



Sun Fire™ V20z and Sun Fire V40z Servers

Server Management Guide

Sun Microsystems, Inc.
www.sun.com

Part No. 817-5249-17
January 2007, Revision A

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

Copyright 2004-2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

This document and the product to which it pertains are distributed under licenses restricting their use, copying, distribution and decompilation. No part of the product or of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

AMD, Opteron, HyperTransport, and PowerNow! are trademarks of Advanced Micro Devices, Inc.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, JumpStart, Solaris and Sun Fire are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and in other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and in other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government Rights—Commercial use. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2004-2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, États-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuelle relatants à la technologie qui est décrite dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains énumérés à <http://www.sun.com/patents> et un ou les brevets plus supplémentaires ou les applications de brevet en attente aux États-Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

AMD, Opteron, HyperTransport, et PowerNow! sont des marques de fabrique ou des marques déposées de Advanced Microdevices, Inc.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux États-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, JumpStart, Solaris et Sun Fire sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux États-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciées de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE «EN L'ÉTAT» ET TOUTES AUTRES CONDITIONS, DÉCLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISÉE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE À LA QUALITÉ MARCHANDE, À L'APTITUDE À UNE UTILISATION PARTICULIÈRE OU À L'ABSENCE DE CONTREFAÇON.



Contents

Preface xiii

1. Introduction to the Sun Fire V20z and Sun Fire V40z Servers 1

Systems Overview 1

User Documentation 2

Acronyms 2

Server Management 4

 Service Processor 4

Operator Panel 8

 Characteristics of Operator Panel Displays 9

User Groups 10

 Users 10

 Passwords Files 10

Systems Management Tasks 11

Initial Setup of the SP 12

 Task I: Assigning Network Settings to the SP 12

 Task II: Securing the SP 16

 Task III: Enabling IPMI Access on the Server 18

 Task IV: Enabling IPMI LAN Access 21

 Upgrading the Linux Kernel 22

Site Integration	23
Daisy-Chaining the Servers	24
Platform Drivers and Applications	26
Updating Software	28
Selecting and Setting Up the File Server	29
Configuring and Starting the Update Server Application	30
Identifying Packages for Update	32
Updating the SP Base Package	32
Updating the SP Value-Add Package	33
Updating the BIOS	34
Updating the Diagnostics	35
Autoconfiguring the SP	36
Files Copied During the Autoconfiguration Process	37
Determining SP and Platform Network MAC Addresses	39
Systems Management Console Features	39
Configuring Network Settings	40
Starting and Stopping the Platform OS	41
Configuring SMTP Event Notification	42
Configuring Directory Services	43
Mapping Directory Service Groups	45
Creating Keytab Files for ADS	46
Configuring Date and Time	47
Configuring SSL	48
Monitoring System Status	50
2. IPMI Server Management	57
Intelligent Platform Management Interface	57
Baseboard Management Controller	58
Manageability	59

Functional Overview	59
IPMI Compliance and LAN Channel Access	60
Sensors	62
Alerting	67
Lights Out Management	67
Viewing the IPMI System Event Log	75
Clearing the IPMI System Event Log	75
IPMI Troubleshooting	76
3. SNMP Server Management	77
Simple Network Management Protocol	77
SNMP Integration	78
SNMP Management Information Base	79
Configuring SNMP on the Servers	80
Out-of-Band Management Configuration	81
SNMP Traps	84
Configuring SNMP Destinations	85
Server MIB Details	86
SNMP Troubleshooting	88
4. Further Management Information	89
Configuring Scripting Capabilities	89
Using Shell Scripts	90
Remote Scripting Using SSH	91
Configuring Multiple Systems for Scripting	91
Guidelines for Writing Server Management Command Scripts	96
Console Redirection Over the Serial Port	98
Linux-based Server	99
Solaris-based Server	102
Enabling and Disabling BIOS Console Redirection	103

Network Share Volume	104
Network Share Volume Structure	105
Serial-Over-LAN	106
Enabling or Disabling the SOL Feature on the Server	106
Starting and Stopping an SOL Session	107
Escape Sequences for Remote Console Terminal	108
A. Server Management Commands Summary	109
Using the ssh Protocol	109
Interactive Shell on the SP	109
Preface Text	110
Commands	110
Return Codes	111
B. Access Commands	113
Access Config-Sharing Subcommands	114
Access Enable Config-Sharing Subcommand	114
Access Disable Config-Sharing Subcommand	115
Access Get Config-Sharing Subcommand	116
Access Group Subcommands	117
Access Get Group Subcommand	117
Access Get Groups Subcommand	118
Access Map Subcommands	118
Access Get Map Subcommand	119
Access Map Subcommand	120
Access Unmap Subcommand	121
Access Directory Services Subcommands	122
Access Disable Service Subcommand	122
Access Enable Service Subcommand	123
Access Get Services Subcommand	125

Access Trust Subcommands	127
Access Add Trust Subcommand	127
Access Delete Trust Subcommand	129
Access Get Trusts Subcommand	130
Access Public Key Subcommands	131
Access Add Public Key Subcommand	131
Access Get Public Key Users Subcommand	133
Access Delete Public Key Subcommand	133
Access User Subcommands	135
Access Add User Subcommand	135
Access Delete User Subcommand	136
Access Get Users Subcommand	137
Access Update Password Subcommand	138
Access Update User Subcommand	139
C. Diagnostics Commands	141
Diags Cancel Tests Subcommand	142
Diags Get Modules Subcommand	143
Diags Get State Subcommand	144
Diags Get Tests Subcommand	145
Diags Run Tests Subcommand	146
Diags Start Subcommand	148
Diags Terminate Subcommand	150
D. Inventory Commands	151
Inventory Compare Versions Subcommand	152
Inventory Get Hardware Subcommand	153
Inventory Get Software Subcommand	156
Inventory Get Remote-Software Subcommand	157

Inventory Get All Subcommand 158

E. IPMI Commands 159

IPMI Disable Channel Subcommand 160

IPMI Enable Channel Subcommand 161

IPMI Disable PEF Subcommand 162

IPMI Enable PEF Subcommand 162

IPMI Get Channels Subcommand 163

IPMI Get Global Enables Subcommand 164

IPMI Get Sel Subcommand 164

IPMI Clear Sel Subcommand 166

IPMI Set Global Enable Subcommand 167

IPMI Reset Subcommand 168

F. Platform Commands 171

Platform Console Subcommands 172

Platform Console Subcommand 172

Platform Get Console Subcommand 175

Platform Set Console Subcommand 177

Platform OS State Subcommands 179

Platform Get OS State Subcommand 180

Platform Set OS State Subcommands 181

Platform Power State Subcommands 187

Platform Get Power State Subcommand 187

Platform Set Power State Subcommand 188

Platform Get Hostname Subcommand 190

Platform Get MAC Subcommand 191

Platform Get Product-ID Subcommand 192

G. Sensor Commands 193

Sensor Subcommands	193
Sensor Get Subcommand	194
Sensor Set Subcommand	197
H. Service Processor Commands	199
SP Date Subcommands	200
SP Get Date Subcommand	200
SP Set Date Subcommand	201
SP DNS Subcommands	202
SP Disable DNS Subcommand	202
SP Enable DNS Subcommand	203
SP Get DNS Subcommand	204
SP Events Subcommands	205
SP Delete Event Subcommand	205
SP Get Events Subcommand	206
SP Hostname Subcommands	208
SP Get Hostname Subcommand	208
SP Set Hostname Subcommand	209
SP IP Subcommands	210
SP Get IP Subcommand	211
SP Set IP Subcommand	212
SP JNET Address Subcommands	213
SP Get JNET Subcommand	213
SP Set JNET Subcommand	214
SP Locatelight Subcommands	215
SP Get Locatelight Subcommand	216
SP Set Locatelight Subcommand	216
SP Logfile Subcommands	217
SP Get Logfile Subcommand	217

SP Set Logfile Subcommand	218
SP Miscellaneous Subcommands	220
SP Create Test Events Subcommand	220
SP Get MAC Address Subcommand	222
SP Get Port 80 Subcommand	222
SP Autoconfigure Subcommand	229
SP Get Status Subcommand	231
SP Get TDULog Subcommand	231
SP Reboot Subcommand	233
SP Reset Subcommand	234
SP Mount Subcommands	236
SP Add Mount Subcommand	237
SP Delete Mount Subcommand	238
SP Get Mount Subcommand	239
SP SMTP Subcommands	241
SP Get SMTP Server Subcommand	241
SP Set SMTP Server Subcommand	242
SP Get SMTP Subscribers Subcommand	243
SP Update SMTP Subscriber Subcommand	244
SP SNMP Subcommands	246
SP Add SNMP-Destination Subcommand	247
SP Delete SNMP-Destination Subcommand	248
SP Get SNMP-Destinations Subcommand	249
SP Get SNMP Proxy Community Subcommand	250
SP Set SNMP Proxy Community Subcommand	250
SP Get SNMP Community Subcommand	251
SP Set SNMP Community Subcommand	252
SP SSH1 Subcommands	253
SP Disable SSH1	253

SP Enable SSH1	254
SP Get SSH1	254
Format	254
SP SSL Subcommands	255
SP Disable SSL-Required Subcommand	255
SP Enable SSL-Required Subcommand	256
SP Get SSL Subcommand	257
SP Set SSL Subcommand	257
SP Update Subcommands	259
SP Update Flash All Subcommand	259
SP Update Flash Applications Subcommand	261
SP Update Diags Subcommand	262
Index	265

Preface

This guide explains how to manage the Sun Fire™ V20z and Sun Fire V40z servers.

How This Book Is Organized

[Chapter 1](#) provides an overview of how you can manage the servers.

[Chapter 2](#) describes how to manage the servers through the Intelligent Platform Management Interface (IPMI).

[Chapter 3](#) describes how to manage the servers through the Simple Network Management Protocol (SNMP).

[Chapter 4](#) provides further management information, such as how to enable scripting capability, console redirection over serial, and serial-over-LAN.

[Appendix A](#) contains an overview of the server management commands that you can use to manage the server. Following appendixes describe each command type in detail.

[Appendix B](#) contains detailed descriptions of Access commands.

[Appendix C](#) contains detailed descriptions of Diagnostics commands.

[Appendix D](#) contains detailed descriptions of Inventory commands.

[Appendix E](#) contains detailed descriptions of IPMI commands.

[Appendix F](#) contains detailed descriptions of Platform commands.

[Appendix G](#) contains detailed descriptions of Sensor commands.

[Appendix H](#) contains detailed descriptions of service processor (SP) commands.

Using UNIX Commands

This document might not contain information about basic UNIX® commands and procedures such as shutting down the system, booting the system and configuring devices. See the following documents for this information:

- Software documentation that you received with your system
- Solaris™ Operating System documentation, which is located at:

<http://docs.sun.com>

Shell Prompts

Shell	Prompt
C shell	<i>machine-name%</i>
C shell superuser	<i>machine-name#</i>
Bourne shell and Korn shell	\$
Bourne shell and Korn shell superuser	#

Typographic Conventions

Typeface*	Meaning	Examples
AaBbCc123	The names of commands, files and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>% You have mail.</code>
AaBbCc123	What you type, when contrasted with on-screen computer output	<code>% su</code> Password:
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be superuser to do this. To delete a file, type <code>rm filename</code> .

* The settings on your browser might differ from these settings.

Related Documentation

For the most up-to-date documentation, go to this site:

http://www.sun.com/products-n-solutions/hardware/docs/Servers/Workgroup_Servers/Sun_Fire_V20z/index.html

Translated versions of some of the documentation are available in French, Simplified and Traditional Chinese, Japanese, and Korean.

Note that the English documentation is revised more frequently and might therefore be more up-to-date than the translated documentation.

Application	Title	Part Number
Safety information	<i>Important Safety Information for Sun Hardware Systems</i>	816-7190
Safety notices and international compliance certification statements	<i>Sun Fire V20z and Sun Fire V40z Servers—Safety and Compliance Guide</i>	817-5251
Hardware and system software installation	<i>Sun Fire V20z and Sun Fire V40z Servers—Installation Guide</i>	817-5246
Server maintenance	<i>Sun Fire V20z and Sun Fire V40z Servers—User Guide</i>	817-5248
Server management	<i>Sun Fire V20z and Sun Fire V40z Servers—Server Management Guide (This document)</i>	817-5249
Operating-system installation	<i>Sun Fire V20z and Sun Fire V40z Servers—Linux Operating System Installation Guide</i>	817-5250
Troubleshooting and diagnostics	<i>Sun Fire V20z and Sun Fire V40z Servers—Troubleshooting Techniques and Diagnostics Guide</i>	817-7184
Late-breaking information	<i>Sun Fire V20z and Sun Fire V40z Servers—Release Notes</i>	817-1771
Comparison of server models	<i>Differences Between Versions of the Sun Fire V20z and Sun Fire V40z Servers</i>	817-7185
Configuring pre-installed Solaris 10 OS	<i>Sun Fire V20z and Sun Fire V40z Servers—Guide for Pre-Installed Solaris 10 Operating System</i>	819-1811

Accessing Sun Documentation

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

<http://www.sun.com/documentation>

Third-Party Web Sites

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

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 V20z and Sun Fire V40z Servers—Server Management Guide, part number 817-5249-17

Introduction to the Sun Fire V20z and Sun Fire V40z Servers

Systems Overview

Strong server-management capabilities are crucial to maintaining mission-critical servers. Advance notification of problems and rapid diagnosis and correction are critical functions to an environment in which a few servers bear the bulk of the workload. The Sun Fire V20z and Sun Fire V40z servers and their extensive server-management capabilities lower costs by reducing failure and by potentially eliminating hands-on management.

This document describes how to perform remote management on the Sun Fire V20z and Sun Fire V40z servers.

The Sun Fire V20z server is an AMD Opteron™ processor-based, enterprise-class one rack unit (1U), two-processor (2P) server. The Sun Fire V40z server is also an AMD Opteron processor-based server, but it is a three rack unit (3U), four-processor (4P) server.

These servers include an embedded service processor (SP), flash memory, RAM, a separate Ethernet interface, and server-management software. They come equipped with superior server-management tools for greater control and minimum total cost of ownership. You can use the command-line interface (CLI), SNMP integration with third-party frameworks, or IPMI to configure and manage the platform with the SP. The dedicated SP provides complete operating-system independence and maximum availability of server management.

