Sun Fire™ Midrange System Controller Command Reference Manual

Firmware Release 5.19.0

Sun Microsystems, Inc.
www.sun.com

Part No. 819-1272-10
July 2005, Revision A

Submit comments about this document at: http://www.sun.com/hwdocs/feedback

---

Please
Recycle

Adobe PostScript™

Contents

Tables

Code Samples

Preface

This book describes the system controller command line interface for the Sun Fire™ midrange systems: E6900/E4900/6800/4810/4800/3800. The system controller is responsible for controlling system functions, environmental monitoring, domain control, and hardware control of the server. Using the command line interface, you can configure the platform and domains, power on and off domains, power supplies, fans, and other components. Access to the Solaris domain console is also possible using the command line interface.

Before You Read This Book

This book is written for system administrators or people who have a working knowledge of the Solaris™ operating environment. If you do not have such knowledge, you should first read the Solaris User and System Administrator document collections and consider UNIX® system administration training. For the documentation URL, see "Accessing Sun Documentation" on page xvi.

How This Book Is Organized

This book contains the following chapters:

Chapter 1 introduces you to the system controller and briefly describes platform administration and domain administration.

Chapter 2 explains system controller command syntax, command names, and command arguments.

Chapter 3 provides a summary in tabular form of all of the system controller commands and describes each command.

Typographic Conventions

Typeface[1]	**Meaning**	**Examples**
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`

`AaBbCc123`	What you type, when contrasted with on-screen computer output	`%` **`su`**
		`Password:`

AaBbCc123	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the *User's Guide*. These are called *class* options. You *must* be superuser to do this. To delete a file, type `rm` *filename*.

[1] The settings on your browser might differ from these settings.

Shell Prompts

Shell	**Prompt**
C shell	*machine-name*`%`
C shell superuser	*machine-name*`#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell superuser	`#`

Related Documentation

Type of Book	**Title**	**Part Number**
Release Notes	*Sun Fire Midrange Systems Firmware 5.19.0 Release Notes*	819-1270
System Administration	*Sun Fire Midrange Systems Platform Administration Manual*	819-1271
Sun Management Center	*Sun Management Center 3.5 Version 3 Supplement for Sun Fire Midrange Systems*	817-3626
Dynamic Reconfiguration	*Sun Fire Midrange Systems Dynamic Reconfiguration User Guide*	817-4585
Overview	*Sun Fire 6800/4810/4800/3800 Systems Overview Manual*	805-7362
	Sun Fire E6900/E4900 Systems Overview Manual	817-4119
Service	*Sun Fire 6800/4810/4800/3800 Systems Service Manual*	805-7363
	Sun Fire E6900/E4900 Systems Service Manual	817-4120

Accessing Sun Documentation

You can view, print, or purchase a broad selection of Sun documentation, including localized versions, at:

`http://www.sun.com/documentation`

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in this document, go to:

`http://www.sun.com/service/contacting`

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can submit your comments by going to:

`http://www.sun.com/hwdocs/feedback`

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

Sun Fire Midrange System Controller Command Reference Manual,
part number 817-7815-10

CHAPTER 1

Introduction to the System Controller

The system controller consists of the System Controller board and the system controller firmware.

The system controller firmware does the following:

- Monitors and controls the system
- Manages hardware
- Configures domains
- Provides the date and time to the Solaris operating environment
- Provides the clock signal used on all system boards
- Provides a platform console and a domain console
- Provides system monitoring and control using SNMP for use with the Sun Management Center software.

For more information on the system controller, refer to the *Sun Fire Midrange Systems Platform Administration Manual*, the *Sun Fire 6800/4810/4800/3800 Systems Overview Manual*, and the *Sun Fire E6900/E4900 Systems Overview Manual*.

Platform and Domain Administration

The platform administration function manages resources and services that are shared among the domains. With this function, you can configure how resources and services are configured and shared.

The domain administration function manages resources and services for a specific domain. With this function you can configure the domain, control the keyswitch position, and access the domain console.

For more information on the platform administration and domain administration functions, refer to the “Overview” chapter of the *Sun Fire Midrange Systems Platform Administration Manual*.

Connecting to the System Controller

To connect to the system controller, refer to the “System Controller Navigation Procedures” chapter in the *Sun Fire Midrange Systems Platform Administration Manual*.

System Controller Navigation

To navigate within the system controller, refer to the “System Controller Navigation Procedures” chapter in the *Sun Fire Midrange Systems Platform Administration Manual*.

Terminating Sessions

To terminate system controller sessions, refer to the “System Controller Navigation Procedures” chapter in the *Sun Fire Midrange Systems Platform Administration Manual*.

CHAPTER **2**

System Controller Syntax, Arguments, and Device Names

This chapter describes the following topics:

Command Syntax and Arguments

The general syntax of system controller commands is:

```
command_name [flags] [arguments]
```

Arguments are the words that follow the command name and are divided into two categories:

- Required arguments
- Optional arguments

Optional arguments always follow the option flags on the command line. In the following example, the optional *domainID* argument, `a`, follows the option flag, `-d`. The *boardname* argument, which is `sb2` in this example, is a required argument and is the board name for a CPU/Memory board.

```
schostname:SC> addboard -d a sb2
```

Command Names

Most system controller commands are in the form of *verbnoun*. Command names are in the English language. System controller commands names are case insensitive, but options for commands are not. However, items such as board names can be specified in upper case characters.

Note – When typing system controller commands, you can type the complete command name or type just enough of the command name to uniquely identify it.

Component Names

TABLE 2-1 shows the component names that are accessible from the platform. TABLE 2-2 shows the component names that are accessible from the domain. In both tables, you can type the component names in either upper case or lower case.

Note – The components in TABLE 2-1 and TABLE 2-2 depend on the system you have. For example, only the Sun Fire E6900 and 6800 systems can have six power supplies and six CPU/Memory boards.

TABLE 2-1 Components That Are Accessible From the Platform

Device Description	Device Name
Power grids*	`GRID0, GRID1`
Power supplies	`PS0, PS1, PS2, PS3, PS4, PS5`
CPU/Memory boards	`SB0, SB1, SB2, SB3, SB4, SB5`
I/O assemblies	`IB6, IB7, IB8, IB9`
Repeater boards	`RP0, RP1, RP2, RP3`
ID/Source board	`ID0`
Fan trays	`FT0, FT1, FT2, FT3`
System controller	`SSC0, SSC1`

*** Power grids are not a component but a division of a Sun Fire E6900 or 6800 system into two distinct halves of the system. Power supplies ps0, ps1, and ps2 comprise power grid 0. Power supplies ps3, ps4, and ps5 comprise power grid 1.**

TABLE 2-2 shows the components that are accessible from a domain.

TABLE 2-2 Components That Are Accessible From a Domain

Device Description	Device Name
CPU/Memory boards	`SB0, SB1, SB2, SB3, SB4, SB5`
I/O Assemblies	`IB6, IB7, IB8, IB9`

Board States for CPU/Memory Boards and I/O Assemblies

TABLE 2-3 lists the board states for the CPU/Memory boards and I/O assemblies. To determine the board state, use the `showboards` command and look under the `State` header.

TABLE 2-3 Board States

Board States	Description
Available	The board is not assigned to any domain.
Assigned	The board belongs to a domain, but the hardware has not been configured or it is not in use.
Active	The board is being actively used by the domain to which it has been assigned. You cannot reassign an active board.

If you have redundant system controllers, the SC state is identified as either the `Main` or `Spare`. The main SC provides all system resources, while the spare SC can provide system resources if the main SC fails.

The ID board, power supplies, and Repeater boards do not have a state. A dash is displayed in the Board Status field for these boards and components (TABLE 2-4). Fan trays have a state of off, low speed, and high speed.

Board Test Status

TABLE 2-4 describes the Status field of the `showboards` command.

TABLE 2-4 Status Field of the `showboards` Command

Test Status	Description
Passed	All board components passed testing.
Failed	The board failed POST and is not usable. This could also indicate corrupt or incompatible firmware.
OK	The component is functioning properly.

TABLE 2-4 Status Field of the `showboards` Command *(Continued)*

Test Status	Description
Under Test	The domain is running POST (power-on self-test); testing has been initiated using the `setkeyswitch` command, the `testboard` command, or an SNMP operation.
Not Tested	The board has not been tested.
Degraded	Certain components on the board have failed or are disabled. A board is degraded when there are still usable parts on the board.
Disabled	The component has been deconfigured from the system. It was disabled either because the component failed POST and cannot be used by the system or because the `setls` command was used to disable the component location status.
–	The slot is empty or not applicable for this device.

Domain Status

In the `showplatform` and `showdomain` commands, one of the fields in the command output is domain status. The main values of domain status are (TABLE 2-5):

TABLE 2-5 Values for the Domain Status Column in `showplatform` and `showdomain` Command Output

Domain State	Description
Powered Off	The domain is not active and the keyswitch is in the off position.
Standby	The domain is not active and the keyswitch is in the standby position.
Running POST	The domain is active and is running POST (power-on self-test).
Active	The domain is active.
Active - OpenBoot PROM	The domain is active and is running the OpenBoot™ PROM.
Active - Booting	The domain is active and is booting the Solaris operating environment.
Active - Solaris	The domain is active and is running the Solaris operating environment.
Active - Halted	The Solaris operating environment is halted in the domain.
Active - Reset	The domain has had an XIR reset and had not been rebooted.
Active - Panicking	The domain is active and is panicking. It is creating the core file.
Active - Debugger	The domain is active and the debugger is presently running.
Not Responding	The domain is not responding.
Paused due to an error	The domain is paused due to a hardware error.

Special Characters

To execute multiple commands on the same line, separate them with a semicolon (;). The following example executes both the `addboard` and `deleteboard` commands.

```
schostname:A> addboard sb2;deleteboard sb3
```

A pound sign (#) signifies the start of a comment on the current line. The following example executes the `addboard` command. Everything you type after the # and before pressing the Return key is ignored.

```
schostname:A> addboard sb2 #this text is ignored
```

Using FTP URLs

When you use certain system controller commands, such as `flashupdate`, `dumpconfig`, or `restoreconfig`, where the URL uses the FTP protocol, specify absolute paths by typing a double slash (//) after the `hostname`. Otherwise, the path is interpreted relative to the home directory of the specified user.

The following examples show the various ways to specify an FTP URL:

- URL that uses an absolute path name:

```
ftp://user:password@hostname//tmp/directory
```

  The example above references the `/tmp/directory`.

- URL that uses a relative path name:

```
ftp://user:password@hostname/tmp/directory
```

  In the example above, the path name references `/home/user/tmp/directory`.

- URL with anonymous FTP (no user name or password):

```
ftp://hostname/tmp/directory
```

The path name in this example references `/home/ftp/tmp/directory`.

Interactive Commands

Some commands may prompt for confirmation before executing the command. You can disable prompting by specifying the `-y` or `-n` flags, which answer yes or no to any prompted question.

Some commands, such as `setupplatform` and `setupdomain`, are always interactive.

- **When an interactive command prompts for input, do the following:**
  - Press the Return key to keep the current setting, displayed in brackets [ ].
  - Type a dash (-) to change the current setting to an empty string (if the input type is used to define a string).

Context-Sensitive Commands

Certain system controller commands can be run on the main SC but not on the spare SC. For a list of the commands applicable to your SC, run the `help` command on your SC in the platform shell. The help listing identifies the valid commands, based on whether the system controller is the main or the spare.

Command Line Editing

The system controller has a basic command line editor that allows you to edit the command line. The keys you press are *not* echoed on the screen.

TABLE 2-6 lists the keystrokes used to perform basic command line editing.

- To use the Control key sequences, hold down the Control key while typing the character that follows(TABLE 2-6).
- To use Escape key sequences, press and release the Escape key. Then press and release the following character (TABLE 2-6).

TABLE 2-6 Command Line Editing Capabilities

Keystroke	**Description**
Backspace (Control-h)	Erases the character before the cursor.
Delete	Erases the character before the cursor.
Control-c	Cancels editing the command line. The command is not executed.
Return	Finishes editing the line.
Control-b	Moves backward one character.
Esc-b	Moves backward one word.
Control-f	Moves forward one character.
Esc-f	Moves forward one word.
Control-a	Moves backward to the beginning of the line.
Control-e	Moves forward to the end of the line.
Esc-h	Erases from the beginning of the word to just before the cursor. Stores erased characters in a save buffer.
Control-w	Erases from the beginning of the line to just before the cursor. Stores erased characters in a save buffer.
Control-d	Erases next character.
Esc-d	Erases from the cursor to the end of the word. Stores erased characters in a save buffer.
Control-k	Erases from the cursor to the end of the line. Stores erased characters in a save buffer.
Control-u	Erases the entire line. Stores erased characters in a save buffer.

TABLE 2-6 Command Line Editing Capabilities *(Continued)*

Keystroke	Description
Control-r	Retypes the line.
Control-q	Quotes the next character. Allows you to insert Control characters.
Control-y	Inserts the contents of the save buffer before the cursor.

Command Line History

Each shell keeps a short history of all entered commands. To display the command line history, use the `history` command. After recalling the commands, you can either edit them or execute them by pressing the Return key. A history of at least eight previously typed command lines are saved.

TABLE 2-7 lists keystrokes you type to recall previously typed commands.

TABLE 2-7 Keystrokes that Recall Previously Typed Command Lines

Keystroke	Description
Control-p	Selects and displays the previous line for subsequent editing.
Control-n	Selects and displays the next line for subsequent editing.
Control-l	Displays the entire command history list.

Command Completion

Using the command completion function makes it easier for you to enter long commands. After typing part of the command, use the completion keystroke (TABLE 2-8). This action searches the list of available commands for commands beginning with the characters you typed thus far.

- If there is only one possible command, the remainder of the characters are entered for you automatically.
- If there are several possibilities, any characters that are common to all of the possible selections are filled in.
- If there are no defined commands starting with the given characters, the characters are erased until there is a minimum of one selection for the remaining characters.

TABLE 2-8 lists the keystrokes to complete a command.

TABLE 2-8 Keystrokes Used to Complete System Controller Commands

Key Sequence	Description
Control-space	Complete this command.
Control-?	Shows all possible matches.
Control-/	Shows all possible matches.

CHAPTER 3

System Controller Alphabetical Command Reference

This chapter provides a summary of the system controller commands designed for an end-user of the Sun Fire midrange systems, describes the system controller command line interface, and provides descriptions and examples for each command.

Command Summary

TABLE 3-1 lists and describes the system controller commands and how you can access them. Many commands are accessible from both the platform shell and the domain shell. The system controller commands may differ in how they are used, the effect of the command, and the scope of the command between the platform and domain shells.

TABLE 3-1 System Controller Command Summary

Command	Description	Platform Shell	Domain Shell
`addboard`	Assigns a board to a domain.	x	x
`addcodlicense`	Adds a Capacity on Demand (COD) right-to-use (RTU) license key to the COD license database.	x	
`break`	Sends a Break signal to the domain console.		x
`connections`	Displays connections to the system controller to a domain.	x	x
`console`	Connects to a domain from the platform.	x	
`deleteboard`	Unassigns a board from a domain.	x	x

TABLE 3-1 System Controller Command Summary *(Continued)*

Command	Description	Platform Shell	Domain Shell
`deletecodlicense`	Removes a Capacity on Demand (COD) right-to-use (RTU) license key from the COD license database.	x	
`disablecomponent`	Deprecated starting with the 5.15.0 release. Replaced by the `setls` command.	x	x
`disconnect`	Disconnects the current or specified connection.	x	x
`dumpconfig`	Saves the platform and domain configurations to a server.	x	
`enablecomponent`	Deprecated starting with the 5.15.0 release. Replaced by the `setls` command.	x	x
`flashupdate`	Updates the firmware.	x	
`forcepci`	Sets the default PCI mode.	x	
`help`	Provides basic help information for commands.	x	x
`history`	Shows the command history with date and time stamps.	x	x
`password`	Sets the shell password.	x	x
`poweroff`	Powers off components.	x	x
`poweron`	Powers on components.	x	x
`reboot`	Reboots the system controller.	x	
`reset`	Resets the domain in the domain shell.		x
`restartssh`	Restarts SSH server, loading and storing latest host keys.	x	
`restoreconfig`	Restores the platform and domain configurations from a server.	x	
`resume`	Exits the domain shell and resumes access to the domain console.		x
`setdate`	Sets the date and time.	x	x
`setdefaults`	Sets the configuration to default values.	x	x
`setescape`	Sets the escape sequence.	x	
`setfailover`	Changes the state of system controller failover.	x	
`setkeyswitch`	Sets the keyswitch position.	x	x
`setls`	Sets the component location status. Replaces the `enablecomponent` and `disablecomponent` commands starting with the 5.15.0 release.	x	x

TABLE 3-1 System Controller Command Summary *(Continued)*

Command	Description	Platform Shell	Domain Shell
`setupdomain`	Configures the domain.		x
`setupplatform`	Configures the platform.	x	
`showboards`	Displays the assignment information and status for all components in the system.	x	x
`showcodlicense`	Displays the current Capacity on Demand (COD) right-to-use (RTU) licenses stored in the COD license database.	x	
`showcodusage`	Displays the current usage statistics for Capacity on Demand (COD) resources.	x	x
`showcomponent`	Shows state of a component.	x	x
`showdate`	Shows the date and time.	x	x
`showdomain`	Displays the configuration and status of the domain.		x
`showenvironment`	Shows environmental information.	x	x
`showerrorbuffer`	Shows the contents of the error buffer.	x	
`showescape`	Displays the escape sequence.	x	x
`showfailover`	Displays system controller and clock failover status.	x	
`showfru`	Displays the field replaceable units (FRUs) currently installed in a Sun Fire midrange system.	x	
`showkeyswitch`	Displays the virtual keyswitch setting.	x	x
`showlogs`	Displays the logs.	x	x
`showplatform`	Displays the configuration and information for the platform and status of the domains.	x	
`showresetstate`	Shows CPUs after a reset or a RED MODE trap.		x
`showsc`	Shows the system controller version and uptime.	x	
`ssh-keygen`	Generates SSH host keys and displays host key fingerprint on the system controller.	x	
`testboard`	Tests the CPU/Memory board.	x	x

Alphabetical Listing of System Controller Commands

The following sections describe the system controller commands.

`addboard`

Assigns a board to a domain.

Scope

platform shell, domain shell

Syntax for the Platform Shell

`addboard -d` *domainID system_board_name* `[`*system_board_name* . . .`]`

`addboard -h`

Syntax for the Domain Shell

`addboard` *system_board_name* `[`*system_board_name* . . .`]`

`addboard -h`

Options/Parameters

`-d` specifies a *domainID* as an option (platform shell only).

domainID is a, b, c, or d.

`-h` displays help for this command.

system_board_name is the board to be added. Possible values for *system_board_name* are `sb0–sb5` (CPU/Memory board) or `ib6` - `ib9` (I/O assembly).

Description

Assigns *system_board_name* to the specified domain within the platform shell or to the current domain. The board state must be *Available*. To display the board state, use the `showboards` command. For more information on board states, see "Board States for CPU/Memory Boards and I/O Assemblies" on page 6.

To assign a board to a domain using a domain shell, the board name must be listed in the access control list (ACL) for the current domain. When a board is listed in the ACL, the system controller software is allowed to process `addboard` requests on that board. The platform shell does not use the ACL and can always add boards to a domain. The platform shell can also delete boards from a domain and either ignores the ACL or overrides the ACL. You configure the ACLs and also restrict the domains a board has access to using the `setupplatform` command.

If a board is assigned to an active domain, the board will not be used. To unassign a board from a domain, you must halt the Solaris operating environment in the domain. For an overview of steps to perform, see the section "Assigning and Unassigning Boards" in the "General Administration" chapter of the *Sun Fire Midrange Systems Platform Administration Manual*.

If a board is not present, the command assigns ownership of the slot to the specified domain.

See Also

`deleteboard`, `setkeyswitch`, `showboards`, `showdomain`, `showplatform`, "Assigning and Unassigning Boards" in the "General Administration" chapter of the *Sun Fire Midrange Systems Platform Administration Manual* (for a step-by-step procedure on assigning boards to a domain)

Example—Platform Shell

From the platform shell to add board name `sb2` to domain A (CODE EXAMPLE 3-1), type:

CODE EXAMPLE 3-1 `addboard` Example in the Platform Shell

```
schostname:SC> addboard -d a sb2
```

Example—Domain Shell

To assign board name `sb2` to the current domain (CODE EXAMPLE 3-2), type:

CODE EXAMPLE 3-2 `addboard` Example in the Domain Shell

```
schostname:A> addboard sb2
```

`addcodlicense`

Adds a Capacity on Demand (COD) right-to-use (RTU) license key to the COD license database.

Scope

platform shell

Syntax

`addcodlicense` *license-signature*

`addcodlicense -h`

Options/Parameters

`-h` displays help for this command.

license-signature is the COD RTU license key to be added to the COD license database.

Description

Adds the specified COD RTU license key to the COD license database on the system controller.

Note – Before you run this command, you must obtain a COD RTU license key from the Sun License Center. For details on COD RTU license keys, refer to the *Sun Fire Midrange Systems Platform Administration Manual*.

See Also

`deletecodlicense`, `showcodlicense`, `showcodusage`

Example

CODE EXAMPLE 3-3 `addcodlicense` Command Example

```
schostname:SC> addcodlicense 01:80d8a9ed:45135285:0201000000:8:00000000:0000000000000000000000
```

Note – The COD RTU license key listed above is provided as an example and is not a valid license key.

`break`

Sends a Break signal to the domain console.

Scope

domain shell

Syntax

```
break [-y|-n]
```

```
break -h
```

Options/Parameters

`-h` displays help for this command.

`-y` answers yes to the warning message. Does not prompt for confirmation.

`-n` answers no to the warning message. Does not execute this command if confirmation is requested.

Description

Sends a Break signal to the domain console. Resumes the domain console.

Note – This command pauses the Solaris operating environment.

When the Solaris operating environment is running in the domain, the usual effect of the `break` signal is to force entry into OpenBoot PROM or the debugger. The Solaris operating environment will ignore the break signal if the keyswitch is set to secure.

See Also

`resume`, `setkeyswitch`, `showkeyswitch`

Example

CODE EXAMPLE 3-4 shows using the `break` command to pause the Solaris operating environment and enter the OpenBoot PROM.

CODE EXAMPLE 3-4 `break` Command Example

```
schostname:A> break

This will suspend Solaris in domain A.
Do you want to continue? [no] yes
Type ‘go’ to resume.
debugger entered.

{1} ok
```

`connections`

Displays connections to the system controller or a domain.

Scope

platform shell, domain shell

Syntax for the Platform Shell

`connections [-d` *domainID*`]`

`connections -h`

Syntax for the Domain Shell

`connections [-h]`

Options/Parameters

`-h` displays help for this command.

`-d` *domainID* shows connections to the specified domain (a, b, c, or d).

Description

Displays hosts currently connected to the system controller. When run from the platform shell, this command shows the connections to the platform and to each domain.

