31): BIOS read failure

TABLE 2 Error Return Codes *(Continued)*

Error Type	Return Code	Displayed Image
BIOS_DECODE_ERROR	32	Error(32): BIOS data decode error
BIOS_DATA_MISMATCH	33	Error(33): BIOS and XML data mismatch
BIOS_REBUILD_ERROR	34	Error(34): BIOS rebuild error
BIOS_UPDATE_FAILURE	35	Error(35): BIOS Flash update Failure
DRV_INIT_COM	36	Error(36): WMI error cannot initialize COM Library (Windows only)
DRV_INIT_SECURE	37	Error(37): WMI error cannot initialize security (Windows only)
DRV_PTR_SVC	38	Error(38): WMI error cannot create pointer to WbemServices (Windows only)
DRV_WMI_NAMESPACE	39	Error(39): WMI error cannot connect to WMI namespace (Windows only)
DRV_SET_PROXY	40	Error(40): WMI error cannot set proxy blanker (Windows only)
DRV_ENUM_CLASS	41	Error(41): WMI error cannot enumerate class object (Windows only)
DRV_FIND_CLASS	42	Error(42): WMI error cannot find class object (Windows only)
DRV_PATH_OBJ	43	Error(43): WMI error cannot get path to the object (Windows only)
DRV_HANDLE_NULL	44	Error(44): WMI error driver handle is passed as null (Windows only)
DRV_GET_OBJ	45	Error(45): WMI error cannot get the class object (Windows only)
DRV_PARAM_DEF	46	Error(46): WMI error cannot get the params definitions (Windows only)
DRV_PARAM_SPAWN	47	Error(47): WMI error cannot spawn param instance (Windows only)
DRV_PARAM_COPY	48	Error(48): WMI error cannot copy the parameters (Windows only)
DRV_WMI_METHOD	49	Error(49): WMI error executing WMI method (Windows only)

TABLE 2 Error Return Codes *(Continued)*

Error Type	Return Code	Displayed Image
DRV_COMPLN_CODE	50	Error(50): WMI error cannot get the completion code (Windows only)
DRV_COMPLN_ERROR	51	Error(51): WMI error completion code is non-zero (Windows only)
DRV_GET_RESP_SIZE	52	Error(52): WMI error response data size mismatch (Windows only)
DRV_GET_RESP_DATA	53	Error(53): WMI error cannot get the ResponseData (Windows only)
DRV_RESP_LEN_OVERFLOW	54	Error(54): WMI error response length too long (Windows only)
ISM_INIT_COM	56	Error(56): ISM error SetupDiGetClassDevs failed (Windows only)
ISM_ENUM_CLASS	57	Error(57): ISM error SetupDiEnumDeviceInterfaces failed (Windows only)
ISM_INSUFF_BUFFER	58	Error(58): ISM error insufficient buffer length (Windows only)
ISM_INTF_MALLOC	59	Error(59): ISM error interface malloc failed (Windows only)
ISM_GET_INTF	60	Error(60): ISM error SetupDiGetInterfaceDeviceDetail failed (Windows only)
ISM_MAKE_FILE	61	Error(61): ISM error createFile failed (Windows only)
ISM_HANDLE_NULL	62	Error(62): ISM error driver handle is passed as null (Windows only)
ISM_IPMI_COMMAND	63	Error(63): ISM error executing SMI method (Windows only)
RC_INSUFFICIENT_PRIVILEGES	64	Error (64): Insufficient privileges to run BIOS config
INTERNAL_ERROR	FF	Error (FF): Internal program error

See also:

"How to Retrieve Static CMOS Settings" on page 35.

"How to Configure a Dynamic Setting" on page 37

"How to Configure NET0_Option_ROM" on page 38

"How to View Chipset-Related Settings" on page 38

"How to Configure System Powered Off" on page 39

"How to Turn Off Quick Boot and Power Off Options" on page 40

Commands 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`.

Note – This issue is scheduled to be fixed in a future release.

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

See also:

"How to Retrieve Static CMOS Settings" on page 35.