User Documentation

For the most up-to-date user documentation, for both the Sun Fire V20z and Sun Fire V40z servers, visit the following web site:

http://www.sun.com/products-n-solutions/hardware/docs/Servers/Workgroup_Servers/Sun_Fire_V20z/index.html

To verify whether a document on the site is more recent than the document that you have, refer to the final two digits (the dash-roll) of the part number for that document or review the release date of the document.

Note – A document explaining the differences among the released versions of the Sun Fire V20z and Sun Fire V40z servers is also available at this web site. Refer to part number (PN) 817-7185.

Acronyms

TABLE 1-1 defines the acronyms found in this document.

TABLE 1-1 Acronyms Used in This Manual

Acronym	Explanation
ACPI	Advanced Configuration and Power Interface
ADS	Active Directory Service
ARP	Address Resolution Protocol
BMC	baseboard management controller
CRU	customer-replaceable unit
DPC	Direct Platform Control
FRU	field-replacement unit
grub	Grand Unified Bootloader
HDT	hardware debugging tool
ICMB	Intelligent Chassis Management Bus
IPMB	Intelligent Platform Management Bus

TABLE 1-1 Acronyms Used in This Manual (*Continued*)

Acronym	Explanation
IPMI	Intelligent Platform Management Interface
KCS	Keyboard Controller Style
KVM	keyboard, video, and mouse
LAN	local area network
LILO	Linux Loader
LOM	lights out management
MIB	Management Information Base
NNM	Network Node Manager
NPS	Newisys Platform Software
PEF	platform event filter
PET	platform event trap
PRS	Power and Reset Sequencer
PXE	Preboot Execution Environment
RMCP	Remote Management Control Protocol
RPM	Red Hat Package Manager
RTC	real-time clock
SDR	sensor data record
SDRR	sensor data record repository
SEL	system event log
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOL	serial-over-LAN
SP	service processor
SSL	Secure Sockets Layer
SSU	System Setup Utility
SunMC	Sun Management Center
UART	universal asynchronous receiver/transmitter
UDP	User Datagram Protocol
UMB	Upper Memory Block

TABLE 1-1 Acronyms Used in This Manual (*Continued*)

Acronym	Explanation
UTC	Coordinated Universal Time
VPD	vital product data
WAN	wide area network

Server Management

There are two options for remotely managing a Sun Fire V20z or Sun Fire V40z server:

- lights out management (LOM) through IPMItool
- Simple Network Management Protocol (SNMP)

Service Processor

The Sun Fire V20z and Sun Fire V40z servers include a dedicated chipset for complete operating-system independence and maximum availability of server-management functions. This chipset, called a service processor (SP), is an embedded PowerPC chip providing the following:

- Environmental monitoring of the platform (such as temperatures, voltages, fan speeds, and panel switches)
- Alert messages when problems occur
- Remote control of server operations (boot, shutdown and reboot of the server's operating system, turning the server's power on and off, stopping the server's boot process in BIOS, and upgrading the BIOS)

Note – In this document, you might see references to a baseboard management controller (BMC). A BMC is a dedicated IPMI controller. The SP found in these servers is a general-purpose, embedded CPU that contains software to emulate a BMC.

The SP runs an embedded version of Linux software, and all the server-management functions are developed as standard Linux applications. Its sole purpose is to support server management; therefore, the full functionality of the operating system is not available in the SP. Many familiar applications, such as FTP and Telnet, are not provided as they are not required to support the server-management feature set.

FIGURE 1-1 shows the back panel of the Sun Fire V20z server.

FIGURE 1-2 shows the back panel of the Sun Fire V40z server.

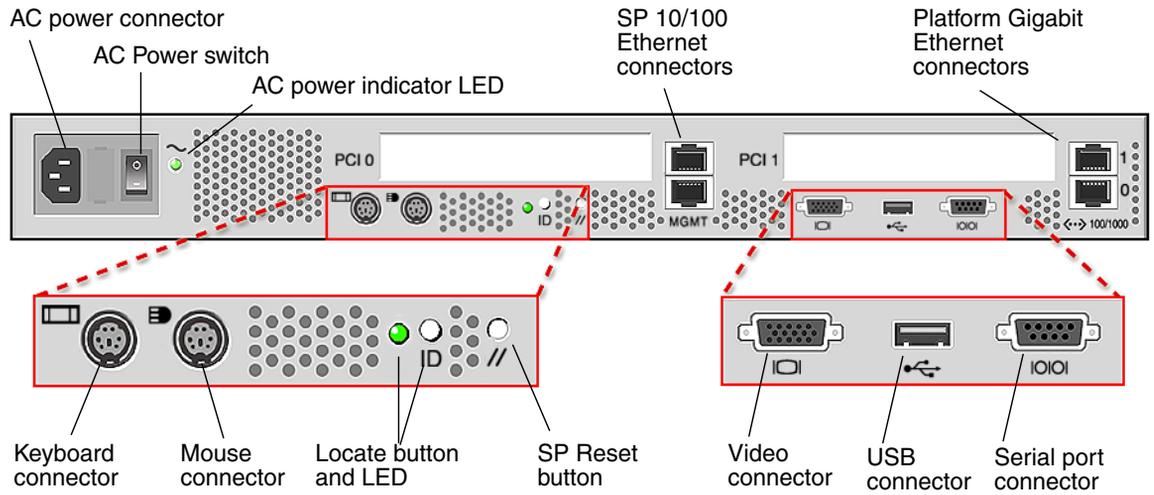


FIGURE 1-1 Sun Fire V20z Back Panel

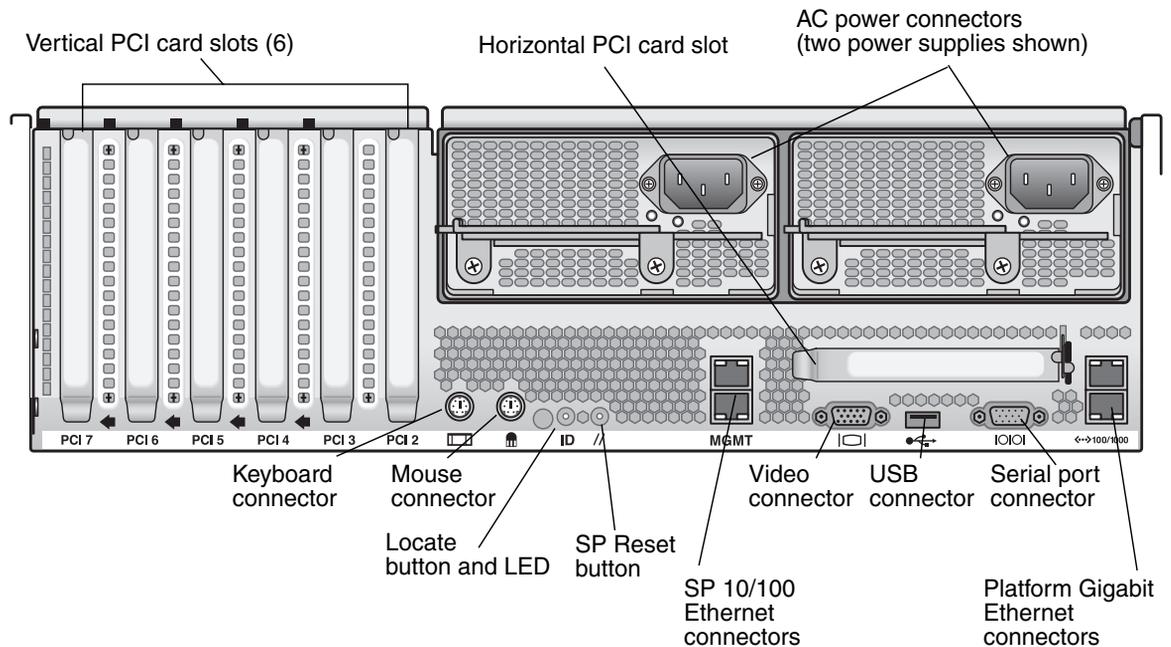


FIGURE 1-2 Sun Fire V40z Back Panel

Server-Management Interfaces

These servers include local and remote server-management capabilities through the SP; the SP supports four server-management interfaces:

- IPMI using a Keyboard Controller Style (KCS) interface and an IPMI kernel driver (in-band)
- IPMI over local area network (LAN) (out-of-band)
- SNMP integration with third-party SNMP management consoles
- Command-line interface (CLI) lights out management (LOM)

Command-Line Interface

Server-management capabilities are available from the command line.

See [Appendix A](#) for a list of server-management commands that you can use with these servers, as well as a description, the command format, a list of arguments, and a list of return codes for each command.

SSH and Scripting Capabilities

A system administrator can log in to the SP using Secure Shell (SSH) and issue commands, or more commonly, write a shell script that remotely invokes these operations.

The server-management commands enable you to efficiently manage each area of the server. From the command line, you can write data-driven scripts that automate the configuration of multiple machines. For example, a central management system can cause many servers to power on and boot at a specified time, or when a specific condition occurs.

For more information about scripting, see [“Further Management Information” on page 89](#).

SNMP Integration

Simple Network Management Protocol (SNMP) management provides remote access by SNMP-compliant entities to monitor the health and status of the server. The SP sends SNMP alerts to external management functions when warranted.

For more information about SNMP, refer to [“SNMP Server Management” on page 77](#).

The diagram in [FIGURE 1-3](#) illustrates the communication paths for the different server-management options.

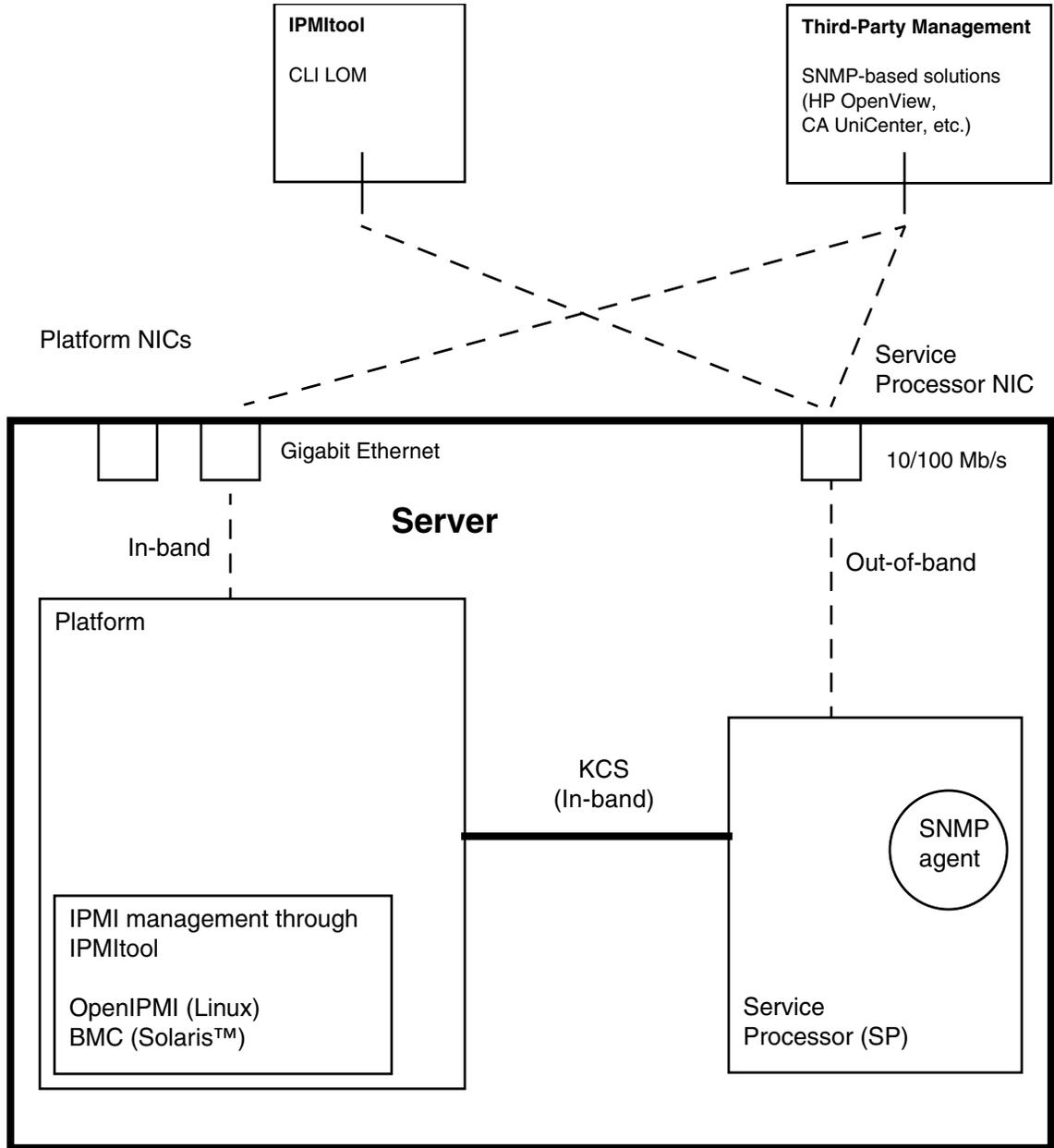


FIGURE 1-3 Diagram of the Server-Management Options

Operator Panel

You can use the operator panel to configure network settings for the SP. See [FIGURE 1-4](#) or [FIGURE 1-5](#) for the operator panel location on your server.

Note – The SP defaults to Dynamic Host Configuration Protocol (DHCP) networking if the operator panel is not interactively engaged on the first power-up.