When run from the domain shell, this command will only show the connection to the domain. There can only be one connection to each domain when using either SSH or Telnet.

For information about the types of connections that you can open to the system controller, see the *Sun Fire Midrange Systems Platform Administration Manual*.

See Also

`disconnect`

Example—Platform Shell

CODE EXAMPLE 3-5 `connections` Command for the Platform Shell

```
schostname:SC> connections

ID  Hostname                  Idle Time Connected On   Connected To
--  ------------------------ --------- -------------- --------------
 1  xxxxxxx                   -         May 02 00:00   Platform
 4  xxx                       -         May 02 00:21   Platform

schostname:SC>
```

TABLE 3-2 describes the headers in the output for CODE EXAMPLE 3-5.

TABLE 3-2 Header Description for the `connections` Command

Header Column in the `connections` Command	Description
ID	Connection ID. This is a unique number.
Hostname	Source of the connection for this session. If the value is `localhost`, the connection was established through the platform or through the serial port.
Idle Time	Amount of time that the SSH, Telnet, or serial port connection has been idle, without any activity. Displays a dash (-) if there has been any activity during the last 30 seconds.
Connected On	Date and time the connection was initiated. The format is `Mon dd hh:mm`.
Connected To	Shell or console this session is connected to. Possible values are `Platform` or *domainIDs* A, B, C, or D.

`console`

Connects to a domain from the platform.

Scope

platform shell

Syntax

`console [-d]` *domainID*

`console -h`

Options/Parameters

`-d` *domainID* is the domain to connect to and is `a`, `b`, `c`, or `d`. The `-d` parameter is optional and does not need to precede the *domainID*.

`-h` displays help for this command.

Description

Connect to a domain. You use the `console` command to navigate from the platform to a domain. If a password was created for connecting to this domain, you must enter the password.

Note – If the OpenBoot PROM is at the `ok` prompt or the Solaris operating environment is at the `login:` prompt, you must press the Enter key in order to see the prompt. Otherwise, you will need to wait for output to be generated by POST, the OpenBoot PROM, or the Solaris operating environment.
If the domain is hung, there will be no output. However, when a domain displays no output, this does not necessarily mean that the domain is hung.

If the domain is active (the domain is running the Solaris operating environment, OpenBoot PROM, or POST), you are connected to the domain console. Otherwise, you are connected to the domain shell. To obtain the domain shell from the domain console, see the chapter “Accessing the System Controller” in the *Sun Fire Midrange Systems Platform Administration Manual*.

See Also

`disconnect`

Examples—Platform Shell

When the domain is not active and there is no password initialized for this domain:

CODE EXAMPLE 3-6 `console` Example and No Password Set Accessing Domain B

```
schostname:SC> console b
Connected to Domain B
Domain Shell for Domain B

schostname:B>
```

When a domain is not active and a password is initialized for this domain:

CODE EXAMPLE 3-7 `console` Example and A Password Set Accessing Domain A

```
schostname:SC> console a
Enter Password:
Connected to Domain A
Domain Shell for Domain A
schostname:A>
```

When a domain is active and a password is not set for this domain:

```
schostname:SC> console a

Connected to Domain A
```

Note that no other output is seen. If the OpenBoot PROM is at the `ok` prompt or the Solaris operating environment is at the `login:` prompt, you must press the Enter key in order to see the prompt. Otherwise, you will need to wait for output to be generated by POST, the OpenBoot PROM, or the Solaris operating environment.

If the domain is hung, there will be no output. However, when a domain displays no output, this does not necessarily mean that the domain is hung.

`deleteboard`

Unassigns a board from the domain where it is currently assigned.

Scope

platform shell, domain shell

Syntax for the Platform Shell and the Domain Shell

`deleteboard` *system_board_name* `[`*system_board_name* `. . .]`

`deleteboard [-h]`

Options/Parameters

`-h` displays help for this command.

system_board_name is the board to be deleted. Values are `sb0` to `sb5` (CPU/Memory board) and `ib6` to `ib9` (I/O assembly).

Description

Unassigns a board from the domain where it is currently assigned. When you use this command from your current domain, you can only unassign boards that are assigned to the current domain. The CPU/Memory board or the I/O assembly board status must be in the *Assigned* state and the board must *not* be part of an active domain. To display the board states, use the `showboards` command. For more information on board states, see "Board States for CPU/Memory Boards and I/O Assemblies" on page 6. The board does *not* have to be in the access control list (ACL) to be unassigned from the domain.

To unassign an active board from a domain, you must halt the Solaris operating environment in the domain. Or, you can use DR to unassign an active board. For an overview of steps to perform, see the section "Assigning and Unassigning Boards" in the "General Administration" chapter of the *Sun Fire Midrange Systems Platform Administration Manual*.

If a board is not present, the command unassigns ownership of the slot from the specified domain.

See Also

`addboard`, `showboards`, “Assigning and Unassigning Boards” in the “General Administration” chapter of the *Sun Fire Midrange Systems Platform Administration Manual* (for a step-by-step procedure on unassigning boards from a domain)

Example—Platform Shell

To delete I/O assembly `ib7`, type:

CODE EXAMPLE 3-8 `deleteboard` Example Showing Deleting I/O Assembly 7

```
schostname:SC> deleteboard ib7
```

Example—Domain Shell

To delete CPU/Memory board `sb3` from the current domain, domain A, type:

CODE EXAMPLE 3-9 `deleteboard` Example Showing Deleting CPU/Memory Board 3

```
schostname:A> deleteboard sb3
```

`deletecodlicense`

Removes a Capacity on Demand (COD) right-to-use (RTU) license key from the COD license database.

Scope

platform shell

Syntax

`deletecodlicense [-f]` *license-signature*

`deletecodlicense -h`

Options/Parameters

`-f` forces the specified COD RTU license key to be deleted from the COD license database, even if the license removal will result in a license violation.

`-h` displays help for this command.

license-signature is the COD RTU license key to be removed from the COD license database.

Description

Removes a COD RTU license key from the COD license database on the system controller. For further information on COD RTU license keys, refer to the *Sun Fire Midrange Systems Platform Administration Manual*.

The system checks the number of licenses against the number of COD CPUs in use. If the license removal will result in an insufficient number of COD RTU licenses for the COD CPUs in use, the system will not delete the license key from the COD license database. If you want to delete the COD RTU license key, you must reduce the number of COD CPUs in use. You can either power off the appropriate number of domains or use dynamic reconfiguration (DR) to disconnect the appropriate number of boards.

See Also

`addcodlicense`, `showcodlicense`, `showcodusage`

Example

CODE EXAMPLE 3-10 `deletecodlicense` Command Example

```
schostname:SC> deletecodlicense 01:80d8a9ed:45135285:0201000000:8:00000000:0000000000000000000000
```

Note – The COD RTU license key listed above is provided as an example and is not a valid license key.

`disablecomponent`

The `disablecomponent` command has been deprecated starting with the 5.15.0 release and has been replaced by the `setls` command. It is suggested that you use the `setls` command even though the `disablecomponent` command is still available. For further information, see the `setls` command description.

`disconnect`

Disconnects the current or specified connection.

Scope

platform shell, domain shell

Syntax for the Platform Shell

`disconnect [`*ID*`]`

`disconnect -h`

Syntax for the Domain Shell

`disconnect [-h]`

Options/Parameters

`-h` displays help for this command.

ID (platform shell only) is the connection to be terminated. *ID* is the number displayed under the heading ID for the `connections` command.

Description

Terminates a connection to the system controller. If this command is used with no arguments, it disconnects the current session. For illustrations of the disconnect command, see the chapter "Accessing the System Controller" in the *Sun Fire Midrange Systems Platform Administration Manual*.

If the connection was initiated from another system, you will be returned to the remote host. If the session was initiated from the platform shell, you will see nothing. To obtain the prompt, press the Return key. If you are connected from the serial port, then you will return to the main menu.

See Also

`connections`

Examples

To disconnect a session from the system controller platform shell with a remote (`SSH` or `Telnet`) connection (CODE EXAMPLE 3-11), type:

CODE EXAMPLE 3-11 Disconnecting a Session From the System Controller Platform Shell

```
schostname:SC> disconnect
Connection closed by foreign host.
```

If you are connected to the system controller platform console with the serial connection, you will see the following after typing `disconnect` (CODE EXAMPLE 3-12):

CODE EXAMPLE 3-12 Disconnecting a Session From the System Controller Platform Console

```
schostname:SC> disconnect

    Type 0 for Platform Shell

    Type 1 for domain A
    Type 2 for domain B
    Type 3 for domain C
    Type 4 for domain D

    Input:
```

To disconnect a session from a system controller domain shell (CODE EXAMPLE 3-13), type:

CODE EXAMPLE 3-13 `disconnect` Example Showing Disconnecting a Session From the Domain A Shell

```
schostname:A> disconnect
Connection closed by foreign host.
```

CODE EXAMPLE 3-13 displays a direct connection to the domain and *is not* a connection to the domain made from the platform.

Note – When you disconnect from a domain shell, you will see nothing. Press the Return key to obtain the prompt.

`dumpconfig`

Saves the platform and domain configurations to a server.

Scope

platform shell

Syntax

`dumpconfig -f` *url* `[-t]`

`dumpconfig -h`

Options/Parameters

`-h` displays help for this command.

`-f` specifies the URL, which must use the FTP protocol.

> *url* is the directory containing the data files. The URL must point to a directory and not a file. Examples are:
>
> `ftp://`*userid:password@hostname*`//`*path*
>
> `ftp://`*hostname*`/`*path*
>
> `ftp://`*IPaddress*`/`*path*

`-t` saves configuration data in plain text (unencrypted) format. (Not recommended. Available for backward compatibility only.)

For details on FTP URLs, see "Using FTP URLs" on page 8.

Note – The *hostname* you enter can be the host name or a host name with a fully qualified domain name, such as *schostname* or *schostname*`.eng.sun.com`. You *cannot* use a partial domain name.

Note – The directory specified in the URL path must have write permission for the specified user. If a user is not specified, then anonymous FTP is used.

Description

Saves the platform and domain configurations to a server for recovery. By default, data is saved in encrypted format. You will be prompted to enter the password for the encrypted configuration data. This command creates two data files at the specified URL: *hostname*`.nvci` and *hostname*`.tod`.

Use this command when you complete setting up the initial configuration of the platform and the domains, and each time you modify the configuration or change the hardware configuration. For example, a configuration change occurs when you use any of the following commands: `setupplatform`, `setupdomain`, `setdate`, `addboard`, `deleteboard`, `setls`, and `password`. By running the `dumpconfig` command again, the new platform and domain configurations are saved to a server.

Caution – Invoking this command is *very important* because if the system controller board fails and you did not use this command to save the platform and domain configurations, you will have to manually reconfigure the platform and the domains. Use this command any time you change the platform or domain configuration (see the list of commands in the previous paragraph).

If you need to replace the system controller board in a single-SC configuration, the configuration information saved through the `dumpconfig` command is used to restore the platform and domain configurations to the replacement system controller board.

You *should not* use this command to revert to an old configuration. The `showplatform`, `showdomain`, `showcomponent`, and `showboards` system controller commands display all of the configuration and can be used for reference.

Caution – Use of the `-t` option causes the `dumpconfig` command to store all data in the `.ncvi` and `.tod` files in (unencrypted) plain text. Sensitive information, such as passwords and SSH private host keys, can be exposed—raising security risks.

See Also

`restoreconfig`

Example

CODE EXAMPLE 3-14 `dumpconfig` Example

```
schostname:SC> dumpconfig -f ftp://hostname/path
Enter Password:
Enter Password Again:
Created: ftp://hostname/path/hostname.nvci
Created: ftp://hostname/path/hostname.tod
```

`enablecomponent`

The `enablecomponent` command has been deprecated starting with the 5.15.0 release and has been replaced by the `setls` command. It is suggested that you use the `setls` command even though the `enablecomponent` command is still available. For further information, see the `setls` command description.

`flashupdate`

Updates the firmware on the system controller and the system boards (CPU/Memory boards and I/O assemblies). The source flash image can be on a server or another board of the same type.

Scope

platform shell

Syntax

```
flashupdate [-y|-n] -f url all|systemboards|scapp|rtos

flashupdate [-y|-n] -f url board [board . . . ]

flashupdate [-y|-n] -u

flashupdate [-y|-n] -c source_board destination_board [destination_board . . .]

flashupdate -h
```

Options/Parameters

`-h` displays help for this command.

`-y` does not prompt for confirmation.

`-n` does not execute this command if confirmation is required.

`-f` specifies a URL as the source of the flash images
url is the URL to the directory containing the flash images.
where:

`ftp://`*hostname*/*path*
`ftp://`*userid:password@hostname*//*path*
`http://`*hostname*/*path*

For details on ftp URLs, see "Using FTP URLs" on page 8.

Note – The *hostname* you enter must be the host name or a host name with a fully qualified domain name. You cannot use a partial domain name. Some examples include: *schostname*, *schostname*`.eng.sun.com.`

board is the board name.

`-c` specifies that the image should be copied from another board.

- *source_board* is the source board for the flash images.
- *destination_board* is the destination board for the flash images.

`-u` automatically updates all of the boards from the board with the highest revision.

`all` updates the system controller, all system boards (CPU/Memory boards and I/O assemblies), and the system controller real time operating system (`rtos`).

`system_boards` are all CPU/Memory boards and I/O assemblies.

`scapp` updates the current system controller. Updating the system controller reboots the system controller and disconnects all the current connections. If you have a second System Controller board installed, when you run `scapp`, also update `scapp` on the second System Controller board.

`rtos` updates the real time operating system for the system controller. If you have a second System Controller board installed, when you update the real time operating system also update the `rtos` on the second System Controller board.

Description

Caution – Any time the firmware is upgraded, check the `Install.info` file for the firmware upgrade procedure. This file is provided with your latest software release. Also refer to the *Release Notes* for any notes or special procedures. When you update the firmware on the system controller, update only one system controller at a time. DO NOT update both system controllers at the same time.

Updates the firmware on the system controller, CPU/Memory boards, and I/O assemblies. There is no firmware on Repeater boards. Boards must be powered on in order to be updated with `flashupdate`. If a board does not have power, the `flashupdate` command will do nothing and stop with an error message. Boards that are in a secure domain *cannot* be updated.

If you install a replacement board into the system:

1. Check that the firmware level of the replacement board is the same as the board you replaced. Incompatible boards are noted by the `Failed` board status in the output of the `showboards` command.

2. If the board level is not the same as the board you replaced, perform the `flashupdate` procedure on the board.

3. If a board was noted by the `Failed` state in `showboards`, after you `flashupdate` a compatible version, power off the board to clear the `Failed` state.

Before you update the flash images, make sure the firmware on the replacement board is compatible with the firmware of the board you are replacing for `scapp`, CPU/Memory board, and I/O assembly PROMs. New firmware is loaded if you reboot the domain or turn the domain keyswitch off with `setkeyswitch off` and then turn it on with `setkeyswitch on`.

- To determine the version number of the current firmware, use the `showboards -p version` command.

  The new firmware may also contain a new `rtos` image. If the `rtos` image that is currently installed is different than the `rtos` supplied with the software, you must upgrade the `rtos` image.

- To determine the version number of the current `rtos`, use the `showsc` or the `showboards -v -p version` command. The `README` file also contains the version number of the new `rtos` image.

Note – If you fail to check all versions for compatibility, the domains can crash or the system controller can fail to reboot normally.

Caution – While upgrading the firmware, heed the following precautions:

- Read the `README` and `Install.info` files before you upgrade the firmware.
- *Do not* change the keyswitch position of any domain.
- *Do not* use DR.
- *Do not* power off any boards.
- *Do not* reboot the system controller.

If you upgrade the `scapp` or `rtos` images for status:

1. It is important to watch the console during the `flashupdate` procedure.

2. Monitor the console if errors are reported.

If the images that are already installed are *incompatible* with the new images:

1. Shut down the domains before performing the firmware upgrade.

2. After you perform the firmware upgrade, reboot the domains.

If the images that are already installed are *compatible* with the new images:

1. Reboot the domains after performing the `flashupdate` procedure. Even though the firmware is compatible with the older firmware version, you must upgrade the firmware to obtain new features and bug fixes.

If you install a replacement board into the system:

1. Check if the firmware level of the replacement board is the same as the board you replaced.
2. If is it not, perform the `flashupdate` procedure on the board.

Caution – Under normal circumstances, the `flashupdate` command will complete successfully. However, if the `flashupdate` command is terminated abnormally (such as a power failure, a failed network connection, and so on), the system controller will prompt you for the URL of the images to be installed.

See Also

Sun Fire Midrange Systems Platform Administration Manual

`flashupdate` Command Platform Shell—Examples

Note – In the following examples, since the output is very long, only the command you type is listed in the code box.

Note – The `flashupdate all, flashupdate rtos`, and `flashupdate scapp` commands cause the system controller to reboot once.

- To update the active System Controller board, all the system boards, and the system controller real time operating system (`rtos`), perform this procedure from the platform console and watch the console output. This command reboots the system controller. Type:

CODE EXAMPLE 3-15 `flashupdate` Example Updating the Active System Controller Board and the System Controller Real Time Operating System

```
schostname:SC> flashupdate -f ftp://host/path all
```

- To upgrade the firmware on a replacement CPU/Memory board, `sb4` (destination_board), which is a replacement CPU/Memory board for CPU/Memory board, `sb0` (*source_board*). Type:

CODE EXAMPLE 3-16 `flashupdate` Example Updating Replacement CPU/Memory Board `sb4`

```
schostname:SC> flashupdate -c sb0 sb4
```

`forcepci`

Scope

platform shell

Syntax

`forcepci` *board_specification* `on | off | info`

Options/Parameters

`on` sets the card in the specified PCI-X board slot to PCI mode.

`off` sets the card in the specified PCI-X board slot to PCI-X mode.

`info` displays the mode (PCI-X or PCI) of the card in the specified PCI-X board slot.

board_specification identifies the target PCI-X board slot (board/port/bus).

Description

The `forcepci on` command enables you to change the mode of a card in a specified PCI-X board slot from the default (PCI-X mode–which has a maximum speed of 100 MHz), to PCI mode (which has a maximum speed of 66 MHz). Using the `forcepci off` command enables you to change the mode of the card in the specified PCI-X board slot from PCI mode back to PCI-X mode.

The changed mode value takes effect upon the next reset of the IO assembly and remains in effect until the mode is changed again, using the `forcepci` command.

Example

To set the mode of the card on PCI-X board IB6 on port 0 bus 0 to PCI mode:

```
schostname:SC> forcepci IB6/P0/B0 on
```

`help`

Provides basic help information for commands.

Scope

platform shell, domain shell

Syntax

`help` [*command_name*] | [*partial_command_name*]

`help -h`

Options/Parameters

`-h` displays help for this command.

command_name is the name of the command.

partial_command_name can be one letter of the command or a portion of the command name, such as `show`.

Description

The `help` command, without arguments, lists currently available commands. When an argument is supplied, the `help` command displays a list of commands that begin with the specified argument. If only one command is found, full help is displayed. Otherwise, a short description is displayed for each command, beginning with the specified argument.

You can also type a partial command name and basic help information will be returned for all commands matching the partial name. The partial command name must contain enough of the command name in order for the command to be recognized by help. The `help` command returns help information for all commands beginning with the characters you typed.

The `help` command is shell sensitive. It displays help information for the current shell only.

Examples

CODE EXAMPLE 3-17 displays help information on the `addboard` command.

CODE EXAMPLE 3-17 `help` Command Example Displaying Information on the `addboard` Command

```
schostname:SC> help addboard

addboard -- assign a board to a domain

Usage: addboard -d <domain> <board> ...
       addboard -h

    -d -- the domain to assign the board to
    -h -- display this help message
```

CODE EXAMPLE 3-18 displays all commands in the platform shell beginning with `show`.

CODE EXAMPLE 3-18 `help` Command Example Displaying Commands Beginning With the Word `show`

```
schostname:SC> help show

showboards        -- show board information
showcodlicense    -- show COD licenses
showcodusage      -- show COD resource usage
showcomponent     -- show state of a component
showdate          -- show the current date and time for the platform
showenvironment   -- show environmental information
showerrorbuffer   -- show the contents of the error buffer
showescape        -- show system controller escape sequence
showfailover      -- show SC Failover information
showfru           -- show FRUID information
showkeyswitch     -- show the keyswitch positions
showlogs          -- show the logs
showplatform      -- show platform configuration, information and domain status
showsc            -- show system controller version and uptime
```

CODE EXAMPLE 3-19 displays all commands in the platform shell beginning with the letter a.

CODE EXAMPLE 3-19 `help` Command Example Displaying Commands Beginning With the Letter A

```
schostname:SC> help a

addboard          -- assign a board to a domain
addcodlicense     -- add a cod license
```

`history`

Shows the command history with date and time stamps.

Scope

platform shell, domain shell

Syntax

`history [-h]`

Options/Parameters

`-h` displays help for this command.

Description

Shows the command history with date and time stamps for when the commands were executed. This command is shown for your shell and displays the last twenty commands only.

See Also

Command line editing can be used to edit the command history. For more information on how to use command line editing, see "Command Line Editing" on page 10.

Example

CODE EXAMPLE 3-20 `history` Command

```
schostname:SC> history
May 07 16:29:21 : showboards
May 07 16:29:24 : showdate
May 07 16:29:29 : history
```

`password`

Sets the password for the platform or the domain.

Scope

platform shell, domain shell

Syntax—Platform Shell

```
password [-d domainID]
```

```
password -h
```

Syntax—Domain Shell

```
password
```

```
password [-h]
```

Options/Parameters

`-d` *domainID* is domain a, b, c, or d.

`-h` displays help for this command.

Description

Sets the password for the platform or the domain. There are separate passwords for each domain and for the platform. If you set a password, entering the password is required for access to the shell or console.

Note – In redundant SC configurations starting with firmware release 5.13.0, the password on the main system controller for the platform shell is also the *same* password on the spare system controller.

It is very important to set the password for the platform and each domain even if a domain is not being used. This prevents people from creating and activating unauthorized domains.

Prior to allowing the password to be changed, the current password will be authenticated. Changed passwords take effect immediately. The old password will no longer be accepted.

You can remove the password by pressing Return at the `Enter new password` and `Enter new password` again prompts.

OpenBoot PROM passwords are different from the platform and domain passwords. The OpenBoot PROM in each domain supports the OpenBoot PROM security mode, which is a standard feature of the OpenBoot PROM software. For more information on the OpenBoot PROM security mode password, see your OpenBoot PROM documentation.

See Also

"Security Guidelines" chapter of the *Sun Fire Midrange Systems Platform Administration Manual*

Examples

You will see the following prompt (CODE EXAMPLE 3-21), when a password is already set and you type the `password` command at either the platform shell or the domain shell.

CODE EXAMPLE 3-21 `password` Command Example With a Password Already Set

```
schostname:SC> password
Enter current password:
Enter new password:
Enter new password again:
schostname:SC>
```

If currently there is not a password assigned to the shell you are entering, you will not be prompted for the current password (CODE EXAMPLE 3-22).