"How to Configure a Dynamic Setting" on page 37

"How to Configure NET0_Option_ROM" on page 38

"How to View Chipset-Related Settings" on page 38

"How to Configure System Powered Off" on page 39

"How to Turn Off Quick Boot and Power Off Options" on page 40

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 View BIOSconfig Commands in Solaris OS" on page 47
- "How to Obtain the BMC Driver" on page 48

How to View BIOSconfig Commands in Solaris OS

- **To view a list of commands, type:**

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

BIOSconfig Specification Version 2.4

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

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

Available options (Required):
-get_version Get version of this tool
-get_boot_order Get the BOOT Devices list
-set_boot_order Set the BOOT Devices list
```

```
-get_bios_settings Get setup configuration from BIOS
-set_bios_settings Set setup configuration to BIOS ROM
-get_CMOS_dump Get 256 bytes CMOS setup data from BIOS
-set_CMOS_dump Set 256 bytes of CMOS setup data to BIOS
```

See Also

- "How to View BIOSconfig Commands in Solaris OS" on page 47
- "How to Obtain the BMC Driver" on page 48

How to Obtain the BMC Driver

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 "How to View BIOSconfig Commands in Solaris OS" on page 47

BIOSconfig for Windows

The BIOS configuration tool for Windows (`BIOSconfig`) is a utility 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.

See also "BIOSconfig for Windows Known Issues" on page 49.

BIOSconfig for Windows Known Issues

The following are known issues with BIOSconfig version 2.2.1 for Windows:

When executing `biosconfig.exe`, you must run it from within the installation directory. Otherwise BIOSsconfig is not able to find its low-level management driver.

See also "BIOSconfig Error Messages" on page 51.

BIOSconfig Error Messages

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

Errors 36-49	Verify that either Microsoft IPMI driver (2003 R2) or Sun ISM driver (Pre–2003 R2) is installed correctly. 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 you 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 21

"BIOSconfig Command Overview" on page 22

IPMItool for Windows

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

To do the procedures in this section, you must have already done the following:

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

This section discusses the following:

IPMItool Overview

IPMItool is a command-line utility that reads the sensor data repository (SDR) and displays:

- Sensor values
- System Event Log (SEL)
- Field Replaceable Units (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 can be installed 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 on using IPMItool with our Management Agents, see "Generating SNMP Traps" in *Sun Server Management Agents User's Guide*.

See also:

"Sun IPMI System Management Driver 2.1" on page 54

"How to Install Sun IPMI System Management Driver 2.1 Manually" on page 54

"How to Perform Unattended Installation of the Sun IPMI System Management Driver 2.1" on page 55

"How to Verify IPMItool Installation" on page 56

Sun IPMI System Management Driver 2.1

Note – See the OS support matrix in "Installing Components" on page 7 to determine if you need this driver.

The Sun IPMI System Management Driver 2.1 for Microsoft Windows Server 2003 SP1 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 versions (including Microsoft Windows Server 2003 R2 and Microsoft Windows Server 2003 SP2) the expectation is that the Microsoft-provided IPMI driver included in their Hardware Management Module will provide the same functionality.

See also:

"IPMItool Overview" on page 53

"How to Install Sun IPMI System Management Driver 2.1 Manually" on page 54

"How to Perform Unattended Installation of the Sun IPMI System Management Driver 2.1" on page 55

"How to Verify IPMItool Installation" on page 56

▼ 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: follow these steps:

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 on Next.**
4. **Check on Yes, I have already connected the hardware and click on Next.**
5. **Select Add a new hardware device from the displayed list and click on Next.**
6. **Check on Install the hardware that I manually select from a list and click Next.**
7. **Select System Devices and click Next.**
8. **Click on the Have Disk tab to specify the filesystem location where the `ism.inf` file is stored.**
9. **Click OK to install the driver.**

See Also

"IPMItool Overview" on page 53

"Sun IPMI System Management Driver 2.1" on page 54

"How to Perform Unattended Installation of the Sun IPMI System Management Driver 2.1" on page 55

"How to Verify IPMItool Installation" on page 56

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

To perform an unattended installation of the Sun IPMI System Management Driver 2.1 for Microsoft Windows Server 2003 SP1, 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
```

See Also "IPMItool Overview" on page 53

"Sun IPMI System Management Driver 2.1" on page 54

"How to Install Sun IPMI System Management Driver 2.1 Manually" on page 54

"How to Perform Unattended Installation of the Sun IPMI System Management Driver 2.1" on page 55

▼ How to Verify IPMItool Installation

1 **To confirm that the driver has been installed, you must first 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 would be entered:**

```
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 "IPMItool Overview" on page 53

Index

V

W