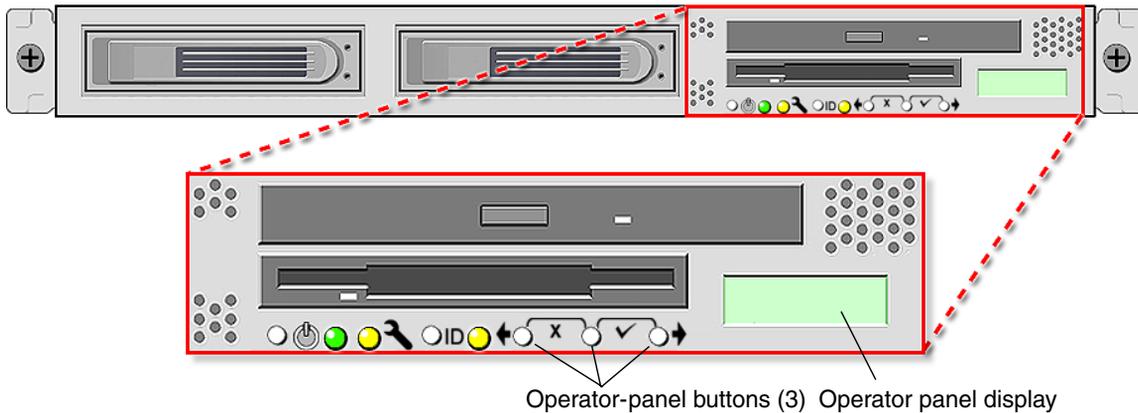


FIGURE 1-4 Sun Fire V20z Server Operator Panel and Buttons

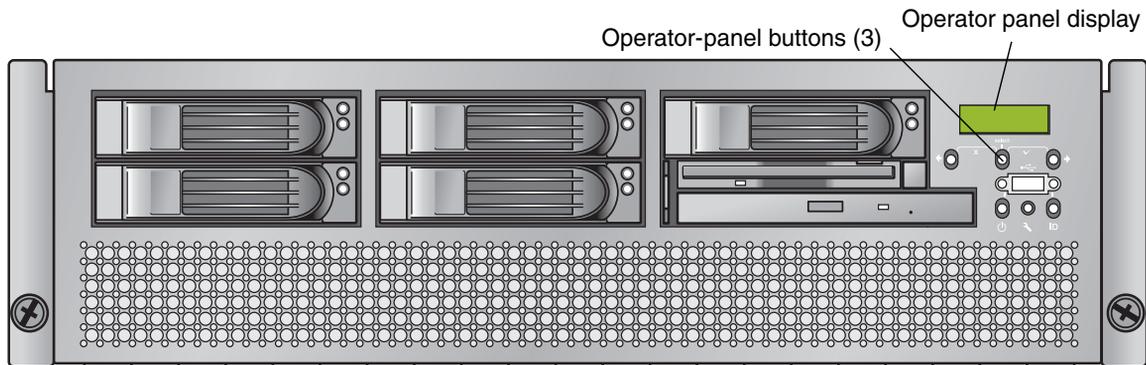


FIGURE 1-5 Sun Fire V40z Server Operator Panel and Buttons

The operator panel displays information on the operator panel display in two lines; you respond to prompts or initiate actions using the following buttons.

TABLE 1-2 Operator-Panel Buttons

Buttons	Function
	Back —Move or step backward through data options for a field. Moves through the bottom line of text, only.
	Select —Move or step forward through menus and fields that display in the top line of text and through field values for octets that display in the bottom line of text. Confirm and save a selected data option that displays in the bottom line of text. (To confirm submenu fields that require octets, use the Enter button combination.)
	Forward —Move or step forward through data options for a field. Moves through the bottom line of text, only.
	Enter —(Select plus Forward, the checkmark combination.) Confirms and saves a selected data option in submenu fields that require octets, such as IP address, netmask, or gateway.
	Cancel —(Back plus Select, the X combination.) Cancels the previous confirmation and steps backward to the previous display.

Characteristics of Operator Panel Displays

- The Enter combination (Select plus Forward) is indicated by a checkmark. This combination confirms a data choice in submenu fields that require octets, such as an IP address. You must press both buttons simultaneously, and you must release both buttons simultaneously. (In most fields, you can press Select to confirm a data choice.)
- The Cancel combination (Back plus Select) is indicated by an X. This combination cancels an action, backs up in a menu, and undoes other actions, depending on the menu. You must press both buttons simultaneously, and you must release both buttons simultaneously.
- For numerical value in octets, such as an IP address, you can press and hold the Back or Forward button to activate the auto-scrolling feature. This enables you to move through the range of numbers more quickly.
- A menu or data entry screen that displays for more than 30 seconds with no action taken is canceled, and the display returns to the idle/background state.
- For every action that you confirm, feedback is displayed to indicate success, failure, or that the action has been initiated.

For a complete list of the menu options on the operator panel, refer to “Operator Panel” in Chapter 1 in the *Sun Fire V20z and Sun Fire V40z Servers—User Guide*.

User Groups

Administrators can define several different user groups, or types, on the server. Capabilities of the different user types are defined in [TABLE 1-3](#).

For example, when you log in to the system the first time using the setup account, you must set up the initial manager account so that other user accounts can be managed. (See “[Creating the Initial Manager Account](#)” on page 16 for details.)

TABLE 1-3 User Types

User Type	Capability
monitor	Read-only access for sensor data and log displays
admin	All capabilities except user-account management and SP field upgrades
manager	All capabilities except SP field upgrade
service	SP field upgrades

Users

There are two classes of SP users: one class of users can log in to the SP through SSH; the other class of users can establish IPMI sessions to the SP.

These two classes of users are managed independently:

- Users who are created using the IPMI cannot access the SP through SSH.
- Users who log in through SSH cannot access the SP through the IPMI.

It is possible to configure the SP so that directory-services (ADS/NFS) users can log in to the SP through SSH. However, these directory-services users cannot log in to the SP through the IPMI.

Passwords Files

Passwords for local, non-IPMI users are stored in a standard Linux shadow-password file, which enhances the security of the system. The hashed passwords are in a file that is not readable by users.

Passwords for IPMI users are stored separately. The IPMI password file is not readable by users, but passwords are stored unencrypted because of limitations imposed by the IPMI authentication algorithms.

Systems Management Tasks

To accomplish most systems management tasks, you can use any of the systems management tools that are included with your server. [TABLE 1-4](#) lists some common systems management tasks and the tools that you can use to accomplish each task. For information about the systems management commands, see the appendixes in this guide. Commands documented in the online help are noted in the table. For information about running diagnostic tests, see the *Sun Fire V20z and Sun Fire V40z Servers—Troubleshooting Techniques and Diagnostics Guide*.

TABLE 1-4 Systems Management Tasks

Task	Systems Management Console	Operator Panel	Systems Management Command	SNMP	IPMI
Analyze events	Yes: online help	Yes, minimal	Yes	Yes	Yes
Autoconfigure SP	Yes	Yes	Yes	No	No
Configure directory services	Yes	No	Yes	No	No
Configure external file system	Yes	No	Yes	No	No
Configure network settings	Yes	Yes	Yes online help	No	Yes
Configure scripting capabilities	Yes	No	Yes online help	No	No
Configure SMTP event notification	Yes	No	Yes	No	N/A
Configure SP date and time settings	Yes	No	Yes	No	N/A
Configure SSL	Yes	No	Yes	No	N/A
Create initial manager account	Yes	No	Yes	No	N/A
Define default system name in Operator Panel	No	Yes	Yes	No	No
Monitoring system status	Yes: online help	Yes	Yes	Yes	Yes
Power on and off	Yes	Yes	Yes	N/A	Yes

TABLE 1-4 Systems Management Tasks (*Continued*)

Task	Systems Management Console	Operator Panel	Systems Management Command	SNMP	IPMI
Remove software	N/A	N/A	Yes	No	N/A
Run diagnostics tests	Yes	No	Yes	No	No
Run Troubleshooting Dump Utility	No	Yes	Yes	No	No
Set SP hostname	Yes	Yes	Yes	No	No
Set up network share volume	Yes	No	Yes: online help	No	N/A
Start and stop platform OS	Yes	Yes	Yes: online help	No	Yes
Update software	Yes	Yes	Yes	No	Yes: only the SP
Update SP software	Yes	Yes	Yes	No	Yes

Initial Setup of the SP

This procedure describes the steps for the initial setup of the SP.

Task I: Assigning Network Settings to the SP

This section contains two alternative methods you can use to define SP network settings:

- [“Assigning SP Network Settings Using DHCP” on page 13](#)
- [“Assigning Static SP Network Settings” on page 14](#)

Note – As an alternative, if no DHCP server or physical access is available, you can configure the SP using IPMItool in conjunction with an IPMI kernel driver. To configure your server for IPMI, perform the correct procedures for your operating system in [“Task III: Enabling IPMI Access on the Server” on page 18](#), then [“Task IV: Enabling IPMI LAN Access” on page 21](#).

Assigning SP Network Settings Using DHCP

The following procedure describes how to configure the SP network settings using DHCP from the operator panel. If your network does not use DHCP, or you want to assign a static IP address to the SP, follow the instructions in [“Assigning Static SP Network Settings” on page 14](#).

Note – This procedure assumes that you have cabled the server and powered it on as described in the *Sun Fire V20z and Sun Fire V40z Servers—Installation Guide*. At least one of the server’s SP ports must be connected to a LAN.

1. **Press any operator-panel button on the server front panel (see [FIGURE 1-6](#)).**

The operator panel displays the first menu option:

```
Menu:  
Server Menu
```

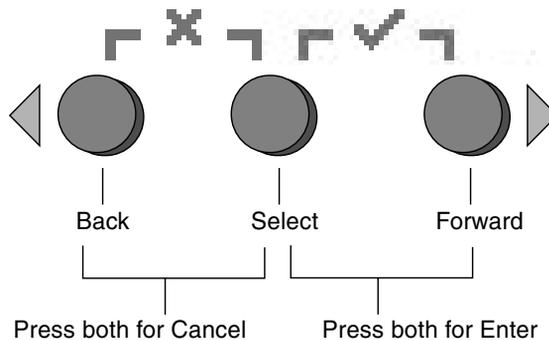


FIGURE 1-6 Operator-Panel Buttons

2. **Press the Forward button until you reach the SP menu.**

```
Menu:  
SP menu
```

3. **Press the Select button to display the SP menu options.**

```
SP Menu:  
Set SP IP info?
```

4. **Press the Select button.**

The following prompt appears with the default response:

```
SP use DHCP?  
No
```

5. **Press the Forward button to change to Yes, then press the Select button.**

6. Press the Select button at the confirmation prompt.

SP use DHCP:
Yes?

The server attempts to contact a DHCP server for an IP address. When the server receives a DHCP response, the operator panel displays the DHCP-assigned SP IP address. The SP IP address is configured and the server is ready for use.

Note – Depending on your network conditions, it may take five to ten seconds for the new IP address allocated by the DHCP server to appear in the operator panel.

Note – A prompt appears that asks whether you want to perform autoconfiguration. As an alternative to configuring an SP manually, you can run autoconfiguration, which replicates the configuration of one SP to another. Refer to [“Autoconfiguring the SP” on page 36](#) for instructions on autoconfiguration.

7. Continue with “Task II: Securing the SP” on page 16 for instructions on creating the initial manager account.

Assigning Static SP Network Settings

From the operator panel, follow these steps to configure the SP network settings using a static IP address. You must specify a subnet mask and default gateway. This example uses the following sample settings:

IP Address: 10.10.30.5
Subnet Mask: 255.255.255.0
Default Gateway: 10.10.30.254

1. Press any operator-panel button on the server front panel (see [FIGURE 1-6](#)).

The operator panel displays the first menu option:

Menu:
Server Menu

2. Press the Forward operator-panel button until you reach the SP menu.

Menu:
SP menu

3. Press the Select operator-panel button to display the SP menu options.

SP Menu:
Set SP IP info?

4. Press the Select operator-panel button.

The following prompt is displayed with the default response:

```
SP use DHCP?  
No
```

5. Press the Select operator-panel button.

The operator panel displays the following:

```
SP IP Address:  
0.0.0.0
```

6. With the cursor in the first field, increase or decrease the value using the Back or Forward operator-panel button.

This field can hold a value between 0 and 255.

```
SP IP Address:  
10.0.0.0
```

7. After reaching your desired value, press the Select operator-panel button to advance the cursor to the next field.

```
SP IP Address:  
10.0.0.0
```

8. Repeat [Step 6](#) and [Step 7](#) for each field until the desired IP address is displayed, then use the Enter button combination to save the IP Address.

The process continues to the next network setting, the subnet mask. The operator panel displays the following:

```
SP netmask:  
255.255.255.0
```

9. Edit the subnet mask setting in the same manner as you did for the IP address. When finished, use the Enter button combination to save the subnet mask.

The process continues to the next network setting, the default gateway. The operator panel displays the following:

```
SP IP Gateway  
10.10.30.1
```

10. Edit the default gateway setting in the same manner as you did for the IP address and the subnet mask. When finished, use the Enter button combination to save the default gateway.

The operator panel displays the following confirmation prompt:

```
Use new IP data:  
Yes?
```

11. Press the **Select operator-panel button to use the new data, or use the Cancel button combination to disregard the new information.**

The SP address is now configured and the server is ready for use.

Note – A prompt appears that asks whether you want to perform autoconfiguration. As an alternative to configuring an SP manually, you can run autoconfiguration, which replicates the configuration of one SP to another. Refer to [“Autoconfiguring the SP” on page 36](#) for instructions on autoconfiguration.

12. Continue with [“Task II: Securing the SP” on page 16.](#)

Task II: Securing the SP

After you install the server and configure the SP’s network settings, you must create the initial manager account. You can then perform initial configuration of the server and create additional user accounts. Only the administrator who does the initial system configuration can create the initial manager account.



Caution – The SP must be secured with a user name and password when the server is first deployed. Failure to secure the SP can expose the server to a potential denial-of-service attack through the SP network interface.

Creating the Initial Manager Account

A setup account is included with each server. This setup account has no password. When you log in to the SP the first time using the setup account, you are prompted to define the initial manager account with a password and an optional public key.

User names and passwords are strings that consist of any alphanumeric character, underscore, hyphen, or period.

- User names must be unique and must begin with an alphabetic character.
- Passwords can contain any printable character and are case sensitive.
- A user name or a password is limited to 32 characters and cannot be a null or an empty string.

There are two methods you can use to create the initial manager account:

- From a command line: see [“Creating the Initial Account From a Command Line” on page 17.](#)
- From the Systems Management (SM) console: see [“Creating the Initial Account From the SM Console” on page 17.](#)

Creating the Initial Account From a Command Line

Log in to the setup account and create the initial manager account by following this procedure:

1. **Using an SSHv1 or SSHv2 client, connect to the IP address of the SP.**
2. **Authenticate as the user `setup` with no password required.**
`# ssh spipaddress -l setup`
3. **Follow the on-screen prompts to create the initial manager account.**

After you create the initial manager account, the setup account is deleted and you are logged out of the server. You can then log in using the new initial manager account, from which you can create other user accounts.

Note – If you are prompted for a password, this indicates that the SP has already been secured with an account. If you do not know the management user name and password, you can reset the SP from the operator panel by navigating to the `SP` menu and selecting the `Use defaults` option. Note that all current settings for users and networks will be lost and the SP will reboot.

Creating the Initial Account From the SM Console

For information about the SM Console features, see [“Systems Management Console Features” on page 39](#).

To create the first manager account from the SM Console:

1. **Type the SP name or IP address as the URL or address in a browser, to enter the SM Console.**

The main SM Console screen appears.