CODE EXAMPLE 3-22 `password` Command Example With No Password Set

```
schostname:SC> password
Enter new password:
Enter new password again:
schostname:SC>
```

`poweroff`

Powers off components.

Scope

platform shell, domain shell

Syntax for the Platform Shell

`poweroff [-y|-n] all|`*grid#*`|`*component_name* `[`*component_name* . . . `]`

`poweroff -h`

Syntax for the Domain Shell

`poweroff [-y|-n] all|`*component_name* `[`*component_name* . . . `]`

`poweroff -h`

Options/Parameters

`-h` displays help for this command.

`-y` will answer yes to any question. This option is potentially hazardous. You can forcefully power off a component with the `-y` option.

`-n` answers no to any question. You cannot forcefully power off a component with the `-n` option.

component_name is the component name.

Platform shell *component_name*(*s*):

- `all` turns off all currently controllable components.
- Power grid (grid0, grid1). The Sun Fire E6900 and 6800 systems have two power grids: `grid0` and `grid1`. Grid 1 controls power supplies ps3, ps4, and ps5. All other midrange systems have one power grid, grid0. This grid controls power supplies ps0, ps1, and ps2.
- Power supply (ps0 - ps5)
- Spare system controller (ssc0 or ssc1)
- CPU/Memory board (sb0 - sb5)
- I/O assembly (ib6 - ib9)

- Repeater board (rp0 - rp3)
- Fan tray (ft0 - ft3)

Domain shell *component_name(s)*:

- CPU/Memory board (sb0 - sb5)
- I/O assembly (ib6 - ib9)

Note – The specified board must be in the current domain.

Description

Powers off a component or a list of components.

For the domain shell, the specified board must be in the current domain.

All slots listed by `showboards` can be powered off except for the main system controller and the ID board (ID0). A powered-off component will not be monitored. Use the `showboards` command to display the power status of each board.

You can power off any component except when the board is in the `Active` state (use the `showboards` command to display the board state). When a component is in the `Active` state, a warning is displayed on the console that tells you the entire domain will go down. You are asked to confirm with a warning of the consequences.

If a board is active in a domain and you forcefully power off a device (board), the keyswitch for the domain will be set to standby. To change the keyswitch setting, use the `setkeyswitch` command.

You can power off the spare SC by running the `poweroff` command from the main SC. When the spare SC is powered off, the hot-plug LED is illuminated.

Repeater boards do not have an `Active` state. You cannot power off a Repeater board if it is being used by a domain.

You can turn off power supplies if turning off the power supply would leave sufficient power in the power grid to support the current load of the power grid. This varies with the configuration. If you attempt to power off a power supply that is required to support the current load, the power supply will not be powered off and a message stating why is displayed. You can power off redundant power supplies.

If powering off a fan would result in insufficient cooling, it cannot be powered off.

See Also

`poweron`, `setkeyswitch`, `showboards`

Examples

To power off CPU/Memory board, `sb2`, from the platform shell (CODE EXAMPLE 3-23), type:

CODE EXAMPLE 3-23 `poweroff` Command Example Showing Powering Off `sb2`

```
schostname:SC> poweroff sb2
```

To power off all power supplies, fan trays, and system boards (CODE EXAMPLE 3-24), type:

CODE EXAMPLE 3-24 `poweroff` Command Example Showing Powering Off All Power Supplies, Fan Trays, and System Boards in the Platform Shell

```
schostname:SC> poweroff all
```

Note – From the domain shell, you cannot power off power supplies, fan trays, Repeater boards, or power grids. To power off these components, connect to the platform shell.

`poweron`

Powers on components.

Scope

platform shell, domain shell

Syntax for the Platform Shell

`poweron all`|*grid#*|*component_name* [*component_name* . . .]

`poweron -h`

Syntax for the Domain Shell

`poweron all`|*component_name* [*component_name* . . .]

`poweron -h`

Options/Parameters

`-h` displays help for this command.

Platform shell *component_name*(*s*):

- `all` is all power supplies and boards.
- Power grid (grid0, grid1). The Sun Fire E6900 and 6800 systems have two power grids: grid0 and grid1. Grid 1 controls power supplies ps3, ps4, and ps5. All of the other mid-range systems have one power grid, `grid0`. This grid controls power supplies ps0, ps1, and ps2.
- Power supply (ps0 - ps5)
- Spare system controller (ssc0 or ssc1)
- CPU/Memory board (sb0 - sb5)
- I/O assembly (ib6 - ib9)
- Repeater board (rp0 - rp3)
- Fan tray (ft0 - ft3)

Domain shell *component_names*:

- `all` powers on all CPU/Memory boards and I/O assemblies in the domain
- CPU/Memory board (sb0 - sb5)
- I/O assembly (ib6 - ib9)

The specified board must be in the current domain.

Description

Powers on a component or a list of components. You *must* specify *component_name*(*s*).

From the domain, only components that are assigned to the domain can be powered on. To power on a component, the power grid power must also be on.

Fan trays are automatically powered on when a power supply is powered on. Fan trays obtain power from the power grid(s).

If the spare system controller was powered off, you can power on the spare system controller by running the `poweron` command from the main system controller. The spare system controller will power on automatically when the system is turned on or when the spare system controller is inserted into a system that has power. When the spare system controller is powered on, the hot plug LED is not illuminated.

See Also

`poweroff`,`setkeyswitch`, `showboards`

Examples

To power on CPU/Memory board, sb2 from the platform shell (CODE EXAMPLE 3-25), type:

CODE EXAMPLE 3-25 `poweron` Example Powering On `sb2`

```
schostname:SC> poweron sb2
```

To power on CPU/Memory boards and I/O assemblies in the domain (CODE EXAMPLE 3-26), type:

CODE EXAMPLE 3-26 `poweron` Example Showing Powering On All CPU/Memory Boards and I/O Assemblies in Domain A

```
schostname:A> poweron all
```

Note – From the domain shell, you cannot power on power supplies, Repeater boards, fan trays, or power grids. To power on these components, use the platform shell.

`reboot`

Reboots the system controller.

Scope

platform shell

Syntax

```
reboot [-y|-n]
```

```
reboot -h
```

Options/Parameters

`-y` does not prompt for confirmation.

`-n` does not execute this command. Confirmation is requested.

`-h` displays help for this command.

Description

Reboots the system controller. The `reboot` command will not change any configuration settings. Active domains will continue to run.

The keyswitch may be set to off if the system controller reboots while a keyswitch operations is in progress.

Caution – Rebooting interrupts any current operation. This includes keyswitch operations, rebooting the Solaris operating environment, testing boards, SNMP, and so on.

Examples

To reboot the system controller (CODE EXAMPLE 3-27), type:

CODE EXAMPLE 3-27 `reboot` Command Example

```
schostname:SC> reboot
Are you sure you want to reboot the System Controller now? [no]
```

`reset`

Resets the domain.

Scope

domain shell

Syntax

```
reset [-y|-n][-x|-a]
```

```
reset -h
```

Options/Parameters

With no arguments, typing `reset` is the same as typing `reset -x`.

`-y` answers yes to the question asked (executes the command).

`-n` answers no to the question asked (does not execute the command).

`-h` displays help for this command.

`-x` resets via XIR (externally initiated reset). Use XIR to try to obtain diagnostic data (default).

`-a` reset is equivalent to the OpenBoot PROM `reset-all` command.

Description

This command resumes the domain console. Resetting the domain is not allowed if the keyswitch is in the secure position, nor is it possible if the keyswitch is configured in either the off or standby position. Before you can reset the domain, change the keyswitch position to on.

By default, `reset` uses XIR (externally initiated reset) to reset the CPUs in the domain. The XIR forces control of the domain into the OpenBoot PROM and begins the OpenBoot PROM error reset recovery actions. The error reset recovery actions preserve most domain states to allow collecting data needed for debugging the hardware and software, including a Solaris operating environment core file. The OpenBoot PROM error reset recovery actions are controlled by setting the OpenBoot PROM `error-reset-recovery` configuration parameter. For the definition of this domain parameter and the various settings, see TABLE 3-8 in the `setupdomain` command description.

Note that the OpenBoot PROM `reset` command *does not* generate a core file as this command does.

You cannot reset a domain that has been paused. The domain is paused automatically when hardware detects an error. Messages on the domain console indicate that the domain is paused.

To take a domain out of the paused state:

1. Turn the keyswitch off with `setkeyswitch off`.
2. Turn the keyswitch on with `setkeyswitch on`.

See Also

`resume`, `setkeyswitch`, `setupdomain`, `showdomain`, `showkeyswitch`, `showresetstate`, "Domain Not Responding" section in the "Troubleshooting" chapter in the *Sun Fire Midrange Systems Platform Administration Manual.* This section describes how to recover from a hung or paused domain.

Examples

To reset the domain (from a domain shell), type the following (CODE EXAMPLE 3-28). This command uses XIR (externally initiated reset) to try to obtain diagnostic data.

CODE EXAMPLE 3-28 `reset` Command Example From Domain A

```
schostname:A> reset
```

To perform the equivalent operation as the OpenBoot PROM `reset-all` command (CODE EXAMPLE 3-29), type:

CODE EXAMPLE 3-29 `reset` -a Command Example From Domain A

```
schostname:A> reset -a
```

`restartssh`

Restarts the SSH server.

Scope

Platform shell.

Syntax

```
restartssh [-h]
```

Options/Parameters

`-h` displays help for this command.

`-y` answers `yes` to the informational message. Does not prompt for confirmation.

`-n` answers `no` to the informational message. Does not execute the command if confirmation is required.

Description

To run this command, SSH must be enabled using the `setupplatform` command.

If you have generated new host keys using `ssh-keygen`, you must restart the SSH server before the new host keys can take effect. By restarting the server, the keys are loaded into memory and stored in the SSH server's dedicated memory structure.

When restarting the SSH server, all existing SSH connections are closed. The command posts an informational message, asking for confirmation before actually restarting the SSH server. You can skip confirmation by specifying the `-y` or `-n` switch.

If you have issued the command over an SSH connection, the connection terminates when the SSH server restarts. Since the process only takes seconds, you can re-establish the SSH connection immediately.

See Also

`ssh-keygen`

`restoreconfig`

Restores the platform and domain configurations from a server.

Scope

platform shell

Syntax

`restoreconfig [-y|-n] -f` *url*

`restoreconfig -h`

Options/Parameters

`-h` displays help for this command.

`-y` does not prompt for confirmation.

`-n` does not execute this command if confirmation is required.

`-f` specifies a URL, which must use the FTP protocol.

> *url* is the directory containing the data files. The URL must point to a directory and not a file. Examples are:
>
> `ftp://`*userid:password@hostname//path*
>
> `ftp://`*hostname/path*
>
> `ftp://`*IPaddress/path*

For details on FTP URLs, see "Using FTP URLs" on page 8.

Note – The *hostname* you enter can be the host name or a host name with a fully qualified domain name, such as *hostname* or *hostname*`.eng.sun.com`. You *cannot* use a partial domain name.

Description

Restores the platform and domain configurations from a server, using the two data files created by the `dumpconfig` command, *schostname*`.nvci` and *schostname*`.tod`. The files are located at the specified URL, which must point to a directory and use the FTP protocol. By default, the configuration data stored is encrypted. To restore the encrypted data, you will be prompted to enter the same password that was used (for encrypting the configuration data) when creating the dumpfile.

Caution – The `dumpconfig` command should have been executed when you set up the system. It is also used in the procedure for replacing a failed system controller in single-SC configurations. For general instructions on how to use `dumpconfig`, see "To Use `dumpconfig` to Save Platform and Domain Configurations" in the chapter "System Power On and Setup" in the *Sun Fire Midrange Systems Platform Administration Manual*.

This command requires that all domains be powered off with the `setkeyswitch off` command. The main and spare system controllers are automatically rebooted when the configuration is restored. The date and time must be set after the configuration has been restored and the system controller has been rebooted. You also need to activate all domains with `setkeyswitch on`, since this command restores all domains with the keyswitch in the off position.

Note – Check the configuration of the platform and the domains before activating any domains to ensure that the desired configuration has been restored. In the platform shell, use `showplatform`, `showboards`, and `showdate`. In each domain shell, use `showdomain`, `showboards`, and `showdate`.

See Also

`dumpconfig`, `setkeyswitch`, `setdate`, `showboards`, `showdate`, `showdomain`, `showplatform`

Example

CODE EXAMPLE 3-30 shows an example of the `restoreconfig` command with the required `-f` option.

CODE EXAMPLE 3-30 `restoreconfig` Example

```
schostname:SC> restoreconfig -f ftp://hostname/path
Enter Password:
Enter Password Again:

This will restore all the platform and domain configurations and passwords.
```

CODE EXAMPLE 3-30 `restoreconfig` Example *(Continued)*

```
The system controller will be rebooted when the restore is complete.
The date will need to be set in the platform and each domain.

Do you want to restore the system controller configuration now? [no] y

Retrieving: ftp://hostname/path/schostname.nvci
Verifying data.
Retrieving: ftp://hostname/path/schostname.tod
Verifying data.
NVCI has been restored.
TOD has been restored.
Enter new password:
Enter new password again:
May 19 11:25:14 hostname Platform.SC: Stopping all services on this SC
Restore complete. The system controller is being rebooted.
The date will need to be set in the platform and each domain.

Rebooting. All telnet connections closed. Reestablish any needed connections.
May 19 11:25:14 hostname Platform.SC: All services on this SC have been stopped.

Software Reset . . .
```

`resume`

Exits the domain shell and resumes access to the domain console.

Scope

domain shell

Syntax

```
resume [-h]
```

Options/Parameters

`-h` displays help for this command.

Description

Exits the domain shell and resumes access to the domain console. This command requires an active domain. If the domain is not active, there is no domain console and there is nothing to resume. When you connect to the domain console, there may not be any output displayed. If either the OpenBoot PROM or the Solaris operating environment is running in the domain, press the Return key to get the prompt. Otherwise, POST output is displayed (may be a delay in displaying POST output).

Other system controller commands that also resume the domain console are listed in the following section.

See Also

`break`, `reset`, `setkeyswitch`

Example

CODE EXAMPLE 3-31 `resume` Command Example in Active Domain A

```
schostname:A> resume
```

Note that no other output is seen. If the OpenBoot PROM is at the `ok` prompt or the Solaris operating environment is at the `login:` prompt, you must press the Enter key in order to see the prompt. Otherwise, you will need to wait for output to be generated by POST, the OpenBoot PROM, or the Solaris operating environment.

If the domain is hung, there will be no output. However, when a domain displays no output, this does not necessarily mean that the domain is hung.

`setdate`

Sets the date and time.

Scope

platform shell, domain shell

Syntax

`setdate [-v] [-t` *time zone*`]` [*mmdd*]*HHMM*

`setdate [-v] [-t` *time zone*`]` *mmddHHMM* `[[`*cc*`]`*yy*`][.`*SS*`]`

`setdate [-v] [-r` *datehost*`]`

`setdate [-v] -t` *time zone*

`setdate [-v] -t GMT<+|->` *offset from GMT* (TABLE 3-3)

`setdate -h`

Options/Parameters

`-t` *time zone* sets the time zone using the time zone abbreviation (TABLE 3-3). Only non-daylight savings time zones can be specified. If you are in an area with daylight time or summer time, the time and time zone are adjusted automatically for daylight time or summer time.

`-t GMT<+|->`*offset* from Greenwich Mean Time—*GMT* (TABLE 3-3).

mm is the month number.

dd day is the number in the month.

HH is the hour number (24-hour clock).

MM is the minute number.

cc is the century minus 1.

yy is the last two digits of the year number.

SS is the second number. For the century minus 1 value, use 21 (Twenty first century) minus 1, which yields 20.

`-r` *datehost* sets the current time using `rdate`. The host must be a valid system name and the system must support `rdate` requests. The *hostname* you enter can be the host name or the host name with a fully qualified domain name, such as *schostname* or *schostname*`.eng.sun.com`. You *cannot* use a partial domain name.

`-v` is verbose mode. Displays detailed information about the time zone that is set.

`-h` displays help for this command.

TABLE 3-3 Time Zone Abbreviations, Time Zone Name, and Offsets From Greenwich Mean Time

Time Zone Abbreviation	Time Zone Name	Offset From Greenwich Mean Time (GMT)
ACT	Australian central time	GMT+9.5
AET	Australian eastern time	GMT+10
AGT	Argentina standard time	GMT-3
ART	Arabic (Egypt) standard time	GMT+2
AST	Alaska standard time	GMT-9
BET	Brazil eastern time	GMT-3
BST	Bangladesh standard time	GMT+6
CAT	Central African time	GMT+2
CNT	Canada Newfoundland time	GMT-3.5
CST	Central standard time	GMT-6
CTT	China Taiwan time	GMT+8
EAT	Eastern African time	GMT+3
ECT	European central time	GMT+1
EET	Eastern European time	GMT+2
EST	Eastern standard time	GMT-5
HST	Hawaii standard time	GMT-10
IET	Indiana eastern standard time	GMT-5
IST	India standard time	GMT+5.5
JST	Japan standard time	GMT+9
MET	Middle East time	GMT+3.5
MIT	Midway Islands time	GMT-11
MST	Mountain standard time	GMT-7
NET	Near East time	GMT+4

TABLE 3-3 Time Zone Abbreviations, Time Zone Name, and Offsets From Greenwich Mean Time *(Continued)*

Time Zone Abbreviation	Time Zone Name	Offset From Greenwich Mean Time (GMT)
NST	New Zealand standard time	GMT+12
PLT	Pakistan Lahore time	GMT+5
PNT	Phoenix standard time	GMT-7
PRT	Puerto Rico and U S. Virgin Islands time	GMT-4
PST	Pacific standard time	GMT-8
SST	Solomon standard time	GMT+11
UTC	Universal Time Coordinated	GMT+0
VST	Vietnam standard time	GMT+7

Description

Sets the date and time for the platform and domains. This command, when invoked from the platform, will have no effect on the date and time in each domain and vice-versa. You can set up to five different times and time zones; one time and time zone for the platform and different times and time zones for each of the four domains.

If your time zone area is using daylight or summer time, this is set automatically.

Note – You cannot set the date from the system controller in a domain while the Solaris operating environment is running. To set the date while the Solaris operating environment is running, use the Solaris operating environment `date` command.

After you set the date and time, you can use the SNTP server to keep the date and time synchronized. For details, refer to the *Sun Fire Midrange Systems Platform Administration Manual*.

See Also

`showdate`, `setupplatform`

Examples

When you type the `setdate` command from the platform, this sets the date and time for the platform. When you type the `setdate` command from the domain, this sets the date and time for the domain.

To set the date and time on the platform to Thursday, April 20, 2001, at 18 hours 15 minutes and 10 seconds, type:

CODE EXAMPLE 3-32 `setdate` Command Example in the Platform Shell

```
schostname:SC> setdate 042018152001.10
Thu Apr 20 18:15:10 PST 2001
```

To set the date from a date host:

CODE EXAMPLE 3-33 `setdate -r` Command Example Setting the Date From a Date Host

```
schostname:SC> setdate -r datehost
Thu Apr 20 18:15:10 PST 2001
```

To set the time zone to Pacific Standard Time (PST), using the offset from Greenwich mean time—GMT, and the date and time on the platform to Thursday, April 20, 2001, at 18 hours 15 minutes and 10 seconds (TABLE 3-3), type:

CODE EXAMPLE 3-34 `setdate -t` Command Example Setting the Time Zone to Pacific Standard Time Using the Offset From Greenwich Mean Time

```
schostname:SC> setdate -t GMT-8 042018152001.10
Thu Apr 20 18:15:10 PST 2001
```

To set the time zone to Eastern Standard Time (EST), using the time zone abbreviations, and the date and time on the platform to Thursday, April 20, 2001, at 18 hours 15 minutes and 10 seconds (TABLE 3-3), type:

CODE EXAMPLE 3-35 `setdate -t` Command Example Setting the Date and Time Zone to Eastern Standard Time Using Time Zone Abbreviations

```
schostname:SC> setdate -t EST 042018152001.10
Thu Apr 20 18:15:10 EST 2001
```

To set just the time zone for Japan Standard Time using offsets from Greenwich Mean Time—GMT (TABLE 3-3) and *not* the date and time, type:

CODE EXAMPLE 3-36 `setdate -t` Command Example Setting the Time Zone to Japan Standard Time Using the Offset From Greenwich Mean Time

```
schostname:SC> setdate -t GMT+9
Thu Apr 20 18:15:10 GMP+9 2001
```

To set just the time zone for European Central Time using the time zone abbreviations (TABLE 3-3) and *not* the date and time, type:

CODE EXAMPLE 3-37 `setdate -t` Command Example Setting the Time Zone to Eastern Central Time Using Time Zone Abbreviations

```
schostname:SC> setdate -t ECT
Thu Apr 20 18:15:10 ECT 2001
```

`setdefaults`

Sets the default configuration values.

Caution – This is a destructive command. Use with caution and care.

Scope

platform shell, domain shell

Syntax

```
setdefaults [-y|-n] [-c] [-p platform]
```

```
setdefaults -h
```

Options/Parameters—Platform Shell

`-y` does not prompt for confirmation. The option sets the defaults for the platform shell and each domain shell.

`-n` does not execute the command if confirmation is requested.

`-c` keeps Capacity on Demand (COD) right-to-use (RTU) license keys stored in the COD license database. This option does not preserve the number of any instant access CPUs (headroom) enabled and COD RTU licenses reserved for domains.

`-h` displays help for this command.

`-p platform` resets the defaults for the platform only.

Options/Parameters—Domain Shell

`-y` does not prompt for confirmation.

`-n` does not execute the command if confirmation is requested.

`-h` displays help for this command.

Description

Caution – This command requires that all domains are inactive (not running the OpenBoot PROM, POST, or the Solaris operating environment) and the keyswitch be set to off.

When you run this command from the platform shell without options, the platform and domain values are set to the default values. When you run this command from the domain shell without options, only the current domain values are reset to the default values.

If a password is set, you will need to type the password in order to set the default values. When you run this command from the platform shell, the password for the platform shell is required. When you run this command from a domain shell, the password for the domain shell is required. There are separate passwords for the platform shell and each domain shell.

In addition, this command does the following:

- Sets the system controller to the default values. This command resets only the system controller setting. It does not affect the domain's configuration in the OpenBoot PROM.

  If a spare System Controller board is present, SC failover is automatically enabled. Otherwise, SC failover is disabled.
- Deletes all COD RTU licenses and resets the instant access CPU (headroom) quantity and domain COD RTU license quantity to zero (0), only if you do not specify the `-c` option.
- Erases all platform and domain configurations (platform).
- Removes passwords (domain and platform).
- Resets the connection type to *none*.
- Erases SSH host keys.
- Reboots both the main and spare system controllers after the defaults are set.

Caution – Make sure that you have a serial connection to the SC before running the `setdefaults` command. After `setdefaults` has been run, remote connectivity to the system controller is disabled. Once remote connectivity (either SSH or Telnet) has been disabled, your only access to the SC platform shell (and the `setupplatform` command) is through a serial connection.

Example

CODE EXAMPLE 3-38 `setdefaults` Example

```
schostname:SC> setdefaults

You are about to reset all configuration data to default values.
All domain configurations, ACLs, passwords,data buffers, network
settings, and SSH host keys will be lost.

The system controller will be REBOOTED after the defaults are set.

Do you want to restore the default values and reboot now? [no]
```

- If you answer yes, the system controller is rebooted after the defaults are set.
- If you answer no, the default values are not restored.
- If a password is set for the platform shell, the password will be required.

`setescape`

Set the sequence of characters used to switch from the Solaris or OpenBoot PROM prompt at the domain console to the system controller prompt at the domain shell.

Syntax

`setescape` *escapechars*

`setescape -h`

Options/Parameters

`-h` displays help for this command.

escapechars is the escape sequence, up to five alphanumeric characters in length. The default sequence is a pound sign (#) followed by a period (.), that is, `#.`

Description

Sets the sequence of characters used to switch from the Solaris or OpenBoot PROM prompt at the domain console to the system controller prompt at the domain shell.

If you are typing at the domain console and type the first character of the escape sequence. The default is the pound sign followed by a period (`#.`). There is a one second delay before the character appears on the screen. This delay occurs because the system waits for one second to see if the next character in the escape sequence is about to be typed. If the next character is typed, the system waits up to one second for the next character and so on. If you type all the characters in the escape sequence, the domain shell prompt appears. If you do not, the characters belonging to the escape sequence that were typed are output to the screen.

It is recommended that you choose an escape sequence that does not start with a sequence of characters that is frequently typed at the console, otherwise the delay between you typing the keys and the character appearing on the screen may be confusing and affect your typing.

See Also

`showescape`

Examples

To set the escape characters to `~~~..` type:

```
schostname:SC>setescape ~~~..
```

Note – To prevent confusion, avoid duplication of the escape sequences pre-defined in the OpenSSH protocol.

To reset the escape characters to `#.` (the default) type:‘

```
schostname:SC>setescape "#."
```

Note – As `#` is the comment character for the platform shell and domain shell, the sequence must be enclosed in quotes.

`setfailover`

Change the state of system controller (SC) failover.

Scope

platform shell

Syntax

```
setfailover [-y|-n] on|off|force
```

```
setfailover -h
```

Options/Parameters

`-y` does not prompt for confirmation.

`-n` does not execute the command if confirmation is requested.

`on` enables failover for systems that previously had failover disabled due to a failover or an operator request.

`off` disables failover. This option prevents a failover until the failover feature is re-enabled.

`force` causes a forced failover to the spare SC.

`-h` displays help for this command.

Description

This command enables you to control automatic or manual SC failover. Be aware that if you force a failover using this command, SC failover is disabled after the manual failover occurs. For further information on SC failover, refer to the "System Controller Failover" chapter in the *Sun Fire Midrange Systems Platform Administration Manual*.

See Also

`setupplatform`, `showfailover`, `showlogs`, `showplatform`, `showsc`

Examples

To manually failover from the main SC to the spare, type:

CODE EXAMPLE 3-39 `setfailover` Command Example (run on the Spare SC) of Manual Failover

```
schostname:sc> setfailover force

SC: SSC0
Spare System Controller
SC Failover: enabled and active.
Clock failover enabled.

This will abruptly interrupt operations on the other System Controller.
This System Controller will become the main System Controller.

Do you want to continue? [no] yes
Oct 26 00:10:33 schostname Platform.SC: SC Failover: becoming main SC ...
Oct 26 00:10:37 schostname Platform.SC: Chassis is in single partition mode.
Oct 26 00:10:42 schostname Platform.SC: Main System Controller
Oct 26 00:10:42 Oct 26 00:10:42 schostname Platform.SC: Added logical IP address
xxx.xxx.xxx.xxx

Oct 26 00:10:43 schostname Platform.SC: SC Failover: disabled

schostname:SC>
```

To disable SC failover, type:

CODE EXAMPLE 3-40 `setfailover` Command Example Disabling Failover

```
schostname:SC> setfailover off
SC Failover: disabled
```

To re-enable SC failover, type:

CODE EXAMPLE 3-41 `setfailover` Command Example Re-enabling Failover

```
schostname:SC> setfailover on
SC Failover: enabled and active.
```

`setkeyswitch`

Changes the position of the virtual keyswitch to the specified value.

Scope

platform shell, domain shell

Syntax for the Platform Shell

```
setkeyswitch [-y|-n] -d domainID off
```

```
setkeyswitch -h
```

Syntax for the Domain Shell

```
setkeyswitch [-y|-n] [off|standby|on|diag|secure]
```

```
setkeyswitch -h
```

Options/Parameters

`-h` displays help for this command.

`-y` does not prompt for confirmation.

`-n` does not execute this command is confirmation is requested.

`-d` *domainID* (platform shell only) is the domain ID of the domain you want power off (`a`, `b`, `c`, or `d`).

For `setkeyswitch` parameters and descriptions, see TABLE 3-4.

TABLE 3-4 Description of the `setkeyswitch` Transition

Value	Description
`off`	Changes the board state of all the boards belonging to a domain to the Assigned board status, and places the boards in low-power mode, which allows you to remove the boards from the system. The domain will not be initialized at system power on.
`standby`	Changes the board state of all the boards belonging to a domain to the Assigned board state and turns on the boards. The domain will not be initialized at system power on, but the boards that comprise the domain will be powered on.
`on`	Powers on and initializes the domain. The system controller brings the domain into OpenBoot PROM through POST. If the OpenBoot PROM has `auto-boot?` set to `true`, then the Solaris operating environment boots automatically.
`diag`	Similar to the `setkeyswitch on` command except the POST verbose mode and the POST diagnostic level are set to max.
`secure`	Similar to the `setkeyswitch on` command except that the `break` command and the `reset` commands are ignored. CPU/Memory board and I/O assembly flash PROM updates are not allowed.

TABLE 3-5 shows the results when you change a keyswitch setting.

TABLE 3-5 Results of Changing the Keyswitch From the Current Setting to a New Setting

Current Setting	New Setting	Result
off	off	Keyswitch position not changed.
off	standby	Powers on all of the boards. The boards belonging to a domain should already be in the Assigned state. The domain will not be initialized at system power on.
off	on	Powers on and initializes the domain. The domain will go through POST and the OpenBoot PROM. The Solaris operating environment boots automatically unless you have set the OpenBoot PROM `auto-boot?` set to false.
off	diag	Similar to on, except POST verbose and diagnostic level is to set to maximum. The domain will boot from the `diag-device`.
off	secure	Similar to on, except the `reset` and `break` commands are ignored by the domain shell.
standby	off	All boards will be powered off (set to low-power mode and all monitoring of the boards will stop). The domain is shut down.

TABLE 3-5 Results of Changing the Keyswitch From the Current Setting to a New Setting *(Continued)*

Current Setting	New Setting	Result
standby	on	Similar to setting the virtual keyswitch from off to on except that you do not have to wait for the system controller to turn on the boards and prepare them.
standby	standby	Keyswitch position not changed.
standby	diag	Similar to on, except POST verbose and diagnostic level is to set to maximum. The domain will boot from the `diag-device`.
standby	secure	Similar to on, except the `reset` and `break` commands are ignored by the domain shell.
on	off	All boards will be powered off (set to low-power mode and all monitoring of the boards will stop). The domain is shut down.
on	standby	The domain becomes inactive. All of the boards remain powered on.
on	on	Keyswitch position not changed.
on	diag	Similar to the `setkeyswitch on` command except the POST verbose mode is set to on and the POST diagnostic level is set to maximum. This change is setting has no effect on a running domain.
on	secure	`break` and `reset` are ignored.
diag	off	All boards will be powered off (set to low-power mode and all monitoring of the boards will stop). The domain is shut down.
diag	standby	The domain becomes inactive. All of the boards remain powered on.
diag	on	Has no effect on a running domain. This operation only changes the keyswitch position and does not perform any other function since the domain is already running.
diag	diag	Keyswitch position not changed.
diag	secure	`break` and `reset` are ignored.
secure	off	All boards will be powered off (set to low-power mode and all monitoring of the boards will stop). The domain is shut down.
secure	standby	The domain becomes inactive. All of the boards remain powered on. The `break` and `reset` commands are no longer ignored.
secure	on	`break` and `reset` are not ignored.
secure	diag	This change is setting has no effect on a running domain.
secure	secure	Keyswitch position not changed.

Description

Each domain has a virtual keyswitch with five positions: off, standby, on, diag, and secure. The `setkeyswitch` command changes the position of the virtual keyswitch to the specified value. The virtual keyswitch replaces the need for a physical keyswitch for each domain.

If it is not possible to access the domain shell, set the keyswitch position for the domain to off from the platform shell. Setting the keyswitch position to off for a domain without first halting the Solaris operating environment may cause problems. If the domain is running the Solaris operating environment, the `setkeyswitch` command parameters off and standby require confirmation.

If the domain is already powered on, the `setkeyswitch on`, `setkeyswitch diag`, and `setkeyswitch secure` commands will change only the position of the virtual keyswitch. If components making up a domain such as a I/O assembly or CPU/Memory board need to be powered on, the system attempts to power on these boards or devices also.

When the keyswitch position changes to on, secure, or diag from any keyswitch position, the domain console is resumed.

If a failure occurred during keyswitch transition, the keyswitch position is reset to `off` or `standby` after recovering from the failure.

The system controller maintains the position of each keyswitch even during power failures or physical power cycles. When you power on the system after either a power failure or system power off, the system controller activates any domains that were active when the power went off.

See Also

`resume`, `showkeyswitch`

Examples

To turn the virtual keyswitch on and power on the domain (CODE EXAMPLE 3-42), type:

CODE EXAMPLE 3-42 `setkeyswitch on` Example

```
schostname:A> setkeyswitch on
```

To shut down the domain by turning the virtual keyswitch off (CODE EXAMPLE 3-43), type:

CODE EXAMPLE 3-43 `setkeyswitch off` Example

```
schostname:A> setkeyswitch off
```

To shut down domain B from the platform shell (CODE EXAMPLE 3-44), type:

CODE EXAMPLE 3-44 `setkeyswitch off` Example Shutting Down Domain B From the Platform Shell

```
schostname:SC> setkeyswitch -d b off
```

Caution – CODE EXAMPLE 3-44 does not properly shut down the Solaris operating environment in the domain before turning the keyswitch in domain B to the off position.

`setls`

Sets the component location status.

Scope

Platform shell, domain shell

Syntax

`setls -s` *new_status* `-l` *location*

`setls -h`

Options/Parameters

`-h` displays help for this command.

`-s` *new_status* sets the location status of a component:

- `enable` – Enables the specified component location.
- `disable` – Disables the specified component location.

`-l` *location* specifies the component location:

- *slot/port/physical_bank/logical_bank* for a CPU/Memory board
- *slot/port/bus* for an I/O assembly
- *slot/card* for an I/O assembly

Description

Use this command to control whether components in a particular location are configured into a system. The location status of a component can be set to one of the following states:

- Enabled – The component residing in the specified location is configured into the system, subject to its component health status (CHS). For details on component health status, refer to the *Sun Fire Midrange Systems Platform Administration Manual.*

  In some cases a disabled component cannot be re-enabled by using the `setls` command. If a a disabled component has a POST status of `chs`, as indicated in `showcomponent` command output, the component cannot be configured into the system. Contact your service provider for further service action.

- Disabled – The component residing in the specified location is not configured into the system.

Note – The location status is updated at the next domain reboot, board power cycle, or POST execution. For example, POST runs automatically whenever you perform a `setkeyswitch on` or `off` operation.

When you disable the location of a component, its subcomponent locations are also disabled. For example, if you disable the location of a CPU slot, the memory locations that are controlled by that CPU are also automatically disabled.

Similarly, when you enable the location of a component, its subcomponent locations are also enabled, except when the subcomponent locations were previously disabled on an individual basis by using the `setls` command. The subcomponent locations cannot be enabled automatically through the parent component location. Each subcomponent location must be enabled individually by using the `setls` command.

Note the following about the component location status:

- If a component location is disabled in the platform shell, that component location is also disabled in all domains.

  The components in the disabled locations are not configured into the system.

- If a component location is disabled in a domain shell, that component location is disabled only in the domain.

  The components in disabled locations are not configured into the system.

  If you change the status of a component location in a domain, the change applies only to that domain. This means that if you move the component to another location or another domain, the component does not retain the same location status.

- If a component location is disabled in the platform, that component location cannot be enabled in the domains. Or, if a component location is disabled in a domain, that component location cannot be enabled in the platform.

  In both situations, the components in the disabled locations are not configured into the system.

Component locations can be the following (see TABLE 3-6 and TABLE 3-7):

- Ports (CPU on a CPU/Memory board and I/O controller on an I/O assembly)
- Physical and logical memory banks
- I/O buses
- I/O cards

TABLE 3-6 *location* Descriptions for a CPU/Memory Board

Board or Device	Component Location
CPU/Memory board *slots*	SB0, SB1, SB2, SB3, SB4, SB5
Ports on the CPU/Memory board	P0, P1, P2, P3
Physical memory banks on CPU/Memory boards	B0, B1
Logical banks on CPU/Memory boards	L0, L1, L2, L3

TABLE 3-7 *location* Descriptions for an I/O Assembly

Board or Device	Component Location
I/O assemblies (*slots*)	IB6, IB7, IB8, IB9
Ports on the I/O assembly	P0, P1
Buses on the I/O assembly	B0, B1
I/O cards in the I/O assembly	C0, C1, C2, C3, C4, C5, C6, C7—the number of cards varies with the I/O assembly

Note – If you are disabling the port locations of an I/O assembly, leave at least one I/O controller 0 enabled in a domain, so that the domain can communicate with the system controller.

See Also

`showcomponent`, the *Sun Fire Midrange Systems Platform Administration Manual*

Examples

CODE EXAMPLE 3-45 enables a component located in slot `sb4`. This means that the CPU/Memory board in slot `sb4` is considered for configuration into the system, subject to the component health status. The status change occurs at the next domain reboot, board power cycle, or POST execution.

CODE EXAMPLE 3-45 `setls` Command Example Enabling the Location of a CPU/Memory Board in Slot `sb4`

```
schostname:A> setls -s enable -l sb4
```

CODE EXAMPLE 3-46 enables an I/O assembly located in port 1 of slot `ib6`. This means that I/O assembly 6, port 1 is considered for configuration into the system, subject to the component health status. The status change occurs at the next domain reboot, board power cycle, or POST execution.

CODE EXAMPLE 3-46 `setls` Command Example Enabling the Location of an I/O Assembly in Slot `ib6`

```
schostname:A> setls -s enable -l ib6/p1
```

CODE EXAMPLE 3-47 disables a component located in slot `sb0` and CPU port `3`.This means that the CPU/Memory board in slot `sb0` is deconfigured from the system at the next domain reboot, board power cycle, or POST execution. Also, any memory banks on this CPU port are unreachable and are implicitly disabled.

CODE EXAMPLE 3-47 `setls` Command Example Disabling the Location of a CPU/Memory Board in Slot `sb0` and CPU port 3

```
schostname:A> setls -s disable -l sb0/p3
```

CODE EXAMPLE 3-48 disables a component located in slot `sb4` and deconfigures the CPU/Memory board in slot `sb4` at the next domain reboot, board power cycle, or POST execution.

CODE EXAMPLE 3-48 `setls` Command Example Disabling the Location of a CPU/Memory Board in Slot `sb4`

```
schostname:A> setls -s disable -l sb4
```

`setupdomain`

Configures the domain.

Scope

domain shell

Syntax

```
setupdomain [-p part]...
```

```
setupdomain -h
```

Options/Parameters

`-h` displays help for this command.

`-p` *part* configures the specified information and has the following options:

- `bootparams` configures the boot parameter information.
- `loghost` configures the log hosts.
- `snmp` configures SNMP information.

TABLE 3-8 lists the parameter values for `setupdomain`. Note that for `diag-level` definition and function of `default` and `max` are the same. Because the tests for `mem1` and `mem2` take a long time to run, use the `default` or `max` values. However, if you suspect memory problems, then use the `mem1` or `mem2` values.

TABLE 3-8 Parameter Values for the `setupdomain` Command

Parameter	Value	Description
`diag-level`	`init`	Only system board initialization code is run. No testing is done. This is a very fast pass through POST.
	`quick`	All system board components are tested using few tests with few test patterns.
	`min`	Core functionalities of all system board components are tested. This testing performs a sanity test of the devices under test.

TABLE 3-8 Parameter Values for the `setupdomain` Command *(Continued)*

Parameter	Value	Description
	`default` (default value)	All system board components are tested with all tests and test patterns, except for memory and Ecache modules. For memory and Ecache modules, all locations are tested with multiple patterns. More extensive, time-consuming algorithms are not run at this level. This is the same as `max`.
	`max`	All system board components are tested with all tests and test patterns. This is the same as `default`.
	`mem1`	Runs all tests at the default level plus more exhaustive DRAM and SRAM test algorithms. These tests take a long time to run.
	`mem2`	This is the same as `mem1` with the addition of a DRAM test that does explicit compare operations of the DRAM data. These tests take a long time to run.
`post-tolerate-ce`	`true`	If set to true it allows the Solaris Operating System to boot with memory exhibiting correctable ECC errors. The Solaris 10 operating system incorporates features that automatically isolate faulty parts of such memory modules, thus avoiding the need to completely disable these modules and increasing system availability.
	`false`	If set to false, memory modules exhibiting correctable ECC errors are disabled by POST and not allowed to participate in the Solaris domain.
`max-panic-diag-limit`	(The same list of values as `diag-level`. The default value is `mem2`)	Defines the maximum level of POST that runs automatically during repeated domain panics. POST level is escalated upon repeated panics until it runs the level specified in `max-panic-diag-limit`. If the domain panics again, it is placed in standby..
`verbosity-level`	`off`	No status messages are displayed. POST output is always displayed. However, with the value set to `off`, there is very little POST output displayed and there may be a long time between output displays. This depends on your hardware configuration and the setting of `diag-level`.
	`min` (default value)	Test names status messages, and error messages are displayed.
	`max`	Subtest trace messages are displayed.
`error-level`	`off`	No error messages are displayed.
	`min`	The failing test name is displayed.

TABLE 3-8 Parameter Values for the `setupdomain` Command *(Continued)*

Parameter	Value	Description
	`max` (default value)	All relevant error status is displayed.
`interleave-scope`	`within-cpu`	The memory banks on a CPU will be interleaved with each other. Each CPU may have up to four logical memory banks.
	`within-board` (default value)	The memory banks on a system board will be interleaved with each other.
	`across-boards`	The memory will be interleaved on all memory banks across all of the boards in the domain. With this value, you cannot remove a CPU/Memory board from the system using DR.
`interleave-mode`	`optimal` (default value)	The memory is mixed-size interleaving in order to gain optimal performance.
	`fixed`	The memory is fixed-size interleaving.
	`off`	There is no memory interleaving.
`reboot-on-error`	`true` (default value)	The domain is rebooted when the system controller detects a hardware error. If `OBP.auto-boot?` is set to true, the Solaris operating environment is booted. Note: The automatic reboot of a given domain can occur up to a maximum of three times. However, after the third time, the domain is paused when the system controller detects another hardware error on the domain. It is recommended that you contact your service provider for assistance rather than manually restart the domain.
	`false`	The domain is paused when the system controller detects a hardware error. To recover, you must manually reset the domain (turn the domain off, then on by using the `setkeyswitch` command).
	`display`	Error messages are reported and logged.
`hang-policy`		Controls the automatic recovery of hung domains.
	`reset`	Automatically reset a hung domain through an Externally Initiated Reset (XIR). Also reports a hung domain through the domain console or domain loghost messages.
	`notify`	Does not reset a hung domain but reports, through the domain console or domain loghost messages, that the domain is hung.

TABLE 3-8 Parameter Values for the `setupdomain` Command *(Continued)*

Parameter	Value	Description
`OBP.use-nvramrc?`		This parameter is the same as the OpenBoot PROM `nvramrc?` parameter. This parameter uses aliases that are stored in `nvramrc`.
	`true`	The OpenBoot PROM executes the script stored in `nvramrc` if this parameter is set to true.
	`false` (default value)	The OpenBoot PROM does not evaluate the script stored in `nvramrc` if this parameter is set to false.
`OBP.auto-boot?`		Controls booting of the Solaris operating environment.
	`true` (default value)	If this value is true, the domain boots automatically after POST has run.
	`false`	If this parameter value is set to false, you will obtain the OpenBoot PROM `ok` prompt after POST runs, from which you must type a `boot` command to boot the Solaris operating environment.
`OBP.error-reset-recovery`		Controls the behavior of the domain after an externally initiated reset (XIR) as well as a red mode trap.
	`sync` (default value)	The OpenBoot PROM invokes `sync`. A core file is generated. If the invocation returns, the OpenBoot PROM performs a reboot.
	`none`	The OpenBoot PROM prints a message describing the reset trap that triggered the error reset and passes control to the OpenBoot PROM `ok` prompt. The message describing the reset trap type is platform specific.
	`boot`	The OpenBoot PROM firmware reboots the domain. A core file is not generated. Rebooting a domain occurs using the OpenBoot PROM settings for `diag-device` or `boot-device`, depending on the value of the OpenBoot PROM configuration variable `diag-switch?` If `diag-switch?` is set to true, the device names in `diag-device` will be the default for boot. If `diag-switch?` is set to false, the device names in `boot-device` will be the default for boot.
`Loghost`		The name or the IP address of the loghost for this domain (SNMP).

TABLE 3-8 Parameter Values for the `setupdomain` Command *(Continued)*

Parameter	Value	Description
`Log Facility`		The configuration of the log facility on the SC should be consistent with that of the SYSLOG server to which the messages are sent. The default is `local0`.
`Domain Description`		Enter a brief description for this domain. For example, its function.
`Domain Contact`		Enter the name of the primary domain administrator.
`Trap Hosts`		Enter the name or the IP address of the SNMP trap host for this domain. The SNMP agent sends traps to the trap host on a SNMP default port number (162). An optional and different port number can be used other than the default port number. The format of the trap host is host[:port].
`Public Community String`		Community string for SNMP readers. The default value is A-public, B-public, and so on. **Note**—For SNMP clients such as the Sun Management Center software to access the system controller using SNMP, their community strings should be set to the same value as the value entered here.
`Private Community String`		Community string for readers and writers. The default value is A-private, B-private, and so on. **Note**—For SNMP clients such as the Sun Management Center software to access the system controller using SNMP, their community strings should be set to the same value as the value entered here.

Description

Configures domain specific values. Configures the domain. You can configure each domain differently and independently This command is an interactive command. You are prompted for parameters to configure.

See Also

`password`, `setdate`, `showdate`, `showdomain`, and the section "To Configure Domain-Specific Parameters" in the "System Power On and Setup" chapter of the *Sun Fire Midrange Systems Platform Administration Manual*.