Note – When you create the initial manager account, you are prompted to accept a license agreement. After you create the initial manager account, this prompt no longer appears.

2. **At the Create Initial Manager-Level User ID screen, type a user name for this account.**
3. **Type a password for the account.**
4. **Re-enter the password to confirm the password.**
5. **Click the checkmark button.**

6. Use the SM Console to select initial configuration options.

After you create the initial manager-level user, the Initial Configuration Checklist screen is displayed in the SM Console. This screen enables you to determine the options you want for the initial setup of the SP.

The Initial Configuration Checklist is a table that lists the SM Console menu options and the commands you use to configure each option. It also includes links to the online help that provides instructions for each option.

Note – This table is displayed only after you create the initial manager-level user. Therefore, only the administrator who initially configures the account or who resets it via the operator panel can access it.

Note – The IP address, user name, and password that you configure are referred to in subsequent examples as the *spipaddr*, *spuser*, and *sppasswd*.

Task III: Enabling IPMI Access on the Server

This section contains two alternative procedures: one for a Linux software-based server and one for a Solaris x86-based server. Use the procedure that corresponds to your OS:

- [“Enabling IPMI Access on a Linux-Based Server \(In-Band\)” on page 18](#)
- [“Enabling IPMI Access on a Solaris x86-Based Server \(In-Band\)” on page 20](#)

Enabling IPMI Access on a Linux-Based Server (In-Band)

1. **Log in to the server and authenticate as the root user.**
2. **Install the custom OpenIPMI Linux kernel driver from the Sun Fire V20z and Sun Fire V40z Servers Documentation and Support Files CD.**

The drivers are located in the CD directory `/support/sysmgmt/`.

Browse to the OS variant installed on your server. The options are:

- `redhat/rhel3` for Red Hat Enterprise Linux 3 (32-bit mode uses the architecture type `“i386”`; 64-bit mode uses the architecture type `“x86_64”`)
- `suse/sles8` for SUSE Linux Enterprise Server 8 (32-bit mode uses the architecture type `“i386”`; 64-bit mode uses the architecture type `“x86_64”`)
- `suse/sles9` for SUSE Linux Enterprise Server 9 (64-bit mode uses the architecture type `“x86_64”`)
- `suse/suse9` for SUSE Linux Professional Server 9

3. Ensure that the kernel-source RPM is already installed on your distribution by running the command:

```
# rpm -qvi kernel-source
```

If this utility reports that the kernel-source software package is not installed, install the kernel-source RPM that is current for your installed Linux distribution.

- On SUSE distributions, install the kernel-source RPM by running the command:

```
# yast2
```

- On Red Hat distributions, download the current kernel-source RPM to a temporary directory (such as /tmp). Install the package by running the command:

```
# rpm -ivh /tmp/kernel-source*.rpm
```

4. Install the OpenIPMI Linux kernel driver RPM.

- a. Browse to the OS variant installed on your server. The options are:

- redhat/rhel3 for Red Hat Enterprise Linux 3 (32-bit mode uses the architecture type "i386"; 64-bit mode uses the architecture type "x86_64")
- suse/sles8 for SUSE Linux Enterprise Server 8 (32-bit mode uses the architecture type "i386"; 64-bit mode uses the architecture type "x86_64")
- suse/sles9 for SUSE Linux Enterprise Server 9 (64-bit mode uses the architecture type "x86_64")
- suse/suse9 for SUSE Linux Professional Server 9

- b. Install the OpenIPMI RPM file by running the command:

```
# rpm -ivh openipmi*.rpm
```

Note – The kernel driver will be compiled using the kernel-source code during installation.

5. Install IPMItool.

IPMItool is the command-line interface (CLI) server-management client.

- If the installed Linux distribution uses the 32-bit "i386" architecture, run the following command:

```
# rpm -ivh ipmitool*.i386.rpm
```

- If the installed Linux distribution uses the 64-bit "x86_64" architecture, run the following command:

```
# rpm -ivh ipmitool*.x86_64.rpm
```

6. Test the IPMI kernel device driver and client application by running the following command:

```
# ipmitool -I open chassis status
```

Successful output looks similar to the following:

```
"
System Power: on
Power Overload: false
Power Interlock: inactive
Main Power Fault: false
Power Control Fault: false
Power Restore Policy: unknown
Last Power Event:
Chassis Intrusion: inactive
Front-Panel Lockout: inactive
Drive Fault: false
Cooling/Fan Fault: false
"
```

Note – On a subsequent reboot, the IPMI kernel driver might have to be loaded with the following command:

```
# modprobe ipmi_kcs_drv
```

Note – If you upgrade your Linux kernel, refer to [“Upgrading the Linux Kernel” on page 22](#).

Enabling IPMI Access on a Solaris x86-Based Server (In-Band)

1. **Log in to the server and authenticate as the root user.**
2. **Run the following command to install the LIPMI Solaris x86 kernel driver and the IPMItool management control application.**

These files are located on the Documentation and Support Files CD in the /support/sysmgmt/solaris9 directory.

```
# pkgadd -d ./
```

Confirm installation of all packages when prompted.

3. **Reboot the server.**

Task IV: Enabling IPMI LAN Access

This section contains three alternative procedures: two in-band procedures and one out-of-band procedure. Use the procedure that corresponds to your OS:

- [“Enabling IPMI LAN Access on a Linux-Based Server \(In-Band\)” on page 21](#)
- [“Enabling IPMI LAN Access on a Solaris x86-Based Server \(In-Band\)” on page 21](#)
- [“Enabling IPMI LAN Access Using an Alternative Method \(Out-of-Band\)” on page 22](#)

Enabling IPMI LAN Access on a Linux-Based Server (In-Band)

1. If the server is powered off, boot the local OS.
2. Log in to the server and authenticate as the root user.
3. Load the OpenIPMI kernel device driver (as installed in [Step 3 of “Enabling IPMI Access on a Linux-Based Server \(In-Band\)” on page 18](#)).

```
# modprobe ipmi_kcs_drv
```

4. Using the following commands in IPMITool, configure the network settings for the SP.

Note – For more information about the syntax for IPMITool commands, refer to [“Syntax” on page 68](#).

```
# ipmitool -I open lan set 6 ipaddr ipaddr
# ipmitool -I open lan set 6 netmask netmask
# ipmitool -I open lan set 6 defgw ipaddr gwipaddr
# ipmitool -I open lan set 6 password ipmipasswd
```

Enabling IPMI LAN Access on a Solaris x86-Based Server (In-Band)

1. If the server is powered off, boot the local OS.
2. Log in to the server and authenticate as the root user.
3. Using IPMITool, configure the network setting for the SP by using the following commands.

Note – For more information about the syntax for IPMItool commands, refer to [“Syntax” on page 68](#).

```
# ipmitool -I lipmi lan set 6 ipaddr ipaddr
# ipmitool -I lipmi lan set 6 netmask netmask
# ipmitool -I lipmi lan set 6 defgw ipaddr gwipaddr
# ipmitool -I lipmi lan set 6 password ipmipasswd
```

Enabling IPMI LAN Access Using an Alternative Method (Out-of-Band)

1. Using an SSHv1 client or SSHv2 client, log in to the IP address of the SP.
2. Authenticate as the newly created manager user (see [“Task II: Securing the SP” on page 16](#)”).

```
# ssh spipaddr -l spuser
```

3. Enable IPMI LAN access and assign a password when prompted.

```
# ipmi enable channel lan
# exit
```

Note – This password will be referred to as *ipmipasswd* in subsequent examples.

4. Using IPMItool, test the IPMI LAN access.

```
# ipmitool -I lan -H spipaddr -P ipmipasswd chassis status
```

Upgrading the Linux Kernel

Upgrading the installed Linux kernel to a newer version requires you to recompile the upgraded IPMI kernel device driver.

1. Install the kernel-source RPM that matches the version of the upgraded kernel binary RPM package.
2. Log in to the server and authenticate as the root user.
3. Change to the following directory.

```
# cd /usr/src/kernel-modules/openipmi
```

4. Recompile the module by running the following commands.

```
# make clean
# make
# make install
```

5. Re-test the IPMI kernel device driver and client application by running the following command.

```
# ipmitool -I open chassis status
```

Successful output should look similar to the following:

```
"
System Power: on
Power Overload: false
Power Interlock: inactive
Main Power Fault: false
Power Control Fault: false
Power Restore Policy: unknown
Last Power Event:
Chassis Intrusion: inactive
Front-Panel Lockout: inactive
Drive Fault: false
Cooling/Fan Fault: false
"
```

Note – On a subsequent reboot, the IPMI kernel driver might have to be loaded with the following command:

```
# modprobe ipmi_kcs_drv
```

Site Integration

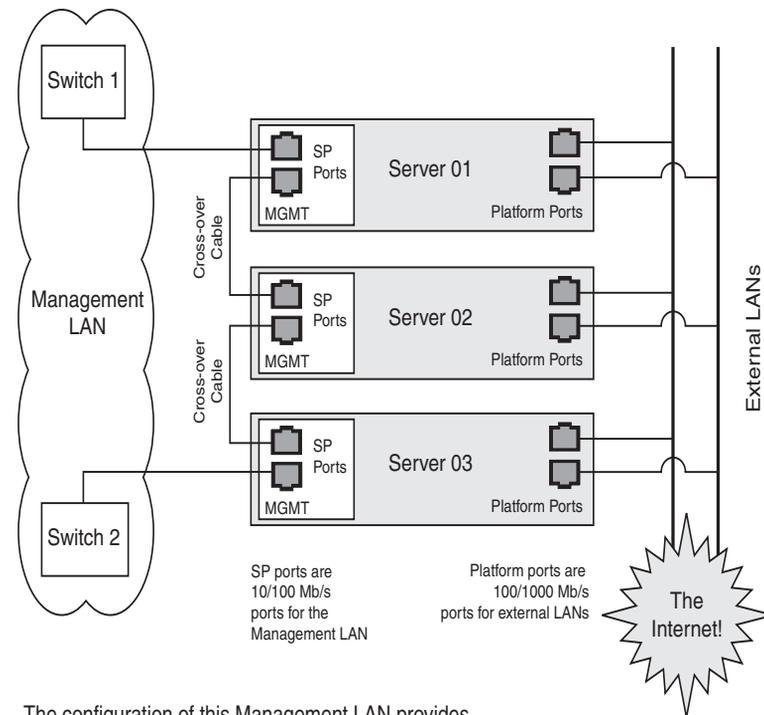
When deploying your server, ensure that you determine the best integration strategy for your environment.

The Sun Fire V20z and Sun Fire V40z servers include network connections for the SP that are separate from network connections for the platform. This enables you to configure the server so that the SP is connected to an isolated, management network and is not accessible from the production network.

Daisy-Chaining the Servers

You can interconnect multiple servers in different daisy-chain configurations by using the SP connectors to form a management LAN, as shown in [FIGURE 1-7](#), [FIGURE 1-8](#), and [FIGURE 1-9](#). The figures also show how the servers are connected to external LANs using the platform Gigabit Ethernet connectors.

Note – It is a best practice to use cross-over cables of at least one meter in length for daisy-chaining the servers.

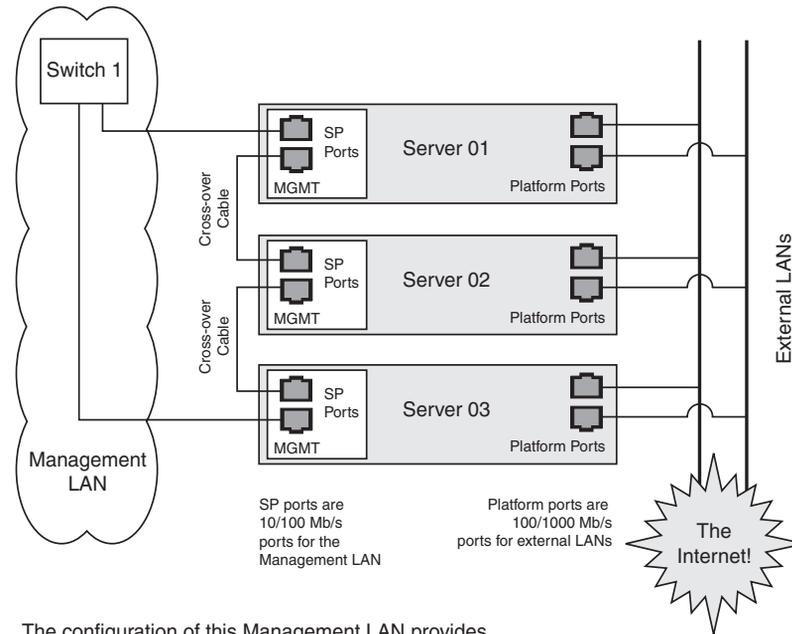


The configuration of this Management LAN provides redundancy at the switch level.

FIGURE 1-7 Daisy-Chain Architecture With Redundancy at Switch Level on the Management LAN

To interconnect the servers, you must use an RJ-45 cross-over cable. Cables can be connected to either the top or bottom SP port. To configure servers in a daisy chain, connect the first and last server in the chain to different switches.

In the configuration shown in [FIGURE 1-7](#), two managed switches that are capable of spanning-tree discovery are required to redundantly connect both the top and bottom of the chain. If the switches are not capable of spanning-tree discovery, then only connect either to the top or the bottom of the chain, but not both.

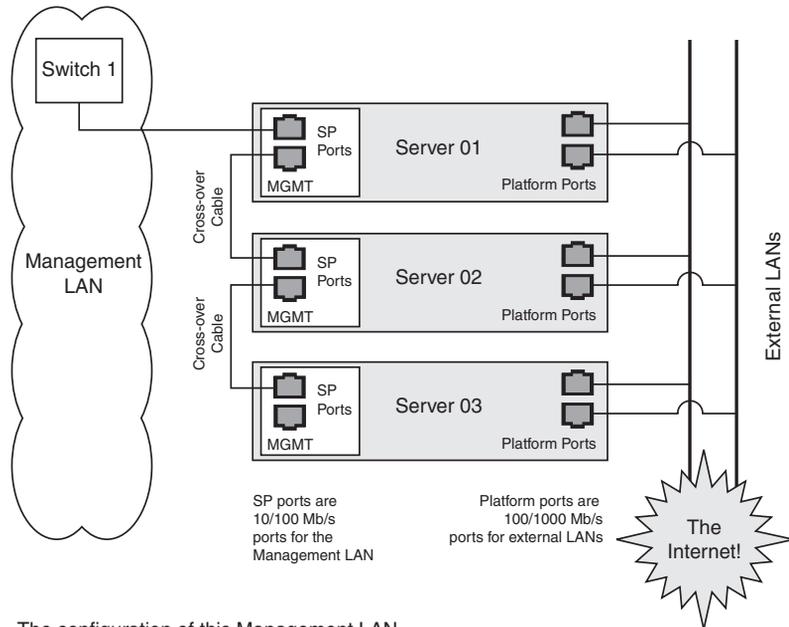


The configuration of this Management LAN provides redundancy at the SP-Port level.