Example

CODE EXAMPLE 3-49 Variables for the `setupdomain` Command

```
schostname:A> setupdomain

Domain Boot Parameters
----------------
diag-level [default]:
post-tolerate-ce [false]:
verbosity-level [min]:
error-level [max]:
interleave-scope [within-board]:
interleave-mode [optimal]:
reboot-on-error [true]:
hang-policy [reset]:
max-panic-diag-limit[mem2]:
OBP.use-nvramrc? [<OBP default>]:
OBP.auto-boot? [<OBP default>]:
OBP.error-reset-recovery [<OBP default>]:

Log hosts
---------
Loghost [ ]:The hostname or the IP address of the loghost for this domain.
Log Facility [local0]: Defined to be consistent with the SYSLOG server receiving the messages.

SNMP
-----
Domain Description[]: A brief description for this domain (its function).
Domain Contact[ ]: The name of the primary domain administrator.
Trap Hosts [ ]: The name or IP address of the SNMP trap host for this domain.
Public Community String [ ]:The community string for SNMP readers.
Private Community String [ ]: The community string for readers and writers.

schostname:A>
```

Note – The default value displayed by the software for the `OBP.*` parameters is `OBP default`.

If you enter an invalid value, the system controller displays "`invalid entry`" and prompts you to enter a proper value (TABLE 3-8). If a value is not specified, it retains its current value. When an invalid entry is entered, it also lists the valid responses. Typing a dash (-), clears the entry.

Current values are displayed in [ ]. TABLE 3-8 lists the `setupdomain` parameter values.

`setupplatform`

Configures the platform specific variables.

Scope

platform shell

Syntax

```
setupplatform [-p part]...
```

```
setupplatform -h
```

Options/Parameters

`-h` displays help for this command.

`-p` *part* configures the specified part and has the following options:

- `acls` is ACLs (access control list)

  - clears the boards from the ACL

  + adds all boards to the ACL

  `-a` *board* adds a board

  `-d` *board* deletes a board

- `cod` configures the Capacity on Demand parameters and has the following options:

  headroom-number configures only the COD instant access CPU quantity.

  `-d` *domainID proc-RTUs* reserves a specific number of COD RTU licenses for a specified domain (a, b, c, or d).

- `hostid` swaps the HostID/MAC address of one domain with another.

  [`-m auto`] restores the HostID/MAC addresses that were swapped to their original domain assignments.

Note – The `HostID/MAC Address Swap` parameter is displayed only when you specify the `hostid` option. For details, see the “Description” section of this command.

- `loghost` configures the loghost and facility with the default of `local0`. The loghost is the IP address or host name of the `syslog` loghost. The host name you enter must be a complete host name (not a partial one).
- `network` includes the network settings. Among other things, this setting controls remote access to the system controller so that either SSH or Telnet services are enabled or disabled. With the exception of specifying the timeout period, all network settings require a reboot before they can take effect. You can also set a timeout period for idle remote or serial port connections to the system controller.
- `partition` sets the partition mode.
- `sc` configures system controller behavior.
- `snmp` is Simple Network Management Protocol (SNMP) configuration.
- `sntp` is Simple Network Time Protocol (SNTP)
- `usiv+` is UltraSPARC IV+ support. This support is available in one domain per partition (domains `a`, `c`) only.

Description

Configures parameters for the platform. This command is an interactive command. You are prompted for parameters to configure. For parameters to configure, see TABLE 3-9.

TABLE 3-9 `setupplatform` Parameter Values

Parameter	**Description**
`Is the System Controller on a Network?`	If the answer is no, leave all network parameters as unconfigured. You will be prompted to enter the hostname. If the answer if yes, you are prompted to configure the network settings.
`Use DHCP or static network settings?`	• `DHCP` means the system controller network configuration is retrieved from a DHCP server. • `Static` means the network settings will be manually configured. If you select the static setting, you will be asked for the following network parameters.
`Hostname` (Network parameter)	The human readable network identity for this system controller.
`IP address` (Network parameter)	The network identity used by computers.
`Netmask` (Network parameter)	For this value, specify how much of the address should be reserved for subdividing networks into subnetworks. Must be specified in dot-notation address. For example `255.255.255.0`.
`Gateway` (Network parameter)	IP address of the gateway.
`DNS Domain` (Network parameter)	Domain name. For example, *xxx.xxx*`.com`.

TABLE 3-9 `setupplatform` Parameter Values *(Continued)*

Parameter	**Description**
`Primary DNS Server` (Network parameter)	IP address of your primary DNS server.
`Secondary DNS Server` (Network parameter)	IP address of your secondary DNS server.
`Loghost`	The name or the IP address of the loghost for the platform.
`Connection type`	Type of remote connection: `ssh`, `telnet` or `none` (serial connection). **Note:** For *secure* remote connections, select `ssh.` Refer to related security recommendations in the "Security Guidelines" chapter of the *Sun Fire Midrange Systems Platform Administration Manual*.
`Idle connection timeout`	Timeout interval, measured in minutes. Zero (0) means no timeout. **Note:** The domain boot process, which does not require user input, is considered to be idle. If your timeout period is shorter than the domain boot time, connections can be dropped during domain bootup. If the connection is dropped, the boot process continues to completion, but the console will not be available until the boot process reaches the OBP state.
`Enable Sun Fire Link?`	This parameter is displayed only when your system contains a Sun Fire Link board. The default is Sun Fire Link disabled. If you answer yes, the Sun Fire Link is enabled.
`Log Facility`	The configuration of the log facility, on the SC, should be consistent with that of the SYSLOG server to which the messages are sent. The default is `local0.` **Note:** Refer to related security recommendations in the "Security Guidelines" chapter of the *Sun Fire Midrange Systems Platform Administration Manual*.
`SNTP server`	The name of the Simple Network Time Protocol (SNTP) server that automatically tracks and corrects the drift (difference) between the local system clocks managed by each system controller. **Note:** SNTP is recommended if you enable SC failover. Refer to related security recommendations in the "Security Guidelines" chapter of the *Sun Fire Midrange Systems Platform Administration Manual*.
`Platform Description`	Default value is the platform model name.
`Platform Contact`	The name of the person who is responsible for this system. This name will be used by SNMP. Note that if SNMP is not going to be enabled, this field is informational only.

TABLE 3-9 `setupplatform` Parameter Values *(Continued)*

Parameter	Description
`Platform Location`	Location of the system (such as a room and/or the location inside of the room). This information will be used by SNMP. Note that if SNMP is not going to be enabled, this field is informational only.
`Enable SNMP Agent?`	The default is SNMP disabled. If you answer yes, the Simple Network Management Protocol (SNMP) agent enables SNMP. **Note:** The default setting (`disabled`) is recommended. Enable SNMP only if you use the Sun Management Center software. Refer to related security recommendations in the “Security Guidelines” chapter of the *Sun Fire Midrange Systems Platform Administration Manual*.
`Trap Hosts`	IP address or hostname of the SNMP trap hosts. The SNMP agent sends traps to the trap host on an SNMP default port number (162). An optional and different port number can be used other than the default port number. The format of the trap host is host[:port].
`Public Community String`	Community string for SNMP readers. The default is `P-public`. **Note:** For SNMP clients such as the Sun Management Center software to access the system controller using SNMP, their community strings should be set to the same value as the value entered here.
`Private Community String`	Community string for readers and writers. The default is `P-private`. **Note**: For SNMP clients such as the Sun Management Center software to access the system controller using SNMP, their community strings should be set to the same value as the value entered here.
`usiv+`	UltraSPARC IV+ support. This support is available in one domain per partition (domains `a`, `c`) only.

TABLE 3-9 `setupplatform` Parameter Values *(Continued)*

Parameter	Description
`ACL for Domain A`	Access control list (ACL). If the board name is listed in the ACL for the specific domain, then you have permission to perform `addboard` or `deleteboard` requests on that board. Default value is to have all slots assigned to all domains. **Note**: Refer to related security recommendations in the "Security Guidelines" chapter of the *Sun Fire Midrange Systems Platform Administration Manual*. • Typing the prefix`-d` deletes the slot from ACLs of this domain. • Typing a prefix of `-a` (add) adds a slot. • Typing a dash (`-`) clears the entire list. • Typing a + (plus sign) adds all. • Pressing the Return key does not change the value displayed inside [ ].
`ACL for Domain B`	See the description for ACL for Domain A.
`ACL for Domain C`	See the description for ACL for Domain A.
`ACL for Domain D`	See the description for ACL for Domain A.
`PROC RTUs installed`	Displays the number of COD RTU licenses currently installed
`PROC Headroom Quantity (0 to disable, 4 MAX) [`*x*`]`	The number of COD instant access CPUs to be enabled. • The maximum number of instant access CPUs that can be enabled (4) is displayed inside the parenthesis. • Specify 0 to disable the instant access CPU quantity only if there are no instant access CPUs currently in use. • The current number of instant access CPUs enabled is displayed inside the brackets.
`PROC RTUs reserved for domain A (`*x*` MAX) [`*y*`]:`	The number of COD right-to-use (RTU) licenses to be reserved for the domain. • The maximum number of COD RTU licenses that can be reserved for the domain is displayed inside the parenthesis. • The current number of COD RTU licenses allocated to the domain is displayed inside the brackets.
`PROC RTUs reserved for domain B (`*x*` MAX) [`*y*`]:`	See the description for PROC RTUs reserved for domain A.
`PROC RTUs reserved for domain C (`*x*` MAX) [`*y*`]:`	See the description for PROC RTUs reserved for domain A.
`PROC RTUs reserved for domain D (`*x*` MAX) [`*y*`]:`	See the description for PROC RTUs reserved for domain A.
`SC POST diag level`	The system controller boot parameters (diag level) can be set to: `off`, `min`, or `max`.

TABLE 3-9 `setupplatform` Parameter Values *(Continued)*

Parameter	Description
`off`	Only system board initialization code is run. No testing is done. POST is not run.
`min`	All System Controller board components are tested using few locations with few test patterns.
`max`	All System Controller board components are tested with all tests and test patterns.
`Enable SC failover?`	If the answer is `yes` and you have a redundant SC configuration, SC failover is enabled (activated). If the answer is `no`, SC failover is disabled until you re-enable failover.
`Logical Hostname or IP Address`	Name of the logical host or IP address that always identifies the working main SC. When an SC failover occurs, this logical host name or IP address is associated (floats) with the new main SC. The floating IP address must be different from the IP address used to identify the system controller.
`Configure chassis for single or dual partition mode?`	The two options are single or dual. The default is single. For a description of single and dual-partition mode, refer to the “Overview” chapter in the *Sun Fire Midrange Systems Platform Administration Manual*.

`HostID/MAC Address Swap` *Parameter*

The `HostID/MAC Address Swap` parameter is displayed only when you specify the `-p hostid` option with the `setupplatform` command. This parameter enables you to swap the HostID and MAC address of one domain with another. For example, if you are running host-licensed software associated with a particular domain HostID and MAC address, but you need to run the host-licensed software on another domain, you can swap the domain HostID and MAC address with that of an available domain. You can then run the host-licensed software on the available domain without encountering license restrictions tied to the original domain HostID/MAC address. For details on using the HostID/MAC address swap, refer to the *Sun Fire Midrange Systems Platform Administration Manual.*

TABLE 3-10 describes the prompts displayed when you specify the `setupplatform -p hostid` command. For an example of the output displayed, see CODE EXAMPLE 3-51:

TABLE 3-10 `setupplatform HostID/MAC Address Swap` Parameter Prompts

Parameter	Description
`Domain to swap HostID/MAC address [A,B,C,D]:`	The domain whose HostID/MAC address will be swapped with another domain. The domain selected must not be an active domain. **Note –** This prompt and the next are used to specify the domains involved in the HostID/MAC address swap. One of the domains selected must be the domain currently associated with the host-licensed software. The other domain specified must be the domain on which the host-licensed software will run.
`Domain to swap HostID/MAC address with [A,B,C,D]:`	The domain whose HostID/MAC address will be swapped with the previously specified domain. The domain selected must not be an active domain.
`Commit swap? [no]:`	If the answer is yes, the HostID/MAC addresses of the specified domains are swapped.
`Swap HostIDs/MAC addresses of another pair of Domains? [no]:`	If the answer is yes, you will be prompted to specify the pair of domains whose HostID/MAC addresses will be swapped.
`Restore automatic HostID/MAC address assignment? [no]: n`	Displayed only when you specify the `-m auto` option with the `-p hostid` option. If the answer is yes, the HostID/MAC addresses that were swapped between domains are restored to the original domains.

See Also

`password`, `setdate`, `setupdomain`, `showdate`, `showdomain`, `showplatform`, and the *Sun Fire Midrange Systems Platform Administration Manual*, which provides instructions on setting up the platform using `setupplatform`.

Examples

CODE EXAMPLE 3-50 `setupplatform` Output

```
schostname:SC> setupplatform

Network Configuration
---------------------
Is the system controller on a network? [yes]:
```

CODE EXAMPLE 3-50 `setupplatform` Output *(Continued)*

```
Use DHCP or static network settings? [DHCP]:
Hostname []:
IP Address []:
Netmask []:
Gateway []:
DNS Domain []:
Primary DNS Server []:
Secondary DNS Server []:

To enable remote access to the system controller, select "ssh" or "telnet."

Connection type: ssh

Rebooting the SC is required for changes in the above network settings
to take effect.

Idle connection timeout (in minutes; 0 means no timeout) [0]:

Enable Sun Fire Link? [no]:

Loghosts
--------
Loghost [ ]:
Log Facility [local0]:

SNTP
----
SNTP server [ntp1]:

SNMP
----
Platform Description [Sun Fire 6800]: System type
Platform Contact [ ]: Platform administrator's name
Platform Location [ ]: Geographic system location

Do not enable SNMP Agent unless you use Sun Management Center software.

Enable SNMP Agent? [yes]:
Trap Hosts []: SNMP trap host IP address or name
Public Community String [P-public]: Community string for SNMP readers
Private Community String [P-private]: Community string for readers and writers

ACLs
----
ACL for domain A [SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9 ]:
ACL for domain B [SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9 ]:
ACL for domain C [SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9 ]:
```

CODE EXAMPLE 3-50 `setupplatform` Output *(Continued)*

```
ACL for domain D [SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9 ]:



COD
---
PROC RTUs installed: 8
PROC Headroom Quantity (0 to disable, 4 MAX) [0]:Number of instant access CPUs to be enabled
PROC RTUs reserved for domain A (6 MAX) [0]: Number of domain COD RTU licenses reserved
PROC RTUs reserved for domain B (6 MAX) [0]: Number of domain COD RTU licenses reserved
PROC RTUs reserved for domain C (4 MAX) [0]: Number of domain COD RTU licenses reserved
PROC RTUs reserved for domain D (4 MAX) [0]: Number of domain COD RTU licenses reserved

SC
--
SC POST diag Level [min]:
Enable SC failover? [yes]:
Logical Hostname or IP address [sp1-sc]:

Partition Mode
--------------
Configure chassis for single or dual partition mode? [single]:

schostname:SC>
```

CODE EXAMPLE 3-51 `setupplatform -p hostid` Output

```
schostname:SC> setupplatform -p hostid

HostID/MAC Address Swap
-----------------------
Domain to swap HostID/MAC address [A,B,C,D]: b
Domain to swap HostID/MAC address with [A,B,C,D]: d
Commit swap? [no]: y

Swap HostIDs/MAC addresses of another pair of Domains? [no]: n

schostname:SC>
```

CODE EXAMPLE 3-52 `setupplatform -p hostid -m auto` Output

```
schostname:SC> setupplatform -p hostid -m auto

HostID/MAC Address Swap
-----------------------
Restore automatic HostID/MAC address assignment? [no]: y

schostname:SC>
```

`showboards`

Displays the assignment information and status for all components in the system.

Scope

platform shell, domain shell

Syntax for the Platform Shell

`showboards` [`-ev`] [`-y`|`-n`] [`-d` *domainID*] [`-p` *part*]

`showboards -h`

Syntax for the Domain Shell

`showboards` [`-aev`] [`-y`|`-n`] [`-p` *part*]

`showboards -h`

Options/Parameters

`-a` includes available boards (domain only).

`-e` includes empty slots in the output.

`-v` verbose mode.

`-d` *domainID* specifies a domain to display (platform only).

`-p` *part* shows only a specific part.

- `board` shows the board status.
- `clock` shows the system clock status.
- `cpu` shows CPU type, speed, and Ecache size.
- `io` shows I/O information.
- `memory` shows memory information for each board.
- `power` shows grid information.
- `version` shows version information.

`-y` answers yes to all questions.

`-n` answers no to all questions.

`-h` displays help for this command.

Description

Displays the assignment information and status for all of the components in the system—CPU/Memory boards, I/O assemblies, fan trays, and so on. For the domain shell, the display shows components assigned to the domain. It can also include boards that are included in the ACL, and not assigned to any other domain.

See Also

`addboard`, `deleteboard`

Examples—Platform Shell

CODE EXAMPLE 3-53 `showboards` Example Output for the Platform Shell

```
schostname:SC> showboards

Slot      Pwr Component Type                State      Status     Domain
----      --- --------------                -----      ------     ------
SSC0      On  System Controller V2          Main       Passed     -
SSC1      On  Present                       Spare      -          -
ID0       On  Sun Fire Replacement ID Board -          OK         -
PS0       On  A184 Power Supply             -          OK         -
PS1       On  A184 Power Supply             -          OK         -
PS2       On  A184 Power Supply             -          OK         -
PS3       On  A184 Power Supply             -          OK         -
PS4       On  A184 Power Supply             -          OK         -
PS5       On  A184 Power Supply             -          OK         -
FT0       On  Fan Tray                      Low Speed  OK         -
FT1       On  Fan Tray                      Low Speed  OK         -
FT2       On  Fan Tray                      Low Speed  OK         -
FT3       On  Fan Tray                      Low Speed  OK         -
RP0       On  Repeater Board                -          OK         -
RP1       On  Repeater Board                -          OK         -
RP2       On  Repeater Board                -          OK         -
RP3       On  Repeater Board                -          OK         -
/N0/SB0   On  CPU Board V3                  Active     Passed     A
/N0/SB1   On  CPU Board V3                  Active     Passed     A
/N0/SB2   On  CPU Board V3                  Active     Passed     A
/N0/SB3   On  CPU Board V3                  Active     Passed     A
/N0/SB4   On  CPU Board V3                  Active     Passed     A
/N0/SB5   On  CPU Board V3                  Active     Passed     A
/N0/IB6   On  PCI I/O Board                 Active     Passed     A
```

CODE EXAMPLE 3-53 `showboards` Example Output for the Platform Shell *(Continued)*

```
/N0/IB7  On  PCI I/O Board                   Active     Passed     A
/N0/IB8  On  Fire Link/CPCI I/O Board        Active     Degraded   A
/N0/IB9  On  Fire Link/CPCI I/O Board        Active     Degraded   A
```

The `showboards` command with the `-v` option displays the verbose option.

The `showboards` command with the `-p version` option displays firmware version and compatibility information.

CODE EXAMPLE 3-54 `showboards` Command with the `-p version` Option

```
schostname:SC> showboards -p version

Component   Compatible Version
---------   ---------- -------
SSC1        Reference  5.17.0
/N0/IB6     Yes        5.17.0
/N0/IB7     Yes        5.17.0
/N0/IB8     Yes        5.17.0
/N0/IB9     Yes        5.17.0
/N0/SB0     Yes        5.17.0
/N0/SB1     Yes        5.17.0
/N0/SB2     Yes        5.17.0
/N0/SB3     Yes        5.17.0
/N0/SB4     Yes        5.17.0
/N0/SB5     Yes        5.17.0

schostname:SC>
```

TABLE 3-11 Output Header Definitions for the `showboards` and the `showboards -v` Command

Header	Description
Slot	Slot designator. The N*x* in the slot descriptor is the node number.
Pwr	Indicates if the power status of the device is off or on.
Component type	Component description, such as System Controller, CPU Board, COD CPU Board, Fan Tray, and so on. Note the following component descriptions: • `System Controller` – SC without enhanced memory. • `System Controller V2` – SC with enhanced memory. • `CPU Board` or `COD CPU Board` – CPU board revision 1. • `CPU Board V2` or `COD CPU Board V2` – CPU board revision 2. • `CPU Board V3` or `COD CPU Board V3` – CPU board revision 3.
State	Describes board state. Possible values are: Active, Assigned, Available, and – (dash). The dash indicates that the board state does not apply to this slot. For system controllers, possible values are: Main or Spare. For more information on board states, see "Board States for CPU/Memory Boards and I/O Assemblies" on page 6.
Status	Current board status. For more information on board status, see "Board Test Status" on page 6.
Domain	Indicates which domain the board belongs to. For more information on domains, refer to the "Overview" chapter of the *Sun Fire Midrange Systems Platform Administration Manual*.
	Memory Information
Component	Component, such as a board, CPU, or memory DIMM.
J-No	Component label on the board.
Size	Lists the size of the DIMM.
Reason	Explains why the value is not reported.
	Firmware Versions
Component	Board that contains a flash PROM.
Segment	Software module: ScApp, iPOST, POST, or OBP. The value Ver indicates that all software modules have been joined to form a particular image.

TABLE 3-11 Output Header Definitions for the `showboards` and the `showboards -v` Command *(Continued)*

Header	Description
Compatible	Indicates whether the given component is compatible with the `scapp` version listed in the Version column. Possible values are Yes, No, and – (dash). The dash indicates that firmware compatibility does not apply to the given component. For system controllers, the value Reference indicates that the firmware version listed in the Version column is the baseline used to determine firmware compatibility.
In	The number that identifies the software modules for the firmware. This number is used to determine the compatibility with the firmware.
Date	The month, day, and year that the flash PROM image was created by Sun Microsystems.
Time	The time of day (hour:minutes) that the flash PROM image was created by Sun Microsystems.
Build	A number that identifies the internal build. Used by Sun Microsystems only.
Version	The firmware version compatible with the component, except for System Controllers and boards that are not powered on (No board power).
	Clock Status
Component	Board.
SSC*x* Signal	Lists the status of the SSC0 or SSC1 clock on the board.
Signal Used	Identifies which clock is used by the board.
Failover	Indicates if the board will respond to clock failover.
	I/O Status
Slot	I/O slot.
Populated	Indicates if a card is installed in the slot.
Slot Description	Describes the slot. The card type cannot be determined by the system controller. The card is determined by the Solaris operating environment.
	CPU Information
Component	CPU.
Cpu Mask	Lists the CPU version.
Description	Indicates the CPU type, speed, and Ecache size.

TABLE 3-11 Output Header Definitions for the `showboards` and the `showboards -v` Command *(Continued)*

Header	Description
	Power Grids
Component	Board.
Pwr	Indicates if the board is on or off.
Grid	Identifies which power grid the board is in.