FIGURE 1-8 Daisy-Chain Architecture With Redundancy at Port Level on the Management LAN

To interconnect the servers, you must use an RJ-45 cross-over cable. Cables can be connected to either the top or bottom SP port. To configure servers in a daisy chain, connect the first and last server in the chain to different switches.

In the configuration shown in [FIGURE 1-8](#), a managed switch that is capable of spanning-tree discovery is required to redundantly connect both the top and bottom of the chain. If the switch is not capable of spanning-tree discovery, then only connect either to the top or the bottom of the chain, but not both.



The configuration of this Management LAN does not provide redundancy.

FIGURE 1-9 Daisy-Chain Architecture With No Redundancy on the Management LAN

To interconnect the servers, you must use an RJ-45 cross-over cable. Cables can be connected to either the top or bottom SP port.

In the configuration shown in [FIGURE 1-9](#), no redundancy is provided on the Management LAN.

Platform Drivers and Applications

Installation of the platform drivers and applications provides the following capabilities:

- Enables communication between the SP and the platform. This allows for better control of the platform. For example, the platform can be shut down or rebooted properly, rather than forcefully, via power-downs and resets.
- Allows the platform SNMP traps to be forwarded through the SP's SNMP daemon.
- Allows the SP to monitor the health of the platform operating system when platform difficulties occur, and to attempt to handle machine check errors.

- Allows the SP to gather additional vital product data about system components.
- Allows the SP to gather inventory information about operating system software.
- Allows updates to the platform BIOS from the SP.

If you do not install the platform software, the following features will not be available from the SP:

- Ability to gracefully shut down and reboot the platform.
- Ability to receive notices of recoverable machine check events and ECC errors.
- Ability to obtain platform hostname.
- Ability to determine the current OS status.
- Ability to determine the current version and inventory of platform software.
- Ability to obtain CPU vital product data and inventory information.
- Ability to determine whether the platform is running, via the platform heartbeat.
- Ability to obtain platform-side SNMP information, if attached to the SP's SNMP server.
- Ability to set the platform JNET address with the `sp set jnet` command.

The features or characteristics below are available without installation of the platform drivers. However, they require that the SP was fully booted during the last BIOS boot:

- BIOS inventory information is available from the SP.
- SP time is synchronized to the platform.
- Optimized thermal management is available via the SP.

Also note these other important points about platform software:

- When you install a platform operating system, you can configure the language support. If you choose a language other than English, ensure that you also install the appropriate version of third-party drivers.
- When you install a platform operating system, you can configure the power state. When you choose a power state, turn off Suspend and Hibernate.
- There is a private network between the SP and the platform that supports internal communication.
 - The link-local address 169.254.101.2 is assigned to the SP.
 - The link-local address 169.254.101.3 is assigned to the platform for communication over this private network. These addresses are physically assigned, not randomly generated, and probed for conflict. You can use the `sp set jnet` command to change these IP addresses. The platform drivers must be installed in order for JNET communication to work.

Updating Software

Note – For complete information about the menu options available through the operator panel, refer to the *Sun Fire V20z and Sun Fire V40z Servers—User Guide*.

If you attempt to update the SP software using the operator panel when the IP address for the SP has not been set, the update fails. Ensure that the IP address has been set prior to attempting an update. For more information, refer to the *Sun Fire V20z and Sun Fire V40z Servers—Installation Guide*.

A new network share volume (NSV) that is installed on your network contains firmware packages. You can make these firmware packages available to a SP in either of these ways:

- The recommended method is through the Update Server, a Java™ application that transfers the packages from the NSV to the SP.
 - You can update multiple SPs, simultaneously, if you use the Update Server application.
 - You must use the Update Server application to update the SP base package.
- Another method is to use the SP to create a network file system (NFS) mount to the NSV. Once you accomplish this, the NSV and the packages it contains appear to be local to the SP and are available for update.

Note – The latest BIOS version number is never the same as the latest NSV version number, as represented in the configuration file example data lines in [“Configuring and Starting the Update Server Application”](#) on page 30.

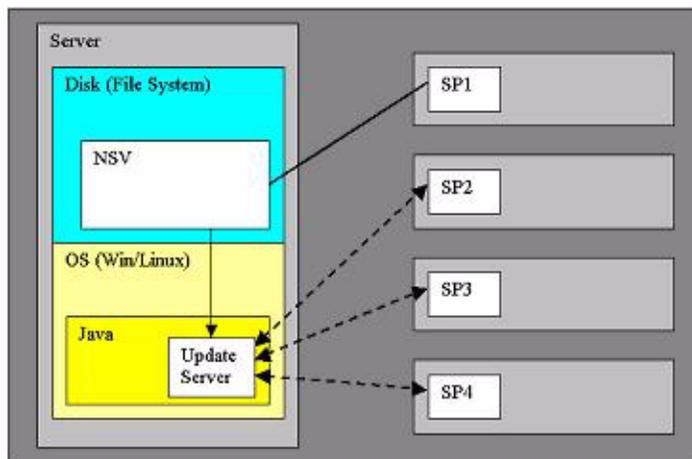


FIGURE 1-10 NSV on Network-Accessible Server

In the illustration above, the NSV has been unzipped and saved to a server that is network-accessible by the SPs that require updated packages. SP1 has mounted the NSV directly. SP2, SP3, and SP4 access the NSV through the Update Server.

Note – In this example, SP1 cannot update the SP base package without using the Update Server application.

Selecting and Setting Up the File Server

Select a server that is network-accessible to the SP(s).

Note – The Update Server application requires Java Version 1.4 software or a subsequent release that is compatible with the software. If you plan to use the Update Server, open a shell or command prompt window and type **java -version** to verify the version.

To install the NSV, follow the steps below.

1. **Download the latest version of the NSV from**
<http://www.sun.com/servers/entry/v20z/downloads.html>.
2. **Extract files from the compressed NSV to a location on your selected file server.**

Note – When you unzip a compressed file on Linux, use the `-a` switch (for example, `unzip -a filename.zip`) to force text files to convert to the target operating system’s appropriate end-of-line termination.

A new release manifest file (`releaseVersion.xml`) is added to the root directory of the NSV. See [“Network Share Volume” on page 104](#) for details about the NSV structure.

3. For Linux-based systems, ensure that the NSV directory has been exported.
4. Decide whether you will update by using the Update Server application or by using an NFS mount:
 - If you plan to use the Update Server application, go to [“Configuring and Starting the Update Server Application” on page 30](#).
 - If you plan to update by using the NFS mount method, log in to the SP and mount the NSV.

For example, if the IP address of the machine with the new NSV is 10.10.20.100 and you extracted the NSV files to a directory named `newNSV`, you would run the command:

```
sp add mount -r 10.10.20.100:/newNSV -l /mnt
```

The NSV will then be available to the SP at `/mnt/sw_images/`.

Continue with [“Identifying Packages for Update” on page 32](#).

Configuring and Starting the Update Server Application

The Update Server configuration file enables you to export multiple packages with multiple versions to one or more SPs. To select the appropriate updates, follow the instructions below.

1. Navigate to `NVS/update_server/Vx.xx` (where `Vx.xx` is the version you want) to find the configuration file.

The configuration file includes example data lines, shown below.

```
SP_BASE          V2.0.0.38 /nsv/sw_images/sp/spbase/V2.0.0.38/install.image
SP_BASE          V2.0.0.40 /nsv/sw_images/sp/spbase/V2.0.0.40/install.image
SP_VALUE_ADD     V2.0.0.38 /nsv/sw_images/sp/spvalueadd/V2.0.0.38/install.image
SP_VALUE_ADD     V2.0.0.40 /nsv/sw_images/sp/spvalueadd/V2.0.0.40/install.image
BIOS-X250Alpha  V1.27.9   /nsv/sw_images/platform/firmware/bios/V1.27.9/bios.sp
```

Note – The latest BIOS version number is never the same as the latest NSV version number, as seen in the example above.

Each data line contains three space-delimited values:

- Package type: SP-BASE, SP_VALUE_ADD, BIOS

In order to support BIOS updates for several products that each require unique BIOS firmware, the BIOS package must include the product ID. The product ID is the value that is returned by the `platform get product-id` command. It is also found in the BIOS software manifest (`swinventory.xml`) that is included in an NSV. The actual product ID used in the example above is `x250 Alpha`. When you include this in the BIOS package type in the configuration file, you must add the hyphen between BIOS and the product ID, and you must remove all spaces from the product ID string.

- Version, in standard version format: `v[major].[minor].[patch].[build]`.
 - File path: actual path and file name of an update file.
2. In the configuration file, each data line is preceded by a # sign. To indicate a file that should be updated, add the correct version number and remove the # sign at the beginning of the data lines.
 3. Navigate to the NSV folder that contains the Update Server application and start the server via the command line:

```
java -jar updateServer.jar -c updateServer.config -p <port> -l logfile.log
```

The `updateServer.jar` file is located in the `update_server` folder of the NSV.

- It is recommended that you use the `-l` flag to create a log file.
- Only the start and the end of an update transaction will be sent to the console.
- Detailed information about the update process is sent to the log file, which can be useful if you need to troubleshoot a failed update.
- By default, the server uses port number 52708.
- If this port number is in use already, use the optional `-p` flag to specify a different port.
- The Update Server application does not start if the file is not found in the specified path. Otherwise, the server is ready to receive update requests from any SP.
- The Update Server application can simultaneously accept multiple update requests from different SPs.

Identifying Packages for Update

1. To determine which packages currently are installed on an SP, run this command from the SP:

```
inventory get software
```

2. To determine which packages are available from a running Update Server application, run this command from the SP:

```
inventory get remote-software -i <server_ipaddress> -p <server_port>
```

Note – Some older versions of the SP do not accept the `-i` or `-p` options. These older versions accept only these arguments: `[{-a|--all}]`, `[{-D|--Delim}]`, and `[{-H|--noheader}]`.

3. To compare currently installed packages with packages that are available on a mounted NSV, run this command from the SP:

```
inventory compare versions -f <manifest_filename>
```

4. To compare currently installed packages with packages that are available on a running Update Server, run this command from the SP:

```
inventory compare versions -i <server_ipaddress> -p <server_port>
```

Note – Some older versions of the SP do not accept the `-i` or `-p` options. These older versions accept only these arguments: `[{-a|--all}]`, `[{-D|--Delim}]`, and `[{-H|--noheader}]`.

Updating the SP Base Package

Note – You can use the Update Server application to install this package, or you can use the SP Update Flash option in the operator panel's SP menu.

The SP base component includes the SP Value-Add component, so it also is updated as part of this process.

Note – Because the Value-Add package can contain all feature updates in a new release, check the *Sun Fire V20z and Sun Fire V40z Servers—Release Notes* to determine which package you should update.

1. Log in to the SP.

2. Execute the SP command to start the update process on the SP:

```
sp update flash all -i <server_ipaddress> -p <server_port> -r <version>
```

- The optional `-p` flag indicates that the server is running on a port other than the default port. This command pings the Update Server application to determine whether it is running. If it is successful, your connection is closed when the SP reboots and the update process begins.
- The `-r` flag indicates the version of the remote package that is requested. If LATEST is specified, the latest available version of the package is requested.

Note – Older versions of the SP do not support the `-r` option. If you run the `sp update flash all` command with an old version of the SP, the Update Server application will automatically update your software to the most recent version.

3. Monitor the update process on the server.

Messages are displayed as the installation process begins and ends. (More details of the update processes are in the Update Server log file.) When the update is complete, the SP reboots with the new version.

Note – If you update to a new version of the SP Base or Value-Add package, but do not install the associated documentation in the NSV, the online help will not work. After you mount the file system, check the software inventory for the version of the SP Value-Add package. Ensure that the latest version of the documentation is installed in the `/docs` directory.

Updating the SP Value-Add Package

The SP Value-Add component can contain all the new features in a new release. Check the *Sun Fire V20z and Sun Fire V40z Servers—Release Notes* to determine whether to update the Value-Add package or the SP Base package.

Note – You do *not* have to perform this upgrade if you already updated the SP Base package.

1. Log in to the SP.

2. Run this command:

```
sp update flash applications -i <server_ipaddress> -p <server_port>  
-r <package_version>
```

Note – If you use an NFS mount, execute this command:
sp update flash applications -f <filename>

Updating the BIOS

There are three methods available for updating the BIOS, as shown in the procedures in this section:

- Use the Update Server application.
- Mount the NSV.
- Copy the BIOS image directly.

Using the Update Server Application to Update the BIOS

1. Follow the steps in [“Configuring and Starting the Update Server Application” on page 30](#) to use the Update Server application.
2. At the SP prompt, type this command:

```
platform set os state update-bios -i <server_ipaddress> -p  
<server_port> -r <package_version>
```

Mounting the NSV to Update the BIOS

1. Log in to the SP and mount the NSV.

For example, if the IP address of the machine with the new NSV is 10.10.20.100 and you extracted the NSV files to a directory named newNSV, you would run the command:

```
sp add mount -r 10.10.20.100:/newNSV
```

The NSV will then be available to the SP at /mnt/sw_images/.

2. At the SP prompt, type this command:

```
platform set os state update-bios  
/mnt/sw_images/platform/firmware/bios/Vx.x.x.x/bios.sp
```

Where *Vx.x.x.x* is the version number you want.

Copying the BIOS Image to Update the BIOS

1. Copy the BIOS image directly from the NSV to the `/tmp` folder on the SP file system.

2. At the SP prompt, type this command:

```
platform set os state update-bios /tmp/bios.sp
```

Updating the Diagnostics

The SP-based diagnostics tests are stored in the NSV and are referenced by the `/diags` symbolic link in the SP. The SP software references a default version of the diagnostics. However, if a new version is released and stored on the NSV, you must point to that new version in order to use it.

1. Log in to the SP.

2. Mount the NSV, using the `sp add mount` command. For example:

```
sp add mount -r <network_path>
```

This mounts the directory specified by `network_path` on `/mnt`.