Example—Domain Shell

CODE EXAMPLE 3-55 `showboards -a` Command for the Domain Shell

```
schostname:A> showboards -a

Slot     Pwr Component Type                    State      Status     Domain
----     --- --------------                    -----      ------     ------
/N0/SB0  On  CPU Board V3                      Active     Passed     A
/N0/SB1  On  CPU Board V3                      Active     Passed     A
/N0/SB2  On  CPU Board V3                      Active     Passed     A
/N0/SB3  On  CPU Board V3                      Active     Passed     A
/N0/SB4  On  CPU Board V3                      Active     Passed     A
/N0/SB5  On  CPU Board V3                      Active     Passed     A
/N0/SB0  On  CPU Board V3                      Active     Passed     A
/N0/SB0  On  CPU Board V3                      Active     Passed     A
/N0/SB0  On  CPU Board V3                      Active     Passed     A
/N0/SB0  On  CPU Board V3                      Active     Passed     A
```

`showcodlicense`

Displays the current Capacity on Demand (COD) right-to-use (RTU) licenses stored in the COD license database.

Scope

platform shell

Syntax for the Platform Shell

`showcodlicense` [`-r`] [`-v`]

`showcodlicense -h`

Options/Parameters

`-h` displays help for this command.

`-r` displays the license information in the raw license key format, as stored in the COD license database.

`-v` verbose mode. Displays both the formatted license information and raw license key format.

Description

TABLE 3-12 describes the default COD information displayed by the `showcodlicense` command.

TABLE 3-12 COD License Information

Item	Description
Description	Type of resource (processor)
Ver	Version number of the license
Expiration	None. Not supported (no expiration date)
Count	Number of RTU licenses granted for the given resource
Status	One of the following states: • GOOD – Indicates the resource license is valid • EXPIRED – Indicates the resource license is no longer valid

See Also

`addcodlicense`, `deletecodlicense`, `showcodusage`

Examples

To show formatted COD RTU license key data, type:

CODE EXAMPLE 3-56 `showcodlicense` Example Output – Formatted COD RTU License Data

```
schostname:SC> showcodlicense
Description  Ver   Expiration  Count  Status
-----------  ---   ----------  -----  ------
PROC          01         NONE      8  GOOD
```

To show COD RTU license keys, type:

CODE EXAMPLE 3-57 `showcodlicense -r` Example Output – COD RTU License Keys

```
schostname:SC> showcodlicense -r
01:80d8a9ed:45135285:0201000000:8:00000000:J0WKZ8vpZ5k0gSJ7e8ZasA
```

To show COD RTU license key data in both formatted and raw format, type:

CODE EXAMPLE 3-58 `showcodlicense -v` Example Output – Formatted and Raw COD RTU License Data

```
schostname:SC> showcodlicense -v
Description  Ver   Expiration  Count  Status
-----------  ---   ----------  -----  ------
PROC          01         NONE      8  GOOD
01:80d8a9ed:45135285:0201000000:8:00000000:J0WKZ8vpZ5k0gSJ7e8ZasA
```

`showcodusage`

Displays the current usage statistics for Capacity on Demand (COD) resources.

Scope

platform shell, domain shell

Syntax for the Platform Shell

```
showcodusage [-v] [-p resource|domains|all] ...
```

```
showcodusage -h
```

Options/Parameters

`-h` displays help for this command.

`-p resource` displays COD usage information according to resource type.

`-p domains` displays COD usage information according to domain type.

`-p all` displays COD usage information both by resource type and by domain.

`-v` is verbose mode.

Description

This command shows current information on the COD RTU licenses in use. By default, the command displays a summary of COD RTU licenses used and installed, along with the current state of each resource.

See Also

`showcodlicense`

Examples

To show information by resource, type:

CODE EXAMPLE 3-59 `showcodusage` Example Output by Resource

```
schostname:SC> showcodusage -p resource
Resource      In Use  Installed  Licensed  Status
--------      ------  ---------  --------  ------
PROC               0          4         8  OK: 8 available Headroom: 2
```

TABLE 3-13 describes the resource information displayed.

TABLE 3-13 `showcodusage` Resource Information

Item	Description
Resource	The COD resource (processor)
In Use	The number of COD CPUs currently used in the system
Installed	The number of COD CPUs installed in the system
Licensed	The number of COD RTU licenses installed
Status	One of the following COD states: • OK – Indicates there are sufficient licenses for the COD CPUs in use and specifies the number of remaining COD resources available and the number of any instant access CPUs (headroom) available • HEADROOM – The number of instant access CPUs in use • VIOLATION – Indicates a license violation exists. Specifies the number of COD CPUs in use that exceeds the number of COD RTU licenses available. This situation can occur when you force the deletion of a COD license key from the COD license database, but the COD CPU associated with that license key is still in use.

To show information by domain, type:

CODE EXAMPLE 3-60 `showcodusage` Example Output by Domain

```
schostname:SC> showcodusage -p domains -v
Domain/Resource  In Use  Installed  Reserved  Status
---------------  ------  ---------  --------  ------
A - PROC              6          8         4
     SB0 - PROC       4          4
     /N0/SB0/P0                               Licensed
     /N0/SB0/P1                               Licensed
```

CODE EXAMPLE 3-60 `showcodusage` Example Output by Domain *(Continued)*

```
      /N0/SB0/P2                                        Licensed
      /N0/SB0/P3                                        Licensed
      SB1 - PROC        2           4
      /N0/SB1/P0                                        Licensed
      /N0/SB1/P1                                        Licensed
      /N0/SB1/P2                                        Unlicensed
      /N0/SB1/P3                                        Unlicensed
B - PROC                0           0           0
C - PROC                0           0           0
D - PROC                0           0           0
Unused - PROC           0           4           0
      SB2 - PROC        0           4
      SB2/P0                                            Unused
      SB2/P1                                            Unused
      SB2/P2                                            Unused
      SB2/P3                                            Unused
```

TABLE 3-14 describes the domain information displayed.

TABLE 3-14 `showcodusage` Domain Information

Item	Description
Domain/Resource	The COD resource (processor) for each domain. An unused processor is a COD CPU that has not yet been assigned to a domain.
In Use	The number of COD CPUs currently used in the domain
Installed	The number of COD CPUs installed in the domain
Reserved	The number of COD RTU licenses allocated to the domain
Status	One of the following CPU states: • Licensed – The COD CPU has an RTU license. • Unused – The COD CPU is not in use. • Unlicensed – A COD RTU license could not be obtained for the COD CPU, so the COD CPU cannot be used.

To show information by both resource type and domain, type:

CODE EXAMPLE 3-61 `showcodusage` Example Output by Both Resource and Domain

```
schostname:SC> showcodusage -v
Resource      In Use  Installed  Licensed  Status
--------      ------  ---------  --------  ------
PROC               0          4         8  OK: 8 available Headroom: 2
Domain/Resource  In Use  Installed  Reserved  Status
---------------  ------  ---------  --------  ------
A - PROC              0          0         4
B - PROC              0          0         4
C - PROC              0          0         0
D - PROC              0          0         0
Unused - PROC         0          4         0
     SB4 - PROC       0          4
     SB4/P0                                   Unused
     SB4/P1                                   Unused
     SB4/P2                                   Unused
     SB4/P3                                   Unused
```

`showcomponent`

Shows the state of a component.

Scope

platform shell, domain shell

Syntax for the Platform Shell

`showcomponent [-v] [`*component*`] ...`

`showcomponent [-v] [-d` *domainID*`] ...`

`showcomponent -h`

Syntax for the Domain Shell

`showcomponent [-v] [`*component*`] ...`

`showcomponent -h`

Options/Parameters

`-h` displays help for this command. Includes *component_name* syntax.

boardname for CPU/Memory boards is SB0 - SB5 and for I/O assemblies is IB6 - IB9.

-d *domainID* is a, b, c, or d. Shows components assigned to the domain (platform only).

Description

Without options, this command shows the state of all components on a board. With one or more options described in the “Syntax” section, the output shows the state of one or more components on a board.

Displays the components that have been disabled by the `setls` commands or components that have been marked as failed when testing (running the `testboard` command). This command also does the following:

- For CPU/Memory boards, provides information about CPUs, memory modules (DIMMs), and Ecache.

- Displays logical memory bank size.
- For Capacity on Demand (COD) CPU components that are not allocated a COD right-to-use (RTU) license, provides the status of such components as Cod-dis (COD disabled). For details, see the "Capacity on Demand" chapter in the *Sun Fire Midrange Systems Platform Administration Manual*.
- Displays the POST status of `chs` (component health status) for those components that require further analysis by your service provider. For details, see the "Automatic Diagnosis and Recovery" chapter in the *Sun Fire Midrange Systems Platform Administration Manual.*

See Also

`setls`, and the *Sun Fire Midrange Systems Platform Administration Manual*.

Examples

CODE EXAMPLE 3-62 shows sample output for the `showcomponent sb0` command. The abbreviations for the `Component` field are:

`N`*x*	Node name
`SB`*x*	CPU/Memory board, where x is 0 – 5
`P`*x*	Port, where x is 0 – 3
`B`*x*	Physical memory bank, where x is 0 – 1
`L`*x*	Logical memory bank, where x is 0 – 3. There are two DIMMs per logical memory bank

CODE EXAMPLE 3-62 `showcomponent` Sample Output

```
schostname:SC> showcomponent

/N0/SB4: is not powered on
Skipping /N0/SB4

Component           Status   Pending  POST   Description
---------           ------   -------  ----   -----------
/N0/SB0/P0          enabled  -        pass   UltraSPARC-IV, 1200MHz, 16M ECache
/N0/SB0/P1          enabled  -        pass   UltraSPARC-IV, 1200MHz, 16M ECache
/N0/SB0/P2          enabled  -        pass   UltraSPARC-IV, 1200MHz, 16M ECache
/N0/SB0/P3          enabled  -        pass   UltraSPARC-IV, 1200MHz, 16M ECache
/N0/SB0/P0/B0/L0    enabled  -        pass   1024M DRAM
/N0/SB0/P0/B0/L2    enabled  -        pass   1024M DRAM
```

CODE EXAMPLE 3-62 `showcomponent` Sample Output *(Continued)*

```
/N0/SB0/P0/B1/L1    enabled  -        untest empty
/N0/SB0/P0/B1/L3    enabled  -        untest empty
/N0/SB0/P1/B0/L0    enabled  -        pass   1024M DRAM
/N0/SB0/P1/B0/L2    enabled  -        pass   1024M DRAM
/N0/SB0/P1/B1/L1    enabled  -        untest empty
/N0/SB0/P1/B1/L3    enabled  -        untest empty
/N0/SB0/P2/B0/L0    enabled  -        pass   1024M DRAM
/N0/SB0/P2/B0/L2    enabled  -        pass   1024M DRAM
/N0/SB0/P2/B1/L1    enabled  -        untest empty
/N0/SB0/P2/B1/L3    enabled  -        untest empty
/N0/SB0/P3/B0/L0    enabled  -        pass   1024M DRAM
/N0/SB0/P3/B0/L2    enabled  -        pass   1024M DRAM
/N0/SB0/P3/B1/L1    enabled  -        untest empty
/N0/SB0/P3/B1/L3    enabled  -        untest empty
/N0/SB2/P0          enabled  -        pass   UltraSPARC-IV, 1050MHz, 16M ECache
/N0/SB2/P1          enabled  -        pass   UltraSPARC-IV, 1050MHz, 16M ECache
/N0/SB2/P2          enabled  -        pass   UltraSPARC-IV, 1050MHz, 16M ECache
/N0/SB2/P3          enabled  -        pass   UltraSPARC-IV, 1050MHz, 16M ECache
/N0/SB2/P0/B0/L0    enabled  -        pass   1024M DRAM
/N0/SB2/P0/B0/L2    enabled  -        pass   1024M DRAM
/N0/SB2/P0/B1/L1    enabled  -        untest empty
/N0/SB2/P0/B1/L3    enabled  -        untest empty
/N0/SB2/P0/B0/L0    enabled  -        pass   1024M DRAM
/N0/SB2/P0/B0/L2    enabled  -        pass   1024M DRAM
.
.
.
```

Definitions of the headings in CODE EXAMPLE 3-62 and CODE EXAMPLE 3-63 follow:

- `Component` is the name of the board and its components.
- `Status` is the status (enabled or disabled) of the board or component.
- `Pending` means that the component will change to the indicated status (enabled or disabled) at the next reboot.
- `POST` is the status of power-on self-test (POST) for the component. The states are: passed, failed, untested, or chs (component health status). For components that have a chs status, contact your service provider to obtain further analysis of those components.
- `Description` is a description of the board or component.

CODE EXAMPLE 3-63 shows sample output for the `showcomponent ib6` command. The abbreviations for the Component field are:

`N`*x*	Node name
`IB`*x*	I/O assembly, where *x* is 6 – 9
`P`*x*	Port, where *x* is 0 – 1
`B`*x*	Bus, where *x* is 0 – 1
`C`*x*	Card in the I/O assembly, where *x* is 0 – 7

CODE EXAMPLE 3-63 `showcomponent ib6` Sample Output

```
schostname:SC> showcomponent ib6

Component           Status   Pending  POST   Description
---------           ------   -------  ----   -----------
/N0/IB6/P0          enabled  -        pass   IO Controller 0
/N0/IB6/P1          enabled  -        pass   IO Controller 1
/N0/IB6/P0/B0       enabled  -        untest 66/33MHz. EPCI/PCI Bus
/N0/IB6/P0/B1       enabled  -        untest 33MHz. PCI Bus
/N0/IB6/P1/B0       enabled  -        untest 66/33MHz. EPCI/PCI Bus
/N0/IB6/P1/B1       enabled  -        untest 66/33MHz. EPCI/PCI Bus
/N0/IB6/P0/B1/C0    enabled  -        untest 33MHz. 3.3V Short PCI card
/N0/IB6/P0/B1/C1    enabled  -        untest 33MHz. 3.3V Short PCI card
/N0/IB6/P0/B0/C2    enabled  -        untest 66/33MHz. 3.3V Long/Short PCIX/EPCI/PCI card
/N0/IB6/P0/B0/C3    enabled  -        untest 66/33MHz. 3.3V Long/Short PCIX/EPCI/PCI card
/N0/IB6/P1/B1/C4    enabled  -        untest 66/33MHz. 3.3V Long/Short PCIX/EPCI/PCI card
/N0/IB6/P1/B1/C5    enabled  -        untest 66/33MHz. 3.3V Long/Short PCIX/EPCI/PCI card
/N0/IB6/P1/B0/C6    enabled  -        untest 66/33MHz. 3.3V Long/Short PCIX/EPCI/PCI card
/N0/IB6/P1/B0/C7    enabled  -        untest 66/33MHz. 3.3V Long/Short PCIX/EPCI/PCI card
```

`showdate`

Displays the date and time.

Scope

platform shell, domain shell

Syntax for the Platform Shell

```
showdate [-tv]
```

```
showdate -h
```

`showdate [-v] [-d` *domainID*`]`

Syntax for the Domain Shell

```
showdate [-tv]
```

```
showdate -h
```

Options/Parameters

`-h` displays help for this command.

`-t` lists available time zones.

`-v` is verbose mode.

-d *domainID* is a, b, c, or d. Shows the date for the domain from the platform.

Description

Displays the current date and time for the platform and for each domain. You can obtain the date for the platform shell and each domain shell from the platform shell. If you type the command from one of the domain shells, the date and time for the current domain is returned.

See Also

`setdate`

Example—Platform Shell

CODE EXAMPLE 3-64 `showdate` Command for the Platform Shell

```
schostname:SC> showdate
Fri Mar 07 18:00:56 PST 2003
```

`showdomain`

Displays the configuration and status of the domain.

Scope

domain shell

Syntax

`showdomain [-v]  [-p` *part*`]...`

`showdomain  -h`

Options/Parameters

`-v` is verbose mode.

`-p` *part* is a parameter for the information to display:

- `acls` shows the access control list
- `bootparams` shows boot parameter information
- `cod` shows the number of COD RTU licenses reserved for the domain
- `loghosts` shows the log hosts
- `mac` shows MAC addresses
- `snmp` shows SNMP configuration
- `status` shows domain status

`-h` displays help for this command.

Description

Displays the configuration and status of the domain. Some of the information is not configurable, or is configurable only through the platform, such as the access control list (ACL), SNMP enabled, and COD parameters. This command without any arguments displays all the parameter values configured by the `setupdomain` command.

See Also

`setupdomain`, `showdate`, `setupplatform`, `showplatform`

Examples

To display parameters assigned by the `setupdomain` command, type `showdomain` (CODE EXAMPLE 3-65).

CODE EXAMPLE 3-65 `showdomain` Command Example

```
schostname:A> showdomain

Domain  Solaris Nodename   Domain Status   Keyswitch
------  ----------------   -------------   ---------
A       -                  Powered Off     off

diag-level = default
post-tolerate-ce = false
verbosity-level = min
error-level = max
interleave-scope = within-board
interleave-mode = optimal
reboot-on-error = true
hang-policy = reset
max-panic-diag-limit = mem2
OBP.use-nvramrc? = <OBP default>
OBP.auto-boot? = <OBP default>
OBP.error-reset-recovery = <OBP default>

Loghost for Domain A:
Log Facility for Domain A: local0

SNMP Agent: enabled
Domain Description:
Domain Contact:
Trap Hosts:
Public Community String:
Private Community String:

ACL for Domain A: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9

PROC RTUs reserved for domain A: 0

schostname:A>
```

To display the verbose output of parameters assigned by the `setupdomain` command, type `showdomain -v` (CODE EXAMPLE 3-66).

CODE EXAMPLE 3-66 `showdomain -v` Command Example

```
schostname:A> showdomain -v

Domain  Solaris Nodename   Domain Status   Keyswitch
------  ----------------   -------------   ---------
A       -                  Powered Off     off

Domain Boot Parameters
----------------------
diag-level = default
post-tolerate-ce = false
verbosity-level = min
error-level = max
interleave-scope = within-board
interleave-mode = optimal
reboot-on-error = true
hang-policy = reset
max-panic-diag-limit = mem2
OBP.use-nvramrc? = <OBP default>
OBP.auto-boot? = <OBP default>
OBP.error-reset-recovery = <OBP default>

                     MAC Address         HostID
                     -----------------   --------
Domain A             xx:xx:xx:xx:xx:xx   xxxxxxxx

Loghosts
--------
Loghost for Domain A:
Log Facility for Domain A: local0

SNMP
----
SNMP Agent: enabled
Domain Description:
Domain Contact:
Trap Hosts:
Public Community String:
Private Community String:

SNMP packets received: 0
SNMP packets sent: 0
SNMP traps sent: 0
```

CODE EXAMPLE 3-66 `showdomain -v` Command Example *(Continued)*

```
ACLs
----
ACL for Domain A: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9

COD
---
PROC RTUs reserved for domain A: 0

schostname:A>
```

To display boot parameter information (CODE EXAMPLE 3-67), type:

CODE EXAMPLE 3-67 `showdomain -p bootparams` Example Displaying Boot Parameter Information

```
schostname:A> showdomain -p bootparams

diag-level = default
post-tolerate-ce = false
verbosity-level = min
error-level = max
interleave-scope = within-board
interleave-mode = optimal
reboot-on-error = true
hang-policy = reset
OBP.use-nvramrc? =<OBP default>
OBP.auto-boot? = <OBP default>
OBP.error-reset-recovery = <OBP default>

schostname:A>
```

`showenvironment`

Displays the current environmental status, temperatures, currents, voltages, and fan status, for the system (platform shell) or the current domain (domain shell).

Scope

platform shell, domain shell

Syntax for the Platform Shell

`showenvironment [-ltvuw] [-d` *domainID*`] [-p` *part*`]`

`showenvironment [-ltvuw]` *board_name*

`showenvironment -h`

Syntax for the Domain Shell

`showenvironment [-ltvuw] [-p` *part*`]`

`showenvironment [-ltvuw]` *board_name* [*board_name . . .*]

`showenvironment -h`

Options/Parameters

board_name is the name of the board. Displays information for this board. You can type multiple board names.

`-d` *domainID* is a, b, c, or d. Displays information for boards assigned to that domain.

`-h` displays help for this command.

`-l` displays the limits that apply to each selected measurement. These values are the threshold for each measurement. Exceeding the threshold causes the status to display `Max` or `Min`.

`-p` *part* shows a specific part (TABLE 3-15):

TABLE 3-15 Parts for `showenvironment -p` *part*

Report	Description
currents	Displays currents (power supplies only)
fans	Displays fan states.
faults	Displays values that are suspected to be invalid.
temps	Displays temperatures only.
voltage	Displays voltages only.

`-t` prints header titles.

`-u` updates data now. Polls all sensors for new values.

`-v` is verbose mode. This is the equivalent of the options `-l` and `-w`.

`-w` shows warning thresholds.

Description

Displays the current environmental status, temperatures, currents, voltages, and fan status for the system (platform shell) or the current domain (domain shell). In a domain shell, the domain can only display boards that have been assigned to the domain, such as CPU/Memory boards and I/O assemblies. The board(s) must be powered on. Boards that are not powered on are not monitored and have no environmental data.

This command displays the minimum and maximum values. If these values are exceeded, the component (and the domain using the component) will be shut down. It also shows the warning levels when the system controllers warns you that you are approaching the limits.

TABLE 3-16 describes the `showenvironment` output headers and values.

TABLE 3-16 `showenvironment` Output Header Description

Header	Value	Description
Slot		Slot ID
Device		Device being monitored by the sensor.
Sensor		Component that measures the environmental data of the device.
Value		Value returned by the sensor.

TABLE 3-16 `showenvironment` Output Header Description *(Continued)*

Header	Value	Description
Units		Applicable unit for the sensor.
	Degrees C	Celsius
	Volts DC	Volts
	Amps	Amps
Age		Age in seconds of the reading being displayed.
Status		Values for Status. See the Value column. When the values listed in the Value column are exceeded, a warning message will be logged.
	WARNING LOW	
	NOTICE Low	
	OK	
	NOTICE High	
	WARNING HIGH	
	error	
	failed	
	ignored	
	unavailable	
LoWarn		Displayed only with the `-v` and `-w` options. Lists values that when exceeded a warning message will be logged.
HiWarn		Displayed only with the `-v` and `-w` options. Lists values that when exceeded a warning message will be logged
Min		Displayed with the `-v` and `-l` options. This is not a warning message. Min is the lowest value when the component will be powered off (including any domains using it) to protect it.
Max		Displayed with the `-v` and `-l` options. This is not a fault condition. Max is the highest value when the component will be powered off (including any domains using it) to protect it.

Example

The `showenvironment` command display for the platform shell returns information on the slot number, device, sensor, value, units, age, and status, as shown in CODE EXAMPLE 3-68.