3. To verify that the mount was successful, type `ls /mnt/diags`. For example:

```
ls /mnt/diags V2.4.1.0
```

4. Use the `sp update diags` command to establish a soft link from `/diags` to the desired diagnostics directory. For example:

```
sp update diags -p /mnt/diags/V2.4.1.0
```

5. To verify the new soft link, type `ls -l /diags`. For example.

```
ls -l /diags /diags -> /mnt/diags/V2.4.1.0
```

6. To verify that the diagnostics subsystem is available, type this command:

```
diags
```

The output lists all the subcommands of the `diags` command.

Note – See the *Sun Fire V20z and Sun Fire V40z Servers—User Guide* for a complete list of diagnostics modules and sample output.

Autoconfiguring the SP

You can also use the autoconfiguration option to update the SP. Autoconfiguration replicates the majority of configuration files from an SP that has already been configured to another SP, so that the two servers have identical configurations, except for the host name and IP address.

For example, after you configure a single SP (set up users, hosts, certificates, mounts and so on), you then run autoconfiguration on each additional SP so that the settings are identical. In addition, if you modify the configuration of one SP, you can update all of them by re-running autoconfiguration on each one. (For this reason, set the IP address of the autoconfigure server to x.x.x.1.)

For a list of files that are copied or not copied during the autoconfiguration process, see [“Files Copied During the Autoconfiguration Process” on page 37](#).

Autoconfiguration has these limitations:

- Autoconfiguration does not merge configurations, it overwrites the existing configuration.
- Autoconfiguration does not work across different server platforms. That is, you cannot configure a Sun Fire V40z SP using settings on a Sun Fire V20z SP.
- Autoconfiguration also does not work across different SP software versions. The servers must be running the same version of the SP software.

You can start autoconfiguration either when you are prompted at the completion of setting the IP address of the SP, or by selecting Autoconfigure from the SP menu option on the operator panel at any time.

To perform autoconfiguration of an SP, follow these steps.

- 1. On the operator panel, press the Forward or Back button until the following prompt appears.**

```
SP Autoconfigure?
```

- 2. Press the Select button.**

The following prompt appears:

```
SP Auto Setup?
```

```
No
```

- 3. Press the Forward or Back button to change the prompt to Yes.**

For instructions on setting an IP address, refer to the *Sun Fire V20z and Sun Fire V40z—Server Installation Guide*.

4. Press the Select button.

The SP attempts to locate an IP address.

- If the SP successfully locates an IP address, the following prompt appears, displaying an IP address for this SP:

```
Setup Server IP:  
x.x.x.1
```

Where *x.x.x* is the first three octets of the SP IP address. For example, if the address is 10.10.30.19, the address that displays in the prompt appears as 10.10.30.1.

In this case, press the Select button on the operator panel to start the autoconfiguration.

- If the SP does not locate an IP address, the following message appears:

```
Unable to get  
SP IP address
```

In this case, you must manually enter an IP address before you press the Select button on the operator panel to start the autoconfiguration.

5. Wait until the autoconfiguration is complete, at which point the SP automatically reboots.

The following message is displayed when the autoconfiguration is running:

```
SP AutoConfigure  
in progress
```

Note – If the autoconfiguration is unsuccessful, a failure message displays. Press any button on the operator panel to clear it.

Files Copied During the Autoconfiguration Process

[TABLE 1-5](#) provides the list of files copied during the autoconfiguration process. [TABLE 1-6](#) provides the list of files that are not copied during the process.

The autoconfiguration process performs some safety checks on some files.

- The `passwd` and `shadow` files are processed to pass only user accounts that can be created through the `access` command.
- The root account, enabled or not, is not cloned.
- The `fstab` file passes only the mount point information for `/mnt`.

- Only the SSH key files for those users who are authorized on the system are replicated on the target. Key files for users who have been removed from the target are removed.
- Under SP software version 2.1.*, the files `IPMIConfig.xml` and `SystemStruct.xml` are copied only if the product ID and board revision are identical. These files cannot be transferred from a server running SP software version 2.1.* to a server running SP software version 2.2.*, and vice-versa.

All files are transferred between the two hosts through an SSL socket connection. This is true even if the option `ssl_not_enforced` is enabled.

TABLE 1-5 Files Copied During the Autoconfiguration Process

File	Purpose
<code>/pstore/passwd</code>	User account list
<code>/pstore/group</code>	User group list
<code>/pstore/shadow</code>	User account passwords (local users only)
<code>/pstore/fstab</code>	/mnt file system information
<code>/pstore/smb.creds</code>	User/password information for SMB mount
<code>/pstore/evcfg.xml</code>	Event manager configuration file
<code>/pstore/seccfg.xml</code>	Security manager configuration file
<code>/pstore/oppanelcfg.xml</code>	Operator panel configuration file
<code>/pstore/snmpd.conf.template</code>	SNMP configuration file
<code>/pstore/snmp_proxy_community.txt</code>	SNMP configuration file
<code>/pstore/resolv.conf</code>	Directory Name Service configuration
<code>/pstore/jnet_config</code>	JNET network configuration
<code>/pstore/krb5.keytab</code>	Kerberos authentication configuration (for Windows authentication)
<code>/pstore/ssl_not_enforced</code>	Disables SSL requirement for SM GUI console
<code>/pstore/user_ssl_server.key, .crt</code>	SSL key and certification for SM GUI console
<code>/pstore/ssh_known_hosts</code>	SSH host keys (trusted hosts)
<code>/pstore/ssh_authorized_keys/*</code>	SSH user keys (trusted users)
<code>/pstore/IPMI/IPMIConfig.xml</code>	IPMI configuration
<code>/pstore/IPMI/ipmiusers</code>	IPMI user list <i>(Note: Not copied by servers or clients running SP software version 2.1.)</i>
<code>/pstore/SystemsStruct.xml</code>	User-modified sensor thresholds
<code>/dev/mtd/custom</code>	Custom configuration area

TABLE 1-6 Files Not Copied During the Autoconfiguration Process

File	Purpose
/pstore/mc.conf	Machine check configuration
/pstore/hostname	Local SP hostname
/pstore/ifcfg2-eth0	Local SP IP configuration
/pstore/dimm.map	Platform DIMM configuration
/pstore/edstatefile	Local SP event log
/pstore/emstatefile	Local SP event log
/pstore/hwinventory	Hardware inventory list
/pstore/inv_manifests/*	Software inventory list
/pstore/snmpd.conf	SNMP engine unique ID
/pstore/sp_uuid	SP unique ID
/pstore/ssh/ssh_host*	SSH host keys
/pstore/IPMI/sdr	IPMI sensor data repository
/pstore/IPMI/SEL	IPMI sensor event log

Determining SP and Platform Network MAC Addresses

Use the following commands if you need to determine the MAC address of your server's SP or platform:

```
# ssh spipaddress -l spusername sp get mac
# ssh spipaddress -l spusername platform get mac
```

Systems Management Console Features

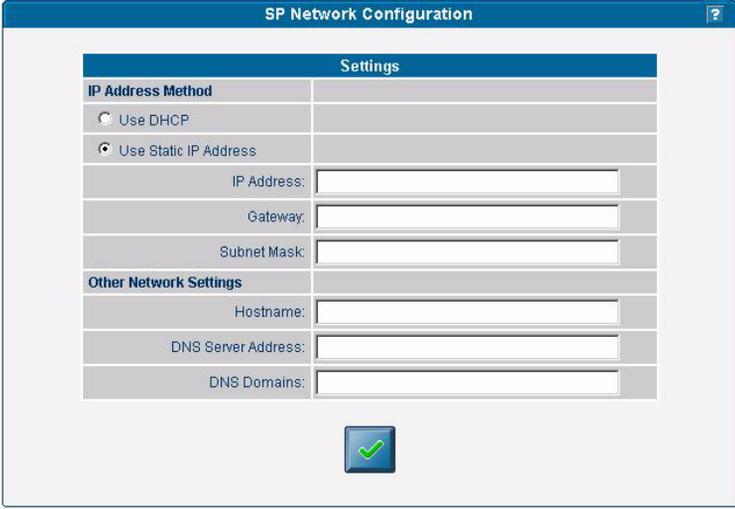
You can manage the servers by using line commands or by using the web-based Systems Management (SM) Console graphical interface. This section gives an overview of the actions you can do with the SM Console interface.

Note – For full information on the line commands, see the appendixes in this guide.

Configuring Network Settings

Admin-level and manager-level users can use the SM Console to configure the SP network settings to define the IP address method (static or DHCP), and other network settings such as the host name, DNS server address, and domains.

Note – As discussed in the previous section, you also can configure network settings from the operator panel, or you can use the `sp ip` commands, which are explained in the online help.



The screenshot shows a window titled "SP Network Configuration" with a "Settings" table. The table is divided into two sections: "IP Address Method" and "Other Network Settings".

Settings	
IP Address Method	
<input type="radio"/> Use DHCP	
<input checked="" type="radio"/> Use Static IP Address	
IP Address:	<input type="text"/>
Gateway:	<input type="text"/>
Subnet Mask:	<input type="text"/>
Other Network Settings	
Hostname:	<input type="text"/>
DNS Server Address:	<input type="text"/>
DNS Domains:	<input type="text"/>

At the bottom center of the window is a blue square button with a white checkmark.

FIGURE 1-11 Network Configuration Screen

To configure the SP network settings from the SM Console:

1. Click **Configuration>SP Network** from the menu bar.
2. In the **Settings** table, select which **IP Address Method (DHCP or Static IP Address)** that you want to use.
3. If you choose **Static IP Address**, type the IP address, gateway address, and subnet mask.
4. **Identify other network settings.**
 - The host name of the SP
 - A single IP address of a DNS server (if available)
 - A space-separated list of search domains (if applicable)
5. Click the **checkmark button** to save the settings.

Note – If you choose DHCP, the SP broadcasts for a DHCP server to obtain a dynamic IP. The IP address information is displayed, but you cannot edit it.

Starting and Stopping the Platform OS

Admin-level and manager-level users can start and stop the platform operating system from the SM Console. Choose Management>Platform Operations from the menu bar, then choose one of the options listed in the table below.

TABLE 1-7 Stop and Start Options for the Platform OS

Option	Description
Power On / Restart	<p>The Power On/Restart option starts the platform operating system.</p> <p>The Boot into BIOS Setup option boots the platform and causes BIOS to enter setup mode. This enables you to modify the BIOS settings from the platform console. After you select this option, you must access the BIOS configuration screen to change BIOS settings. See the <i>Sun Fire V20z and Sun Fire V40z Servers—User Guide</i> for details about configuration of BIOS settings.</p> <p>For remote access, log in via the SM Console. Choose Troubleshooting>SP SSH Console. Then execute the <code>platform console</code> command.</p> <p>The Forced Restart option bypasses the operating system shutdown stage during a system restart. It can cause loss of data.</p>
Shutdown / Power Off	<p>The Shutdown/Power Off option shuts down the platform operating system and powers off the machine.</p> <p>The Forced Power Off option bypasses the operating system shutdown stage. It might cause loss of data. Use the Forced Power Off option if you must force a shutdown.</p>

After you choose an option and click the checkmark button, the operation is initiated on the server. The help text displays any processing and results messages. The current state (normal, warning, or critical) is reflected in the System Status button, so you can monitor the progress.

When you move the cursor over the Platform Operating System button, one of these states is displayed in the help panel:

- Off
- On
- Communicating
- Diagnostics
- Sleeping

- BIOS booting
- BIOS setup
- OS booting
- OS shutting down

Note – You can perform platform state management from the command line with the platform subcommands. See [Appendix F](#) for more information.

If the power currently is off, the Shutdown/Power Off option returns a message that the option was not executed, due to the current state.

Configuring SMTP Event Notification

Admin-level and manager-level users can configure the system to:

- Send email for generated events, via a Simple Mail Transfer Protocol (SMTP) server.
- Route email based on the severity of events.
- Send email that contains a subject and content (long format), or only a subject (short format) to support target devices such as phones, pagers, and so on.

SMTP notification ensures rapid notification about events and rapid response to critical situations. You can use the SM Console or the `sp smtp` commands to configure SMTP event notification.

Follow these steps to configure automatic SMTP email alerts from the SM Console.

1. **Click Configuration>SMTP Event Notification from the menu bar.**
2. **Type the SMTP server name (either the host name or IP address of the SMTP server where you want email to be routed). Use an IP address unless DNS is configured on the SP.**
3. **For each level of severity, type a comma-separated list of email addresses.**

These are the addresses that will receive email for each level of severity. Severity levels are:

- Informational
- Warning
- Critical

FIGURE 1-12 SMTP Event Notification Screen

Note – Type separate address lists for short and for long email formats. Type a comma between each email address. Type a separate list for pager email addresses that require shorter text.

4. Click the checkmark button to save the settings.

Configuring Directory Services

Configure the directory services options to define how SP user name and password information is stored or accessed.

TABLE 1-8 Directory Services Options for the Service Processor

Option	Description
NIS	Network Information Service: A UNIX-originated solution to directory service. Both local files and a remote NIS server authenticate users.
ADS	Active Directory Service: Microsoft's directory service. Both local files and a remote ADS server authenticate users.

You can use the SM Console or the `access` subcommands to configure the directory services options. See [Appendix B](#) for more information.

To configure directory services from the SM Console:

1. Click Access Control>Directory Services from the menu bar.

Select the type of directory service you want to use:

- To use local authentication only, select Use local etc/pw files in the Settings table.
- To use directory services, select Use NIS or Use ADS for the network directory services database you want to use.

Settings	
<input checked="" type="radio"/> Use local etc/pw files	
<input type="radio"/> Use NIS	
NIS Domain:	<input type="text"/>
NIS Server(s):	<input type="text"/>
<input type="radio"/> Use ADS	
ADS Domain:	<input type="text"/>
ADS Server:	<input type="text"/>
Organizational Unit:	<input type="text"/>
ADS Logon ID:	<input type="text"/>
Keytab File	<input type="text"/> <input type="button" value="Browse..."/>

FIGURE 1-13 Directory Services Configuration Screen

2. Type the domain name for the option you selected in [Step n](#).
3. Type the server name for the option you selected in [Step n](#). (For multiple servers, type a comma between each server name.)
4. If you chose ADS, also type the organizational unit, ADS Logon ID, and the location of the keytab file.
See [“Creating Keytab Files for ADS”](#) on page 46.
5. Click the checkmark button to save the settings.

Note – If you use ADS, the clock on the service processor must be synchronized with the clock on the ADS server. Also, the service processor and the ADS server must be able to resolve each other’s host names using DNS.

Remote users who are authenticated via directory services have access to the SP only via a group mapping that maps the user’s remote group to an SP administrative group.