CODE EXAMPLE 3-68 `showenvironment` Platform Shell Example

```
schostname:SC> showenvironment

Slot    Device    Sensor    Value  Units     Age     Status
------- --------- --------- ------ --------- ------- ------
SSC0    SBBC 0    Temp. 0   47     Degrees C   4 sec OK
SSC0    CBH 0     Temp. 0   53     Degrees C   4 sec OK
SSC0    SSC 0     Temp. 0   33     Degrees C   4 sec OK
SSC0    SSC 0     Temp. 1   33     Degrees C   4 sec OK
SSC0    SSC 0     Temp. 2   37     Degrees C   4 sec OK
SSC0    Board 0   1.5 VDC 0 1.50   Volts DC    4 sec OK
SSC0    Board 0   3.3 VDC 0 3.37   Volts DC    4 sec OK
SSC0    Board 0   5 VDC 0   5.05   Volts DC    4 sec OK
PS0     48 VDC 0  Current 0 10.47  Amps        3 sec OK
PS0     48 VDC 0  48 VDC 0  56.07  Volts DC    3 sec OK
PS0     48 VDC 0  Temp. 0   33     Degrees C   3 sec OK
PS0     48 VDC 1  Current 0 0.78   Amps        3 sec OK
PS0     48 VDC 1  48 VDC 0  54.76  Volts DC    3 sec OK
PS1     48 VDC 0  Current 0 9.61   Amps        3 sec OK
PS1     48 VDC 0  48 VDC 0  56.07  Volts DC    3 sec OK
PS1     48 VDC 0  Temp. 0   28     Degrees C   3 sec OK
PS1     48 VDC 1  Current 0 0.15   Amps        3 sec OK
PS2     48 VDC 0  Current 0 11.50  Amps        2 sec OK
PS2     48 VDC 0  48 VDC 0  56.07  Volts DC    2 sec OK
PS2     48 VDC 0  Temp. 0   26     Degrees C   2 sec OK
PS2     48 VDC 1  Current 0 0.15   Amps        2 sec OK
PS2     48 VDC 1  48 VDC 0  54.76  Volts DC    2 sec OK
FT0     Fan 0     Cooling 0 Low                2 sec OK
FT1     Fan 0     Cooling 0 Low                1 sec OK
FT2     Fan 0     Cooling 0 Low                1 sec OK
RP0     Board 0   1.5 VDC 0 1.51   Volts DC    9 sec OK
RP0     Board 0   3.3 VDC 0 3.39   Volts DC    9 sec OK
RP0     Board 0   Temp. 0   32     Degrees C   9 sec OK
RP0     Board 0   Temp. 1   33     Degrees C   9 sec OK
RP0     SDC 0     Temp. 0   71     Degrees C   9 sec OK
RP0     AR 0      Temp. 0   59     Degrees C   9 sec OK
.
.
.
```

CODE EXAMPLE 3-68 `showenvironment` Platform Shell Example *(Continued)*

```
/N0/SB0 Board 0   1.5 VDC 0 1.51   Volts DC    8 sec OK
/N0/SB0 Board 0   3.3 VDC 0 3.29   Volts DC    8 sec OK
/N0/SB0 SDC 0     Temp. 0   66     Degrees C   9 sec OK
/N0/SB0 AR 0      Temp. 0   52     Degrees C   9 sec OK
/N0/SB0 DX 0      Temp. 0   59     Degrees C   9 sec OK
/N0/SB0 DX 1      Temp. 0   61     Degrees C   9 sec OK
/N0/SB0 DX 2      Temp. 0   61     Degrees C   9 sec OK
/N0/SB0 DX 3      Temp. 0   59     Degrees C   9 sec OK
/N0/SB0 SBBC 0    Temp. 0   61     Degrees C   9 sec OK
/N0/SB0 Board 1   Temp. 0   29     Degrees C   9 sec OK
/N0/SB0 Board 1   Temp. 1   30     Degrees C   9 sec OK
/N0/SB0 CPU 0     Temp. 0   63     Degrees C   9 sec OK
/N0/SB0 CPU 0     Core 0    1.36   Volts DC    9 sec OK
/N0/SB0 CPU 1     Temp. 0   65     Degrees C   9 sec OK
/N0/SB0 CPU 1     Core 1    1.36   Volts DC    9 sec OK
/N0/SB0 SBBC 1    Temp. 0   47     Degrees C   9 sec OK
/N0/SB0 Board 1   Temp. 2   30     Degrees C   9 sec OK
/N0/SB0 Board 1   Temp. 3   30     Degrees C   9 sec OK
/N0/SB0 CPU 2     Temp. 0   67     Degrees C   9 sec OK
/N0/SB0 CPU 2     Core 2    1.36   Volts DC    9 sec OK
/N0/SB0 CPU 3     Temp. 0   65     Degrees C   9 sec OK
/N0/SB0 CPU 3     Core 3    1.36   Volts DC    9 sec OK
/N0/SB2 Board 0   1.5 VDC 0 1.51   Volts DC    9 sec OK
/N0/SB2 Board 0   3.3 VDC 0 3.27   Volts DC    9 sec OK
/N0/SB2 SDC 0     Temp. 0   61     Degrees C   9 sec OK
/N0/SB2 AR 0      Temp. 0   49     Degrees C   9 sec OK
/N0/SB2 DX 0      Temp. 0   57     Degrees C   9 sec OK
/N0/SB2 DX 1      Temp. 0   60     Degrees C   9 sec OK
/N0/SB2 DX 2      Temp. 0   62     Degrees C   9 sec OK
/N0/SB2 DX 3      Temp. 0   61     Degrees C   9 sec OK
.
.
.
/N0/SB2 Board 1   Temp. 2   28     Degrees C   9 sec OK
/N0/SB2 Board 1   Temp. 3   28     Degrees C  10 sec OK
/N0/SB2 CPU 2     Temp. 0   58     Degrees C  10 sec OK
/N0/SB2 CPU 2     Core 2    1.34   Volts DC   10 sec OK
/N0/SB2 CPU 3     Temp. 0   56     Degrees C  10 sec OK
/N0/SB2 CPU 3     Core 3    1.34   Volts DC   10 sec OK
/N0/IB6 Board 0   1.5 VDC 0 1.51   Volts DC    9 sec OK
/N0/IB6 Board 0   3.3 VDC 0 3.33    Volts DC    9 sec OK
.
.
.
/N0/IB8 SBBC 0    Temp. 0   59     Degrees C  10 sec OK
/N0/IB8 IOASIC 0  Temp. 0   60     Degrees C  10 sec OK
/N0/IB8 IOASIC 1  Temp. 1   54     Degrees C  10 sec OK
```

`showerrorbuffer`

Shows the contents of the error buffer.

Scope

platform shell

Syntax

```
showerrorbuffer [-p [-n nnn]]
```

```
showerrorbuffer -h
```

Options/Parameters

`-p` displays the system error messages stored in the persistent system error buffer of SC V2s. This option applies to systems configured with SC V2s, which provide persistent storage of log messages and system errors.

`-n` *nnn* displays a specified number (where *nnn* is an integer) of error messages in chronological order. For example, `-p -n 5` displays the last five error messages in the persistent system error buffer.

`-h` displays help for the command.

Description

This command captures error message information detected by the system hardware error registers and stores them in an error buffer.

All Sun Fire midrange systems have a dynamic showerrorbuffer that provides short-term storage of system error records. Once the system errors are recorded in the message log buffer, system error records are cleared automatically from the dynamic showerrorbuffer whenever more space is required.

Midrange systems configured with SC V2s have both dynamic and persistent showerrorbuffers. The persistent showerrorbuffer captures the system errors that occur and store the system error records until the showerrorbuffer is full. Once the persistent showerrorbuffer is full, any new system error records will overwrite the existing error records in the persistent buffer, starting with the records at the beginning of the buffer.

- If your system is configured with SC V2s, which feature a persistent error buffer, you can use the `-p` and `-n` options to display the messages stored in the persistent storage buffer. This information is maintained even after a system reboot occurs.
- Systems that have a small error buffer (where the SCs do not have enhanced memory) cannot maintain persistent error messages. However, if your domains are set to reboot automatically upon error, the output from the `showerrorbuffer` command displays errors messages that otherwise might be lost when your domains are rebooted.

You and your service provider can use this command to obtain information for troubleshooting purposes.

See Also

None.

Examples

CODE EXAMPLE 3-69 shows the error resulting from an interconnect test.

CODE EXAMPLE 3-69 `showerrorbuffer` Example Output for an Interconnect Test

```
schostname:SC> showerrorbuffer
ErrorData[0]
 Date: Tue Jun 04 11:11:32 PDT 2002
 Device: /partition0/domain0/SB0/ar0
 Register: InterconnectTestError[0x40] : 0x00080400
                FailBit [15:00] : 0x400
              FailGroup [31:16] : 0x8
ErrorData[1]
 Date: Tue Jun 04 11:11:42 PDT 2002
 Device: /partition0/domain0/IB8/ar0
 Register: InterconnectTestError[0x40] : 0x00020001
                FailBit [15:00] : 0x1
              FailGroup [31:16] : 0x2
ErrorData[2]
 Date: Tue Jun 04 11:11:42 PDT 2002
 Device: /partition0/domain0/IB8/ar0
 Register: InterconnectTestError[0x40] : 0x00040001
                FailBit [15:00] : 0x1
              FailGroup [31:16] : 0x4
```

CODE EXAMPLE 3-70 shows a hardware error.

CODE EXAMPLE 3-70 `showerrorbuffer` Example Output for a Hardware Error

```
schostname:SC> showerrorbuffer
ErrorData[0]
  Date: Thu Jul 25 11:13:30 PDT 2002
  Device: /SSC0/sbbc0/systemepld
  Register: FirstError[0x10] : 0x0200
            SB0 encountered the first error
ErrorData[1]
  Date: Thu Jul 25 11:13:30 PDT 2002
  Device: /partition0/domain0/SB0/bbcGroup0/repeaterepld
  Register: FirstError[0x10]: 0x0040
            sbbc0 encountered the first error
ErrorData[2]
  Date: Thu Jul 25 11:13:30 PDT 2002
  Device: /partition0/domain0/SB0/sdc0
  Register: SafariPortError[0x200] : 0x00000001
              ParBidiErr [00:00] : 0x1 ParityBidi error
ErrorData[3]
  Date: Thu Jul 25 11:13:30 PDT 2002
  Device: /partition0/domain0/SB0/bbcGroup0/sbbc0
  Register: ErrorStatus[0x80] : 0x00000100
                  SafErr [09:08] : 0x1 Fireplane device asserted an error
```

CODE EXAMPLE 3-71 shows the last two hardware errors stored in the persistent system error buffer.

CODE EXAMPLE 3-71 `showerrorbuffer` Example Output – Persistent Error Information

```
schostname:SC> showerrorbuffer -p -n 2
  Date: Mon Jan 12 13:44:41 PST 2004
  Device:/partition0/domain1/SB4/bbcGroup1/cpuCD/cpusafariagent1/cheetahplusplus0
  ErrorID: 0x73a2113f
  Register: AFSR_2 (high)[0x591] : 0x00080000
                    PERR [19:19] : 0x1

  Date: Mon Jan 12 13:44:41 PST 2004
  Device: /partition0/domain1/SB4/bbcGroup1/cpuCD/cpusafariagent1/cheetahplusplus0
  ErrorID: 0x7372102f
  Register: EMU C[0x521] : 0x00000004
                  CPQ_TO [02:02] : 0x1 CPQ system bus timeout
```

`showescape`

Display the current escape sequence.

Syntax

```
showescape
showescape -h
```

Options/Parameters

`-h` shows help for this command.

Description

Shows the current escape sequence.

See Also

`setescape`

Example

CODE EXAMPLE 3-72 shows the escape sequence.

CODE EXAMPLE 3-72 `showescape` Example Output

```
schostname:SC>showescape
#.
```

`showfailover`

Displays system controller (SC) and clock failover status.

Scope

platform shell

Syntax

```
showfailover -v
```

```
showfailover -h
```

Options/Parameters

`-v` is verbose mode. Displays all available command information, which includes both SC and clock failover status. If the failover status is considered as degraded (explained in the Description section), this option will display any boards that can be controlled by the main SC but not the spare.

`-h` displays help for the command.

Description

Enables you to monitor the state of the SC and clock failover. The SC failover state can be one of the following:

- `enabled and active` - SC failover is enabled and functioning normally.
- `disabled` - SC failover has been disabled due to an operator request (`setfailover off`) or because a failover has occurred.
- `enabled but not active` - SC failover is enabled, but certain components, such as the spare SC or the centerplane between the main and spare, are not in a failover-ready state (available and responding).
- `degraded -` The SC failover status is *degraded* when both the main and the spare SC are running different firmware versions and the following conditions exist:
  - The main SC has a higher firmware version than the spare.
  - A board in the system can be controlled by the main SC but not the spare.

The clock failover state can be one of the following:

- `enabled` - Clock failover is enabled.
- `disabled` - Clock failover has been automatically disabled due to a hardware problem.

See Also

`setfailover`, `setupplatform`, `showsc`, `showlogs`, `showplatform`

Examples

- Failover is enabled and functioning:

CODE EXAMPLE 3-73 `showfailover` Command Indicating Failover is Functioning Normally

```
schostname:SC> showfailover
SC Failover: enabled and active.
```

- Failover is disabled:

CODE EXAMPLE 3-74 `showfailover` Command Indicating Failover Has Been Disabled

```
schostname:SC> showfailover
SC Failover: disabled
```

- SC and clock failover status:

CODE EXAMPLE 3-75 `showfailover` Command in Verbose Mode

```
schostname:SC> showfailover -v
SC:SSC1
Main System Controller
SC Failover: enabled and active.
Clock failover enabled.
```

- Failover is enabled but inactive because the system is not in a failover-ready state:

CODE EXAMPLE 3-76 `showfailover` Command Indicating SC Failover is Inactive

```
schostname:SC> showfailover
SC Failover: enabled but not active.
```

`showfru`

Displays the field-replaceable units (FRUs) currently installed in a Sun Fire midrange system.

Scope

platform shell

Syntax

`showfru [-v] -r` *record*

`showfru -h`

Options/Parameters

`-h` displays help for this command.

`-v` is verbose mode.

`-r` *record* where *record* is a FRU identification (FRUID) record. A FRUID record contains specific information about a FRU. The supported record type is `manr` (manufacturing record).

Description

Lists the FRUs in a Sun Fire midrange system. This information is used by your service provider to track the FRUs installed.

In some cases, it may take a few minutes for the FRU information to be displayed. If so, a message tells you to wait while FRU data is being read.

Examples

CODE EXAMPLE 3-77 shows manufacturing record output for FRUs in a Sun Fire midrange system.

CODE EXAMPLE 3-77 `showfru -r manr` Command Output

```
schostname:SC> showfru -r manr
Component         Part #         Serial Date       Time                Vend
---------         ------         ------ ----       ----                ----
SSC0              501-5407-06-50 001304 01/03/2001 01:35:36/PST        012c
ID0               501-4970-03-08 000071 12/30/2000 07:50:35/PST        001c
PS0               300-1459-03-01 D00040 11/29/2000 12:18:53/PST        009d
PS1               300-1459-03-01 D00041 11/28/2000 11:48:36/PST        009d
PS2               300-1459-03-01 D00493 11/27/2000 09:21:48/PST        009d
FT0               540-3639-01-02 WM1035 12/29/2000 10:41:32/PST        028f
FT1               540-3639-01-02 WM105L 01/18/2001 11:57:09/PST        028f
FT2               540-3639-04-02 WM100M 08/15/2000 09:37:47/PDT        028f
RP0               501-4953-08-50 001298 12/11/2000 11:35:59/PST        012c
RP2               501-4953-08-50 001480 12/08/2000 10:06:34/PST        012c
/N0/SB0           501-4362-07-50 007072 12/24/2000 03:11:01/PST        012c
/N0/SB2           501-4362-06-04 004011 08/11/2000 03:39:54/PDT        012c
/N0/SB0/P0/B0/D0 501-5401-06-01 021929 12/31/2000 04:00:00/PST        7800
/N0/SB0/P0/B0/D1 501-5401-06-01 021923 12/31/2000 04:00:00/PST        7800
/N0/SB0/P0/B0/D2 501-5401-06-01 021821 12/31/2000 04:00:00/PST        7800
/N0/SB0/P0/B0/D3 501-5401-06-01 021888 12/31/2000 04:00:00/PST        7800
/N0/SB0/P1/B0/D0 501-5401-06-01 021922 12/31/2000 04:00:00/PST        7800
/N0/SB0/P1/B0/D1 501-5401-06-01 021921 12/31/2000 04:00:00/PST        7800
/N0/SB0/P1/B0/D2 501-5401-06-01 021870 12/31/2000 04:00:00/PST        7800
/N0/SB0/P1/B0/D3 501-5401-06-01 021939 12/31/2000 04:00:00/PST        7800
.
.
.
/N0/SB0/P2/E1     370-4125-01-35 391A6H 12/03/2000 10:55:47/PST        00ce
/N0/SB0/P3/E0     370-4125-01-35 391A6A 12/03/2000 10:57:29/PST        00ce
/N0/SB0/P3/E1     370-4125-01-35 391A5Y 12/03/2000 10:59:48/PST        00ce
/N0/IB6           501-4404-08-50 002404 12/21/2000 04:31:04/PST        012c
IB8               501-4868-04-07 000165 10/03/2000 09:55:38/PDT        012c
```

`showkeyswitch`

Displays the keyswitch setting.

Scope

platform shell, domain shell

Syntax for the Platform Shell

`showkeyswitch [-v] -d` *domainID*

`showkeyswitch -h`

Syntax for the Domain Shell

`showkeyswitch [-v]`

`showkeyswitch -h`

Options/Parameters

`-h` displays help for this command.

`-v` is verbose mode.

-`d` *domainID* (platform shell only) is `a`, `b`, `c`, or `d`.

See Also

`setkeyswitch`, `showdomain`, `showplatform`

Description

Displays the keyswitch setting.

Examples

CODE EXAMPLE 3-78 `showkeyswitch` Example Showing the Keyswitch Set to On

```
schostname:A> showkeyswitch
keyswitch is: on
```

CODE EXAMPLE 3-79 `showkeyswitch` Example Showing the Keyswitch Set to Off

```
schostname:A> showkeyswitch
keyswitch is: off
```

CODE EXAMPLE 3-80 `showkeyswitch` Example Showing the Keyswitch Set to Standby

```
schostname:A> showkeyswitch
keyswitch is: standby
```

`showlogs`

Displays the system controller logged events stored in the system controller message buffer.

Scope

platform shell, domain shell

Syntax for the Platform Shell

`showlogs [-d` *domainID*`] [-p [-f` *filter*`][-n` *nnn*`]] [-v]`

`showlogs -h`

Syntax for the Domain Shell

`showlogs [-v]`

`showlogs -h`

Options/Parameters

`-h` displays help for this command.

`-d` *domainID* (platform shell only) specifies a domain (`a`, `b`, `c`, or `d`).

`-p` displays the log messages retained in persistent storage for systems configured with SC V2s.

`-f` *filter* indicates a certain type of log message is to be displayed, where *filter* is one of the following:

- `alert` – alert messages
- `critical` – critical messages
- `emergency` – emergency messages
- `error` – error messages
- `fault` – fault messages
- `warning` – warning messages

`-n` *nnn* displays a specified number *nnn* (integer) of messages in chronological order. For example, `-p -n 5` displays the last five log messages stored.

`-v` is verbose.

Description

Displays the log messages for the platform and each domain that are stored in the dynamic buffer of the system. Message storage in this dynamic buffer is temporary:

- Once the dynamic buffer is filled, the old messages that are logged to the consoles are overwritten by the new messages (these are messages from the system controller, not the Solaris operating environment).
- The dynamic buffer is cleared when you reboot the system controller or when a loss in power occurs.

In systems with SC V2s, messages in the dynamic buffer that have the following severity levels are retained in persistent storage, and will survive a system reboot or loss in power: `.alert`, `.error`, `.emerg`, `.warning`, and `.critical`. If the persistent storage buffer becomes full, any new messages will wrap to the beginning of the buffer and the existing messages at the beginning of the buffer will be overwritten by the newest messages.

There is a separate log for each system controller in the platform and for each domain. If the loghost has been configured (the platform and each domain are configured separately—using the `setupplatform` command for the platform loghost and `setupdomain` for each domain loghost), then the messages will also be logged to the loghost for storage.

See Also

`setupdomain`, `setupplatform`, `showdomain`, `showplatform`

Example

CODE EXAMPLE 3-81 displays the output of the `showlogs` command, run at the platform shell after rebooting the system.

CODE EXAMPLE 3-81 Sample Output of the `showlogs` Command Run After Rebooting the System

```
schostname:SC> showlogs
Dec 01 16:50:32 hostname Platform.SC: [ID 827764 local0.notice] Boot: ScApp 5.17.0, RTOS 38
Dec 01 16:50:38 hostname Platform.SC: [ID 595276 local0.notice] Clock Source: 75MHz
```

CODE EXAMPLE 3-81 Sample Output of the `showlogs` Command Run After Rebooting the System

```
Dec 01 16:51:02 hostname Platform.SC: [ID 367723 local0.notice] Chassis is in single
partition mode.
Dec 01 16:53:40 hostname Platform.SC: [ID 530026 local0.notice] Main System Controller
Dec 01 16:53:48 hostname Platform.SC: [ID 588369 local0.notice] Starting SNMP agent.
Dec 01 16:53:49 hostname Platform.SC: [ID 942133 local0.notice] Starting telnet server ...
Dec 01 16:53:49 hostname Platform.SC: [ID 710711 local0.notice] SC Failover: disabled
Dec 01 16:54:01 hostname Platform.SC: [ID 728126 local0.notice] Clock failover enabled.
Dec 01 16:54:55 hostname Platform.SC: [ID 862046 local0.warning] Frame Manager connected.
ID: 0003ba:1d57f6
```

CODE EXAMPLE 3-82 show the persistent logs for a system with SC V2s, when the `showlogs` command is run at the platform shell.

CODE EXAMPLE 3-82 Sample `showlogs` Persistent Log Output

```
schostname:SC> showlogs -p

Jan 05 10:55:29 schostname Platform.SC: [ID 207559 local0.error] SB2/P2/B0/D2:
SepromSection(constructor): Invalid CRC observed=00 expected=59
Jan 05 10:55:30 schostname Platform.SC: [ID 494536 local0.error] SB2/P2/B0/D3:
SepromSection(constructor): Invalid Header Length offset=1800 seg_count=5
headerLen=22
Jan 05 10:55:30 schostname Platform.SC: [ID 328118 local0.error] SB2/P2/B0/D3:
SepromSection(constructor): Invalid CRC observed=00 expected=59
.
.
.
Jan 05 10:55:51 schostname Platform.POST: [ID 232495 local0.error] SB2/P0 has
invalid DFRUID status: -16
.
.
.
Jan 05 11:08:11 schostname Platform.SC: [ID 353467 local0.crit] ErrorMonitor:
Domain B has a SYSTEM ERROR
Jan 05 11:08:13 schostname Platform.SC: [ID 325303 local0.error] [AD] Event: SF4800
     CSN: 041H3371 DomainID: B ADInfo: 1.SCAPP.17.0\
     Time: Mon Jan 05 11:08:13 PST 2004
     FRU-List-Count: 0; FRU-PN:  ; FRU-SN:  ; FRU-LOC: UNRESOLVED
     Recommended-Action: Service action required

Jan 06 14:01:26 schostname Platform.SC: [ID 695969 local0.crit] Domain B is
currently paused due to an error.  This domain must be turned off via
"setkeyswitch off" to recover
Jan 06 14:18:54 schostname Platform.POST: [ID 416500 local0.error]
/partition0/domain1/IB8/bbcGroup0/pci/schizo0:  has invalid DFRUID status: 16
.
```

CODE EXAMPLE 3-82 Sample `showlogs` Persistent Log Output *(Continued)*

```
.
Jan 12 13:41:45 schostname Platform.SC: [ID 430298 local0.warning] WARNING: Host
ID information is missing
Jan 12 13:42:14 schostname Platform.SC: [ID 273116 local0.error] PS0:
SepromSection.readSegments: invalid section header data seg_offs = 1f00 seg_len
= 0100
.
.
.
Jan 14 17:46:03 schostname Platform.POST: [ID 416500 local0.error]
/partition0/domain1/IB8/bbcGroup0/pci/schizo0:  has invalid DFRUID status: 16
```

CODE EXAMPLE 3-83 show persistent log output in which only critical messages are displayed.

CODE EXAMPLE 3-83 Sample `showlogs` Persistent Log Output, Critical Messages Only

```
hostname:SC> showlogs -p -f critical

Jan 05 11:08:11 schostname Platform.SC: [ID 353467 local0.crit] ErrorMonitor:
Domain B has a SYSTEM ERROR
Jan 05 11:08:13 schostname Platform.SC: [ID 695969 local0.crit] Domain B is
currently paused due to an error.  This domain must be turned off via
"setkeyswitch off" to recover
Jan 06 11:06:28 schostname Platform.SC: [ID 353467 local0.crit] ErrorMonitor:
Domain B has a SYSTEM ERROR
.
.
.
Jan 06 15:16:09 schostname Platform.SC: [ID 695969 local0.crit] Domain B is
currently paused due to an error.  This domain must be turned off via
"setkeyswitch off" to recover
Jan 12 09:23:30 schostname Platform.SC: [ID 353467 local0.crit] ErrorMonitor:
Domain B has a SYSTEM ERROR
Jan 12 09:23:37 schostname Platform.SC: [ID 695969 local0.crit] Domain B is
currently paused due to an error.  This domain must be turned off via
"setkeyswitch off" to recover
```

CODE EXAMPLE 3-84 shows persistent log output displayed for a specified number of messages. The output displays the last five messages stored.

CODE EXAMPLE 3-84 Sample `showlogs` Persistent Log Output, Five Messages Displayed

```
hostname:SC> showlogs -p -n 5

Jan 12 13:43:44 schostname Platform.POST: [ID 416500 local0.error]
/partition0/domain1/IB8/bbcGroup0/pci/schizo0:  has invalid DFRUID status: 16
Jan 12 13:44:41 schostname Platform.SC: [ID 806378 local0.error] /N0/SB4: marked
as Failed! errorReg = 0x0040
"setkeyswitch off" to recover
Jan 12 13:44:52 schostname Platform.SC: [ID 797753 local0.error] [AD] Event: SF4800
    CSN: 041H3371 DomainID: B ADInfo: 1.SCAPP.17.0
    Time: Mon Jan 12 13:44:45 PST 2004
    FRU-List-Count: 1
    ASIC.CHEETAH.EMU_NCPQ_TO.7062119f:
           FRU-PN: 5014362; FRU-SN: 004419; FRU-LOC: /N0/SB4
    Recommended-Action: Service action required

Jan 14 17:44:53 schostname Platform.POST: [ID 416500 local0.error]
/partition0/domain1/IB8/bbcGroup0/pci/schizo0:  has invalid DFRUID status: 16
Jan 14 17:46:03 schostname Platform.POST: [ID 416500 local0.error]
/partition0/domain1/IB8/bbcGroup0/pci/schizo0:  has invalid DFRUID status: 16
```

`showplatform`

Displays the configuration and information for the platform and domain status.

Scope

platform shell

Syntax

`showplatform [-v] [-d` *domainID*`] [-p` *part*`]`

`showplatform [-p` *part*`]`

`showplatform -h`

Options/Parameters

`-h` displays help for this command.

`-d` *domainID* specifies a domain to show (`a`, `b`, `c`, or `d`).

`-p` *part* displays information on a specific part. Use the options in this list when you specify a domain with the `-d` option. These options are the same as the `-p` *part* options to `showdomain`.

where:

- `acls` shows the access control list (ACL).
- `bootparams` shows boot parameter information.
- `cod` shows the Capacity on Demand (COD) parameters.
- `loghosts` shows the log hosts for the domain.
- `mac` shows MAC addresses for the domain.
- `snmp` shows SNMP configuration for the domain.
- `status` shows domain status.

`-p` *part* displays information on a specific part. Use the options to `-p` *part* when you *do not* specify a domain with the `-d` option.

where:

- `acls` shows the access control list (ACL).
- `cod` shows the Capacity on Demand (COD) parameters.
- `frame` shows FrameManager information.
- `hostid` shows the HostID/MAC address swap information

Note – The `HostID/MAC Address Swap` parameter information is displayed only when you specify the `hostid` option. See the "Description" section of this command for details.

- `loghosts` shows log hosts for the platform.
- `mac` shows the MAC addresses, HostID, and the platform's serial number for the domains and the system controllers.
- `network` displays the network settings, including whether remote access to the system controller has been enabled or disabled, whether the Sun Fire Link is enabled or disabled (displayed only if your system is configured with a Sun Fire Link board), and the idle connection timeout value
- `partition` shows the partition mode.
- `scpost` shows the system controller POST diag level.
- `snmp` shows Simple Network Management Protocol (SNMP) information for the platform.
- `sntp` shows the SNTP server.
- `status` shows domain status for all domains.
- `usiv+` shows UltraSPARC IV+ support. This support is available in one domain per partition (domains `a`, `c`) only.

`–v` is verbose mode.

Description

This command displays all configuration variables for this platform, including:

- Network attributes of the platform
- ACLs for each domain
- COD configuration
- Domain status
- SNMP status
- SNTP server
- UltraSPARC IV+ support
- Failover status
- Remote connection type
- Idle connection timeout
- Loghosts
- Partition configuration
- Other values that are used but are not configurable such as the MAC address, host ID, and system serial number
- HostID/MAC address swap information is displayed only when you specify the `-p hostID` option with the `showplatform` command. If you swapped the HostID/MAC address between domains (see the `setupplatform` command) the domains will reflect the swapped HostID/MAC addresses. The HostID/MAC address mapping mode is listed as manual.

If the HostID/MAC addresses have not been swapped, the mapping mode is listed as automatic, indicating that the HostID/MAC addresses reflect the automatic assignments made by the system. For details on the HostID/MAC address swap parameter, refer to the *Sun Fire Midrange Systems Platform Administration Manual*.

See Also

`setupplatform`, `showdate`

Examples

CODE EXAMPLE 3-85 `showplatform` Output for a Sun Fire 6800 System

```
schostname:SC> showplatform

Domain   Solaris Nodename   Domain Status   Keyswitch
------   ----------------   -------------   ---------
A        -                  Powered Off     off
B        -                  Powered Off     off
C        -                  Powered Off     off
D        -                  Powered Off     off

The system controller is configured to be on a network.
Network settings: DHCP
Hostname: schostname
IP Address: xxx.xxx.xxx.xxx
Netmask: xxx.xxx.xxx.x
Gateway: xxx.xxx.xxx.xxx
DNS Domain: domainname
Primary DNS Server: xxx.xxx.x.xxx
Secondary DNS Server: xxx.xxx.x.xxx
Connection type: ssh
Idle connection timeout: No timeout
Sun Fire Link Enabled: no
Loghost for Platform:
Log Facility for Platform:local0

SNTP server: ntp1

SNMP Agent: disabled
Chassis Description: Sun Fire 6800
Chassis Contact:
Chassis Location:
```

CODE EXAMPLE 3-85 `showplatform` Output for a Sun Fire 6800 System *(Continued)*

```
ACL for Domain A: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain B: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain C: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain D: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9

Chassis HostID: xxxxxxxx
PROC RTUs installed: x
PROC Headroom Quantity: x
PROC RTUs reserved for domain A: x
PROC RTUs reserved for domain B: x
PROC RTUs reserved for domain C: x
PROC RTUs reserved for domain D: x

SC POST diag Level: off
SC Failover is enabled
Logical Hostname: sp1-sc

Chassis is in dual partition mode.

Frame information is not available.

schostname:SC>
```

CODE EXAMPLE 3-86 shows the `showplatform -v` command output.

CODE EXAMPLE 3-86 `showplatform -v` Output for a Sun Fire 6800 System

```
schostname:SC> showplatform -v

Domain   Solaris Nodename   Domain Status   Keyswitch
------   ----------------   -------------   ---------
A        -                  Powered Off     off
B        -                  Powered Off     off
C        -                  Powered Off     off
D        -                  Powered Off     off

Network
-------
The system controller is configured to be on a network.
Network settings: DHCP
Hostname: schostname
IP Address: xxx.xxx.xxx.xx
Netmask: xxx.xxx.xxx.x
Gateway: xxx.xxx.xxx.xxx
DNS Domain: domainname
Primary DNS Server: xxx.xxx.x.xxx
```

CODE EXAMPLE 3-86 `showplatform -v` Output for a Sun Fire 6800 System *(Continued)*

```
Secondary DNS Server: xxx.xxx.x.xxx
Connection type: ssh
Idle connection timeout : No timeout

              MAC Address          HostID
              -----------------    --------
Domain A      xxxxxxxxxxxxxxxxx    xxxxxxxx
Domain B      xxxxxxxxxxxxxxxxx    xxxxxxxx
Domain C      xxxxxxxxxxxxxxxxx    xxxxxxxx
Domain D      xxxxxxxxxxxxxxxxx    xxxxxxxx
SSC0          xxxxxxxxxxxxxxxxx    xxxxxxxx
SSC1          xxxxxxxxxxxxxxxxx    xxxxxxxx

System Serial Number: xxxxxxxx
Chassis HostID: xxxxxxxx

Loghosts
--------
Loghost for Platform:
Log Facility for Platform: local0

SNMP
----
SNMP Agent: enabled
Chassis Description: Sun Fire 6800
Chassis Contact:
Chassis Location:
Trap Hosts:
Public Community String: P-public
Private Community String: P-private

SNMP packets received: x
SNMP packets sent: xxx

ACLs
----
ACL for Domain A: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain B: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain C: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9
ACL for Domain D: SB0 SB1 SB2 SB3 SB4 SB5 IB6 IB7 IB8 IB9

COD
---
Chassis HostID: xxxxxxxx
PROC RTUs installed: x
PROC Headroom Quantity: x
PROC RTUs reserved for domain A: x
```

CODE EXAMPLE 3-86 `showplatform -v` Output for a Sun Fire 6800 System *(Continued)*

```
PROC RTUs reserved for domain B: x
PROC RTUs reserved for domain C: x
PROC RTUs reserved for domain D: x

SNTP
-----
SNTP server: ntp1


SC
--
SC POST diag Level: off
SC Failover is enabled
Logical Hostname: sp1-sc

Partition Mode
--------------
Chassis is in dual partition mode.

Frame information is not available.

schostname:SC>
```

In CODE EXAMPLE 3-85 and CODE EXAMPLE 3-86, the Domain Status header has the following possible states:

- Powered Off
- Standby
- Running POST
- Active
- Active - OpenBoot PROM
- Active - Booting
- Active - Solaris
- Active - Halted
- Active - Reset
- Active - Panicking
- Active - Debugger
- Not Responding
- Paused due to an error

For a description of these states, see "Domain Status" on page 7.

CODE EXAMPLE 3-87 shows `showplatform -p hostid` command output.

CODE EXAMPLE 3-87 `showplatform -p hostid` Output after a HostID/MAC Address Swap

```
schostname:SC> showplatform -p hostid

               MAC Address         HostID
               -----------------   --------
Domain A       08:00:20:d8:88:99   80d88899
Domain B       08:00:20:d8:88:9c   80d8889c
Domain C       08:00:20:d8:88:9b   80d8889b
Domain D       08:00:20:d8:88:9a   80d8889a
SSC0           08:00:20:d8:88:9d   80d8889d
SSC1           08:00:20:d8:88:9e   80d8889e

System Serial Number: xxxxxxxx
Chassis HostID: xxxxxxxx
HostID/MAC address mapping mode: manual

schostname:SC>
```

CODE EXAMPLE 3-87 shows `showplatform -p hostid` command output after the HostID/MAC addresses have been restored to their original domains.

CODE EXAMPLE 3-88 `showplatform -p hostid` Output after a HostID/MAC Address Restore

```
schostname:SC> showplatform -p hostid

               MAC Address         HostID
               -----------------   --------
Domain A       08:00:20:d8:88:99   80d88899
Domain B       08:00:20:d8:88:9a   80d8889a
Domain C       08:00:20:d8:88:9b   80d8889b
Domain D       08:00:20:d8:88:9c   80d8889c
SSC0           08:00:20:d8:88:9d   80d8889d
SSC1           08:00:20:d8:88:9e   80d8889e

System Serial Number: xxxxxxxx
Chassis HostID: xxxxxxxx
HostID/MAC address mapping mode: automatic

schostname:SC>
```

`showresetstate`

Shows that state of CPUs after a reset or a RED MODE trap.

Scope

domain shell

Syntax

`showresetstate [-w | -s | -v] [-f` *url*`]`

Options/Parameters

`-w` displays the contents of register windows. The first windows displayed in the output are the stack trace for the failing CPU. The windows that are not part of the stack trace follow in the same order.

`-s` displays the contents of the secondary save area. Normally, the command displays the contents of the primary save area that holds the registers from the time of the first failure. If more than one failure occurs on a CPU, the secondary save area holds the selected registers from the time of the last failure.

`-v` displays a summary of all available registers. Includes the registers available with:

- `-s` and `-w` options
- Default display (no options)

`-f` specifies a URL, which must use the ftp protocol.

url is the directory that will contain the output from this command. The output is stored in a file with a name in the format

hostname`.resetstate.`*yymmddHHMMSS*

and can be reviewed by Sun service providers to analyze a failure or problem. The URL must point to a directory and not a file. Examples are:

`ftp://`*userid:password@hostname*`//`*path*

`ftp://`*hostname*`/`*path*

For details on specifying ftp protocol URLs, see "Using FTP URLs" on page 8.

Description

Prints a summary report of the contents of registers from the save areas of every CPU in the domain that has a valid saved state. By specifying certain options, the format and content of the printed report can be controlled. The information displayed is used by your service provider for troubleshooting purposes.

With no options, the command displays the following registers:

- Register window (globals, outs, locals, ins) that was current at the time of the failure.
- `CANSAVE`, `CANRESTORE`, `OTHERWIN`, and other selected privileged registers.
- `AFAR` and `AFSR`.
- `TICK` and some related CPU-specific registers.

Since a CPU that does not have a saved state does not produce output, there is no output when CPUs are reset or have not failed. After a failure, the saved state is available until the domain restarts the OpenBoot PROM, either by rebooting the Solaris operating environment or by turning the virtual keyswitch from off to on.

See Also

`reset`

`showsc`

Shows the system controller and clock failover status, version, and uptime.

Scope

platform shell

Syntax

`showsc [-v]`

`showsc -h`

Options/Parameters

`-h` displays help for this command.

`-v` is verbose mode.

Description

Shows the version of ScApp and the SC uptime.

Example

In CODE EXAMPLE 3-89, `RTOS` is the Real Time Operating System for the system controller and `ScApp` is the system controller application. Also, in this example the system controller description is followed by V2, which indicates that the system controller has enhanced memory.

CODE EXAMPLE 3-89 `showsc` Command

```
schostname:SC> showsc

SC: SSC0
Main System Controller V2
SC Failover: disabled.
Clock failover enabled.

SC date: Sun Jan 25 17:39:59 PST 2004
```

CODE EXAMPLE 3-89 `showsc` Command *(Continued)*

```
SC uptime: 5 days 5 hours 46 minutes 20 seconds

ScApp version: 5.17.0
RTOS version: 38
```

`ssh-keygen`

Generates Secure Shell (SSH) host keys or displays the SSH host key fingerprint.

Scope

Platform shell.

Syntax

```
ssh-keygen [-l] [-t <rsa|dsa>]
```

```
ssh-keygen [-r]
```

```
ssh-keygen [-h]
```

Options/Parameters

`-l` shows the host key fingerprint, default is RSA.

`-t` specifies the type of host key, default is RSA.

`-r` regenerates host key.

`-h` displays help for this command.

Description

Generates SSH host keys or displays the host key fingerprint on the SC.

When the SSH server has been enabled on the SC, the firmware checks whether an SSH host key exists. If not, the firmware generates (automatically) a pair of RSA private/public host keys. If you want DSA host keys on the SC, you must invoke `ssh-keygen` manually with the `-t` switch.

Once generated, new host keys take effect (get loaded into memory) after:

- The SSH server is (re-) enabled
- The system is rebooted and the SSH server is enabled
- The `restartssh` command is invoked while the SSH server is enabled

It is good security practice for well-managed machines to get new host keys periodically. If you suspect that the key has been compromised, you can run the `ssh-keygen` command to regenerate system host keys. Host keys, once generated, can only be replaced and not deleted. For newly generated host keys to take effect, the SSH server must be restarted, either by running the `restartssh` command or with a reboot.

Since host keys are large, 1 Kbyte in size, it is difficult to verify an entire host key. `ssh-keygen` can be used to display a host key fingerprint, which is the output of the md5 message-digest algorithm presented as a sequence of 16 octets printed as hexadecimal with lowercase letters and separated by colons. See CODE EXAMPLE 3-90.

Since host keys are stored on the SC, they get backed up with `dumpconfig` and can be restored by `restoreconfig` (By default, the `dumpconfig` command saves keys in encrypted format). When an SC failover occurs, the keys get copied to the redundant SC. In other words, the main SC and the redundant SC share the same set of SSH keys.

You can regenerate the host keys at any time by running `ssh-keygen`. If the host key already exists, you must specify the `-r` switch.

See Also

`restartssh`

Example

CODE EXAMPLE 3-90 `ssh-keygen` Command Example

```
schostname:SC> ssh-keygen -l
''d1:c1:30:22:5d:6c:11:09:de:38:40:9e:20:a5:b3:10''
```

`testboard`

Tests the CPU/Memory board.

Scope

platform shell, domain shell

Syntax

```
testboard board_name
```

```
testboard -h
```

Options/Parameters

`-h` displays help for this command.

board_name is sb0-sb5 for CPU/Memory boards.

Description

Tests the CPU/Memory board name specified on the command line. This board cannot be used in an active domain and must be powered on. If running from a domain, the board must be assigned to the domain. The Repeater boards required to run the domain must also be powered on.

The `testboard` command runs tests at the levels set in the domain. Testing a CPU/Memory board is most accurately done when it is part of a domain. It is best if the CPU/Memory board can be tested by POST. However, if that is not possible, testing in the domain is the next best test.

Caution – When testing a CPU/Memory board that is assigned to a domain and the domain is active, there is a very slight possibility that problems can occur with the active domain. To prevent this from being a problem, unassign the board from the domain and test the board from the platform shell.

See Also

`showcomponent`, `showboards`

Example

To test CPU/Memory board, `sb0` in the domain A shell (CODE EXAMPLE 3-91), type:

CODE EXAMPLE 3-91 `testboard` Example Testing CPU/Memory Board `sb0`

```
schostname:A> testboard sb0
<Command output not shown.>
```

Glossary

auto-diagnosis (AD) engine	A firmware feature that detects and diagnoses hardware errors that affect the availability of a platform and its domains.
Capacity on Demand (COD)	Capacity on Demand (COD) is an option that provides additional processing resources (CPUs) when you need them. These additional CPUs are provided on COD CPU/Memory boards that are installed in your system. You can access the COD CPUs after you purchase the COD right-to-use (RTU) licenses for them. For details on COD, refer to the *Sun Fire Midrange Systems Platform Administration Manual*
CHS	Component health status. The component maintains information regarding its health, including the diagnosis information generated by the *auto-diagnosis (AD) engine*.
domain shell	With the domain shell, you have access to system controller commands that you need to perform on a domain. There are up to four domain shells (A through D). The domain shell prompt is *schostname*`:A>` (or `B>`, `C>`, or `D>)`.
domain	A set of one or more system boards that is capable of booting the operating system and running independently of other domains. Domains do not depend on each other and do not interact with each other.
domain console	If the Solaris operating environment or the OpenBoot PROM is running, you can access the domain console. With the domain console, you can have the `ok`, `login`, `#`, or `%` prompts.
DSA	Digital Signature Algorithm standard, published by the National Institute of Standards and Technology. The digital authentication standard of the U.S. government.

environmental monitoring	All systems have a large number of sensors that monitor temperature, voltage, and current. The system controller polls devices in a timely manner and makes the environmental data available. The system controller will shut down various components to prevent damage.
failover	The switchover of the main system controller to its spare or the system controller clock source to another system controller clock source when a failure occurs in the operation of the main system controller or the clock source.
keyswitch	See virtual domain keyswitch.
partition	A group of Repeater boards, also referred to as a *segment*, that are used together to provide communication between CPU/Memory boards and I/O assemblies in the same domain.
persistent logging	The storage of certain SC-generated message logs and system errors in the NVRAM of SC V2s (enhanced-memory system controllers).
platform shell	The platform shell enables access to the entire system and provides: configuration control, environmental status, ability to rearrange domains, ability to power on and off power grids, ability to change the system controller password, and other generic system controller functions.
POST	Power-on self-test. This is the program that takes uninitialized system hardware and probes and tests its components, configures what seems worthwhile into a coherent initialized system, and hands it off to the OpenBoot PROM.
RSA	Created by Rivest, Shamir, and Adleman (RSA), it is the most popular form of public-key cryptography.
SSH	A client/server protocol that provides secure access to a shell.
SRAM	Static Random Access Memory. A type of high-speed memory device used for Ecache modules.
system controller	The system controller consists of the System Controller board and the system controller software. The system controller software manages platform and domain resources, monitors and controls the platform and domains, configures domains and the platform, provides access to domain consoles, provides the date and time to the Solaris operating environment, provides the reference clock signal used throughout the system, provides console security, performs domain initialization, provides a mechanism for updating the firmware on the boards installed in the system, and provides an external management interface using SNMP.
System Controller board	A board containing a CPU (central processing unit), which oversees operation of the system and provides clocks and the console bus. The Sun Fire midrange systems support two System Controller boards.

TOD	See virtual time of day (TOD).
virtual domain keyswitch	The system controller provides a virtual keyswitch for each domain. The `setkeyswitch` command controls the position of the virtual keyswitch for each domain.
virtual time of day (TOD)	The TOD/NVRAM chip is located on the System Controller board. The system controller provides TOD services for each domain and for the system controller.

Index