To simplify configuration on the SP, manager-level users can map directory service groups to predefined groups. When you map those users (members of directory services groups) to an SP administrative group, they automatically have appropriate access rights.

Mapping Directory Service Groups

The Directory Service Group Mappings table identifies existing group mappings. This table also provides options for mapping other directory service groups to an SP group. For example, if Directory Services group 5 is mapped to manager, all members of group 5 are granted manager-level privileges on the SP.

To map directory service groups from the SM Console:

1. Click Access Control>DS Group Mappings from the menu bar.

The current group mappings table is displayed. Directory service groups are listed alphabetically.



FIGURE 1-14 Directory Service Group Mappings Screen

2. Choose one of the following:

- Select which mapping you want to create (monitor, admin, or manager).
- Select No Mapping to disable access to the SP.

- Enter a new group in the text field and select one of the corresponding options to indicate the SP group for the new entry.
 - Select the Verify checkbox so that an error will display if a new group name is not located in the directory service.
3. Click the check mark button.

Creating Keytab Files for ADS

To use Active Directory Service (ADS) as a directory service on the SP, you must create an active directory account. The *name service library* on the SP uses this account to authenticate itself to the LDAP interface of the active directory server.

ADS Server Requirements

- The ADS server must have Certificate Services and the High Encryption Pack installed.
- The Microsoft Windows administrator must create an Active Directory account and a keytab (for that account) that the SP(s) will use to conduct LDAP queries. You can create keytab files with the `ktpass` command that is located in the Microsoft Windows 2000 resource kit:

```
ktpass -princ <logon>@<domain> -pass <password> -mapuser <logon> -out  
<output filename>
```

Note – The keytab you create with this command can be uploaded to the SP with the `scp` command, or can be accessed from an exported file system that is mounted by the SP. See your Microsoft documentation for details about this command.

ADS SP Requirements

- You must configure DNS.
 - The canonical name of each host must be the fully-qualified host name (including the domain).
 - The IP address of each host must reverse-resolve to the canonical name.
- The time on the SP must be accurate to within five minutes of the time on the ADS server (domain controller). When the platform is started, the SP clock is synchronized with the platform clock.

- You must configure ADS properly. From the SM Console, type:
 - The ADS domain
 - The ADS server name
 - The organization unit (OU) under which the SP searches for group information
 - The ADS log in ID (the name of the account that was created for the SP to use)
 - The keytab file that was uploaded and installed on the SP

Configuring Date and Time

Admin-level and manager-level users can configure the date and time setting for the SP clock. Use the `sp date` command from the command line or configure date and time from the SM Console.

- The clock is synchronized automatically when platform drivers are installed. If the platform is running (and the drivers are installed) the platform time takes precedence over the SP time.
- The platform time must be set correctly for ADS to function.

If you configure the SP before you load the platform operating system and you want to set the time to synchronize with ADS and other network services, follow the procedure below.

From the SM Console:

1. Click **Configuration>SP Date/Time** from the menu bar.
2. Identify the date and time on the SP clock.

The current SP time displays in `yyyy:mm:dd hh:mm:ss` format. [FIGURE 1-15](#) illustrates an example of this format.

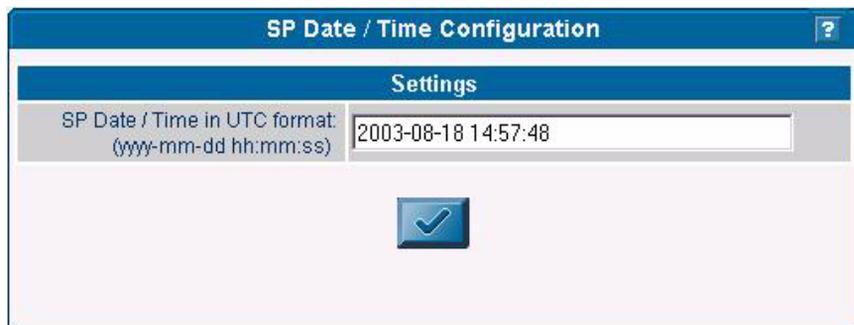


FIGURE 1-15 Date/Time Configuration Screen

3. Click the checkmark button to save the settings.

Configuring SSL

Set up web access to the SP with either an encrypted or a non-encrypted communication method.

By default, all messages between your browser and the SP are encrypted according to Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS). Version 0.9.6j is supported.

1. Allow browsers to communicate with the SP via non-encrypted messages by one of the following two methods:

- From the command line, use this command:

```
sp disable ssl-required
```

- From the SM Console, select Optional (disable) or Required (enable) from the SSL Configuration screen as described in [“Configuring the SSL Certificate From the SM Console” on page 48](#).

With SSL disabled, HTTP requests are serviced directly without any redirection to HTTPS. HTTPS requests continue to be secure.

Note – If you enable the HTTPS protocol, your browser will display a warning message that states that it cannot verify the validity of the Server Certificate. This warning is informational and you can ignore it safely. To upload a certificate that is signed by your own organization or by an independent certificate authority, select User Supplied, as described in [“Configuring the SSL Certificate From the SM Console” on page 48](#).

2. To revert to the default behavior, type this command:

```
sp enable ssl-required
```

With SSL enabled, HTTP requests are redirected automatically to equivalent HTTPS requests to maintain site security.

Configuring the SSL Certificate From the SM Console

Admin-level and manager-level users can enable or disable SSL encryption and can define the SSL certificate that is used to manage transmission security.

Note – You also can use `sp ssl` commands to configure the SSL certificate. See [Appendix H](#) or the SM Console online help for details about commands.

Follow these steps to configure the SSL certificate from the SM Console:

1. Click **Configuration>SSL Certificate** from the menu bar.
2. Select **Required** or **Optional** to designate SSL access.



Settings	
SSL Access	
<input checked="" type="radio"/> Required	
<input type="radio"/> Optional	
SSL Certificate Configuration	
<input checked="" type="radio"/> Factory Certificate	
<input type="radio"/> User-supplied Certificate	
Certificate File:	<input type="text"/> Browse...
Key File:	<input type="text"/> Browse...
	

FIGURE 1-16 SSL Configuration Screen

3. **Do one of the following, depending on whether you selected Required or Optional:**
 - If you selected **Required**, select the option for the type of SSL certificate configuration you want to use, factory-installed or your own in-house certificate management.
 - If you selected **User-supplied Certificate**:
 - a. **Type the name of your generated certificate file to be installed with Apache on the SP, or click the Browse button to search for a file.**
 - b. **Type the name of your generated key file to be installed with Apache on the SP, or click the Browse button to search for a file.**
4. **Click the checkmark button.**

Monitoring System Status

The System Status window displays an image that represents the physical layout and status of all hardware components and sensors. You can use this window to identify components that have problems, or failed components that must be replaced. To access this window, click the System Status button from the toolbar in the SM Console.

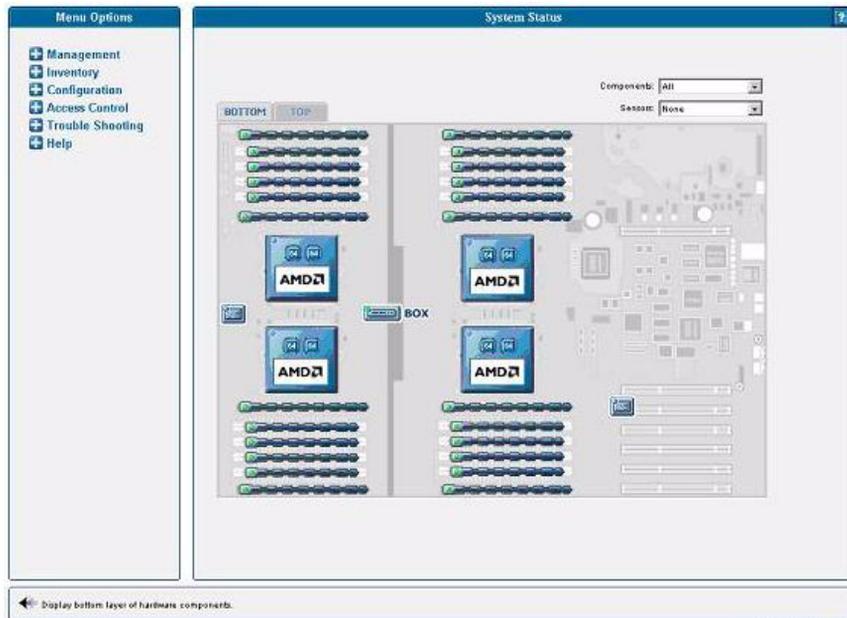


FIGURE 1-17 System Status Window, Bottom Tab, Sun Fire V40z Server

The component images represent the actual physical hardware components including their approximate location, size, and status. The hardware layout for the Sun Fire V40z server is represented in two layers. (FIGURE 1-17 illustrates the default, bottom tab view for the dual-core Sun Fire V40z server.) Click the Bottom and Top tabs at the top of the image to change views.

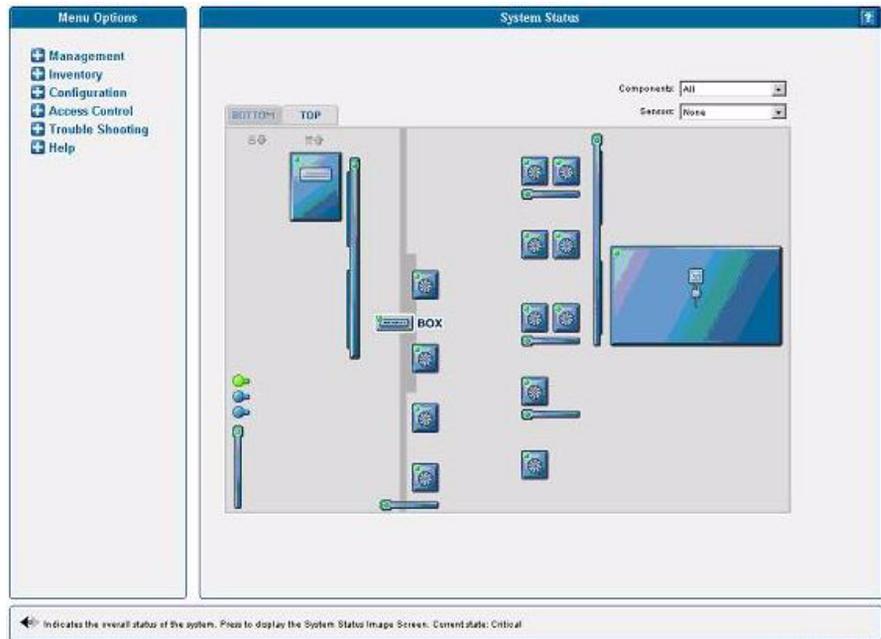


FIGURE 1-18 System Status Window, Top Tab, Sun Fire V40z Server

You also can display component details for troubleshooting purposes. To view details about a component, click on the image of that component. You also can use the pulldown menus in the top right corner of the window to locate specific component types (CD-ROM drives, CPUs, disk drives, fans, and so on) and specific sensor types (fan, power, and temperature sensors).

Sensor images represent the approximate location, current value, and warning or critical thresholds of system sensors. Current information about the sensor (name, type, current value, low and high warning, critical thresholds, and status) is displayed in a gauge component in the top right corner of the image.

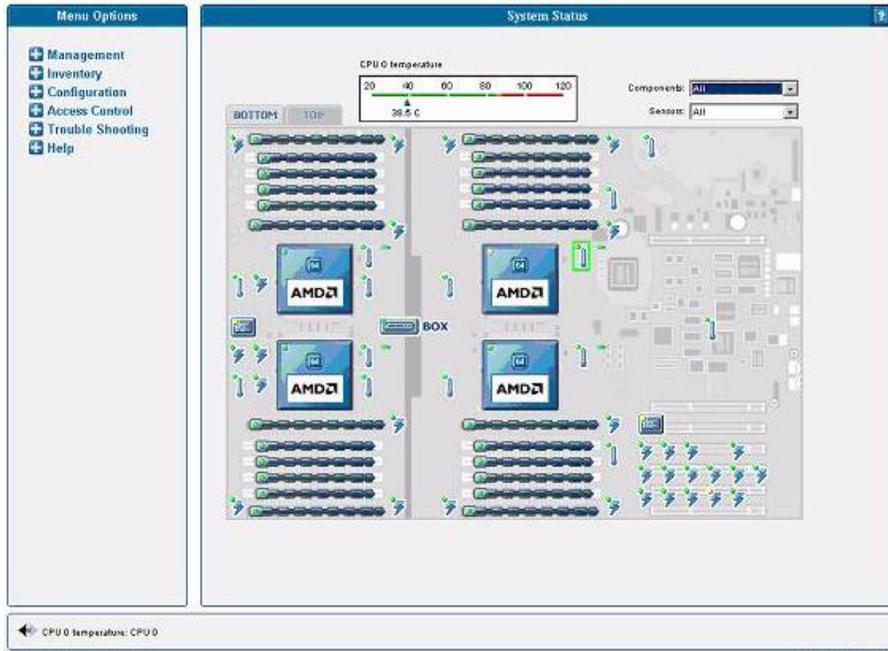


FIGURE 1-19 Temperature Sensors Displayed, Bottom Layer

To view values for a sensor, click on the image of the sensor. In [FIGURE 1-19](#), which illustrates the bottom layer, the temperature sensor gauge for CPU 0 is displayed.

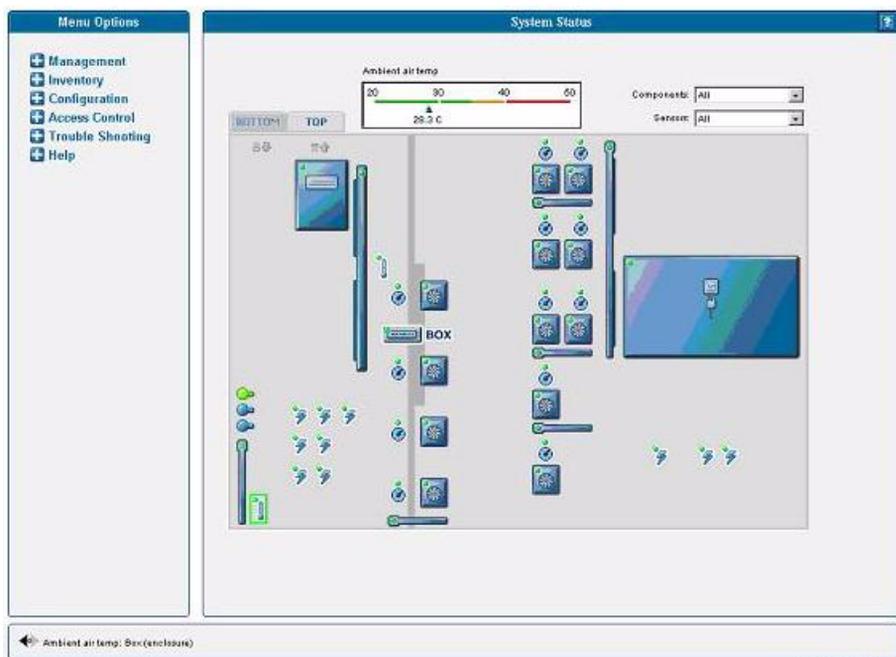


FIGURE 1-20 Ambient Air Temperature Gauge Displayed, Top Layer

In [FIGURE 1-20](#), which illustrates the top layer, the sensor gauge for the ambient air temperature is displayed.

Note – See the SM Console online help for additional instructions and details about management tasks.

System Events

Administrators with appropriate privileges can use the SM Console’s System Events table to view detailed information about all active events. They also can perform various actions that are related to each event.

Each active event is displayed on its own row in the table, as illustrated in [FIGURE 1-21](#).



FIGURE 1-21 System Events Table

[TABLE 1-9](#) describes the columns in the System Events table.

TABLE 1-9 System Events Table

Column	Description
Component	The unique name of the component that has caused this event. Components can be hardware or software.
Detail	Displays details for the component.
ID	Contains a unique event ID for each event enabling you to track the event in your external trouble-ticket system and query the log files for all actions related to the event.
All	Displays the highest severity that this event has ever achieved: informational (green), warning (yellow), and critical (red) icons. Click the icon to view details for the event.
Current	Displays the current severity: informational (green), warning (yellow), and critical (red) icons. Also displays a descriptive message. These descriptions might be lengthy, (for example, describing the nominal, warning, critical, and current temperature for a fan) and therefore are displayed in the help panel.
Type	Identifies the event type. Refer to TABLE 1-10 for a description of the icons representing each type.

TABLE 1-9 System Events Table (*Continued*)

Column	Description
First	Lists the date and time at which the event was initially generated. Only the time is displayed in the table; help text at the bottom of the window displays the entire date and time.
Last	The date and time at which the event was most recently generated. Only the time is displayed in the table; help text at the bottom of the window displays the entire date and time.
Count	Lists the number of times the event has occurred. If a new event has the same component and event type as a current, uncleared event, a new event is not created but the count for the current event is incremented and the current severity is updated in the Current column.
Clear	Clicking this button clears the event. You must manually clear all events. A cleared event is deleted from the server and automatically cleared from any other connected SM Console System Events screen. Only admin-level and manager-level users can clear events.
Help	Clicking this button displays online help for this table.

Note – You also can use the `sp get events` command to obtain information about events. See [Appendix H](#) or the online help for more information.

Icons Used to Identify System Events

The SM Console displays the icons that are illustrated in [TABLE 1-10](#) to represent specific types of events.

TABLE 1-10 System Event Type Icons

Event Type	Icon
BIOS Events	
Fan Speed Events	
Machine Check Events	
Miscellaneous Operating System Events	
Platform State Events	
Switch Events	

TABLE 1-10 System Event Type Icons (*Continued*)

Event Type	Icon
Temperature Events	
Unknown Event	—
Voltage Events	

Note – See the *Sun Fire V20z and Sun Fire V40z Servers—Troubleshooting Techniques and Diagnostics Guide* for a table of system events and troubleshooting suggestions.

IPMI Server Management

Server manufacturers today have to re-invent how each new server manages itself. The hardware and software design for one server does not necessarily work with another. Every server supplier provides basic monitoring and data collection functions but no two do it exactly the same. These proprietary implementations for manageability only complicate the problem.

Intelligent Platform Management Interface

The standardization of server-based management, called Intelligent Platform Management Interface (IPMI), provides a solution. IPMI enables you to interconnect the CPU and devices being managed. It allows for:

- Easy replication of the monitoring functions from server to server
- Support for a reasonably large number of monitoring devices
- Common driver-level access to management instrumentation
- More cost-effective implementations
- Increased scalability of the server management functions

IPMI is an industry-standard, hardware-manageability interface specification that provides an architecture defining how unique devices can all communicate with the CPU in a standard way. It facilitates platform-side server management and remote server-management frameworks, by providing a standard set of interfaces for monitoring and managing servers.

With IPMI, the software becomes less dependent on hardware because the management intelligence resides in the IPMI firmware layer, thereby creating a more intelligently managed server. The IPMI solution increases server scalability by distributing the management intelligence closer to the devices that are being managed.

Baseboard Management Controller

In order to perform autonomous platform-management functions, the processor runs embedded software or firmware. Together, the processor and its controlling firmware are referred to as the Baseboard Management Controller (BMC), which is the core of the IPMI structure. Tightly integrating an IPMI BMC and management software with platform firmware provides a total management solution.

Note – Another way to perform IPMI queries and actions on the BMC is through the IPMI client utility IPMITool, which is used extensively in the testing process. For more information, see [“Lights Out Management” on page 67](#).

The BMC is a service processor integrated into the motherboard design, providing a management solution independent of the main processor. The monitored server can communicate with the BMC through one of three defined interfaces, which are based on a set of registers shared between the platform and the BMC.

Note – In the Sun Fire V20z and Sun Fire V40z servers, the SP has software that emulates a BMC.

The BMC is responsible for:

- Managing the interface between server management software and platform management hardware
- Interfacing to the system sensors, such as fan speed and voltage monitors
- Providing access to the system event log
- Providing autonomous monitoring, event logging, and recovery control
- Acting as a gateway between the management software and the IPMB/ICMB
- Monitoring the system watchdog timer
- Facilitating the remote-management tasks, even when the main server hardware is in an inoperable state

The BMC provides the intelligence behind IPMI. In the Sun Fire V20z and Sun Fire V40z servers, the SP serves as the BMC, providing access to sensor data and events through the standard IPMI interfaces.

Manageability

IPMI defines a mechanism for server monitoring and recovery implemented directly in hardware and firmware. IPMI functions are available independent of the main processors, BIOS, and operating system.

IPMI monitoring, logging, and access functions add a built-in level of manageability to the platform hardware. IPMI can be used in conjunction with server-management software running under the OS, which provides an enhanced level of manageability.

IPMI provides the foundation for smarter management of servers by providing a methodology for maintaining and improving the reliability, availability, and serviceability of expensive server hardware.

Functional Overview

The following list details the main features of IPMI in the servers:

- A fully functional Sensor Data Record Repository (SDRR) is the container for and interface from which you can access sensor data records (SDRs).

The BMC owns all sensors within the repository. The SDRR features include:

- A single management controller record
- Threshold-based analog sensors for temperature, voltage, and fans, as well as power
- Device locator records for FRUs, to which the physical sensor records are linked (through entity ID/instance relationships)
- Various discrete sensors and event-only sensors
- The system event log (SEL) is a 16K maximum persistent file. For more information, refer to [“System Event Log” on page 61](#).
- The watchdog timer (WDT) supports all timer uses, no pre-timeout interrupt actions, and all (reset, power down, power cycle) timeout actions. For more information, refer to [“Watchdog Timers” on page 66](#).
- A field-replaceable unit (FRU) is read-only. It is tightly integrated with the SP’s inventory-management functionality. For more information on the `inventory` commands, see [Appendix D](#). Vital product data (VPD) that is available through the `inventory` command is also available from the FRU.
- The following chassis-control actions are available:
 - Power down
 - Power up
 - Power cycle
 - Hard reset
 - Soft shutdown

- Event filtering and unacknowledged Platform Event Trap (PET) alerts are supported. For more information, refer to [“Event Filters” on page 66](#).
- Both SMS and LAN channels are supported. Refer to [“IPMI Compliance and LAN Channel Access” on page 60](#).
- Serial-over-LAN (SOL) provides serial-port redirection over the LAN channel. Refer to [“Serial-Over-LAN” on page 106](#).

IPMI Compliance and LAN Channel Access

The servers support IPMI with both SMS and LAN channels through the SP software version 2.2 and later. These servers meet compliance standards for IPMI version 2.

The SMS is implemented as a Keyboard Controller Style (KCS) interface.

The IPMI implementation on these servers also supports LAN channel access. (Refer to the IPMI specification v2 for details.) By default, the LAN channel access is disabled. To enable it, use the `ipmi enable channel` command and specify the ID of the channel to enable for the LAN Interface, as follows.

Note – This ID is case-sensitive and must be lowercase.

```
# ssh spipaddr -l spuser ipmi enable channel {sms | lan}
```

As part of this command, you also specify the password for the default null user. The null user can then use IPMI over the LAN interface. For more information, see [“User Names and Passwords” on page 60](#).

For more information about enabling or disabling the IPMI channel, refer to [Appendix E](#).

User Names and Passwords

Operator-level and admin-level access over the LAN channel requires a valid user name and password. These servers are not preconfigured with user accounts enabled. When you initially enable the LAN channel through the command `ipmi enable channel`, you are required to provide the password for the null user. See [“IPMI Compliance and LAN Channel Access” on page 60](#).

Note – For security reasons, the LAN channel access is disabled by default.

Note – IPMI user identities are in no way associated with user accounts defined for server-management capabilities. Refer to [“Initial Setup of the SP” on page 12](#) for more information about these server-management user accounts.

Server Boot-Option Support

IPMI enables you to set a number of boot options for interpretation by the BIOS. [TABLE 2-1](#) describes important information about the server boot options and parameters that the BIOS supports.

TABLE 2-1 Server Boot Options Supported by the BIOS

Parameter	Number	Details
Set In Progress	0	This parameter is fully supported except for the rollback functionality.
BMC boot flag valid bit clearing	3	Fully supported.
Boot info ack	4	BIOS supports indicating that it has handled boot information.
Boot Flags	5	<ul style="list-style-type: none">• Data byte 1 is supported for the boot flags valid bit.• Data byte 2 (CMOS Clear) is supported; however, when this bit is set, all other bits in this byte are ignored.• Lock keyboard is fully supported.• Boot device selector is supported except for booting to BIOS Setup.• Data byte 3 is supported for user password bypass

System Event Log

The IPMI system event log (SEL) is part of the BMC. Several types of information are logged to the SEL, from administrative messages to indications of important events, such as sensor-threshold crossings.

The size of the log is 16K, which allows for 1024 records.

Sensors

Sensors generate events, obtain readings, and set thresholds. The Sensor Data Record Repository (SDRR) contains several types of sensors.

You access all sensors through the BMC. Many sensors represent physical sensors that are distributed on the motherboard and contained within FRUs. These sensors are polled. When they cross a threshold, an entry is entered in the SEL.

For more information on sensor commands, see [Appendix G](#).

Determine Sensor Presence

To determine the presence of a sensor, run the subcommand `sensor get`.

A sensor that is offline (not reporting) or physically not present in the system is indicated by state `unavailable` in the command response data.

Sensor Thresholds

To retrieve sensor thresholds, run the subcommand `sensor get`.

To set sensor thresholds, run subcommand `sensor set`.

If you specify no thresholds, the result is no change and the return code is `success`.

[TABLE 2-2](#) lists the completion codes that are returned by the subcommand `set sensor`.

TABLE 2-2 Completion Codes for Sensor Thresholds

Code	Cause
0x00 (success)	Sensor thresholds set as requested.
0xCD (illegal command)	Sensor thresholds are unchangeable.
0xCC (invalid request)	Attempting to set an unsettable threshold or attempting to set thresholds in an improper order (for example, the upper critical threshold is set lower than the upper noncritical threshold).
0xC0 (node busy)	Processing resources are temporarily unavailable.

Temperature Sensors

Temperature sensor readings are defined within a range of 0° C to 150° C, a difference of 151° C. The CPU die temperature thermal trip occurs at approximately 140° C.

Temperature sensors can generate the following SEL events:

- Upper Critical Moving Higher Assertion
- Upper Critical Moving Higher De-assertion
- Upper Non-critical Moving Higher Assertion
- Upper Non-critical Moving Higher De-assertion

Memory Sensors for DIMMs

Each DIMM has its own record, which is used only to log IPMI events.

For more information, refer to the section “Analyzing Events” in the *Sun Fire V20z and Sun Fire V40z Servers—Troubleshooting Techniques and Diagnostics Guide*.

Voltage Sensors

All voltage sensor readings are indicated in volts (V). The largest voltage swing that is measured is 15V (the bulk voltage sensor ranges from 0V to 15V). Many of the voltage sensors have much lower maximums and smaller ranges. Voltage sensors can generate the following SEL events:

- Upper Critical Moving Higher Assertion
- Upper Critical Moving Higher De-assertion
- Lower Critical Moving Lower Assertion
- Lower Critical Moving Lower De-assertion

Fan Sensors

The values reported for all fan-speed sensor readings are indicated in revolutions per minute (RPMs). The sensors have an upper bound of 15,000 RPM.

Fan sensors can generate the following SEL events:

- Lower Critical Moving Lower Assertion
- Lower Critical Moving Lower De-assertion

Power-Supply Sensors

All power sensor readings are indicated in watts (W) and are defined within a range of 0W to 600W.

- Power sensors do not generate SEL events.
- There are no thresholds for power sensors.

Management Controllers

One management-controller sensor represents the BMC. The management controller has the following capabilities:

- **Global Initialization**—The `init` agent enables the controller to generate messages.
- **Device Capabilities**—This device acts as all of the following:
 - Chassis device
 - IPMB event receiver
 - FRU inventory device
 - SEL device
 - SDRR device
 - Sensor device

Miscellaneous Sensors

The following additional sensors also are supported:

- System event
- Event logging disabled
- System firmware progress
- Watchdog

System Event

The system-event sensor indicates a variety of system events. However, no event conditions are reflected from the subcommand `sensor get`.

PEF actions—Pending actions matched against a platform event filter (PEF) are logged if the event sensor has been configured to do so. Only assertions of pending PEF action conditions are logged.

Sensor Type Code: 0x12 [System Event]
Sensor Specific Offset: 0x04 [PEF Action]

Time sync—Time-sync events occur in pairs: one before and one after a SEL time sync.

Sensor Type Code: 0x12 [System Event]
Sensor Specific Offset: 0x05 [Time sync]

Event Logging Disabled

The sensor event logging disabled indicates certain SEL-related events. This sensor is represented as a “type 2” SDR record.

SEL Full—When the SEL reaches the “maximum-1” number of records, a record is logged and any subsequent add SEL commands return a limit-exceeded code. This record becomes the last record in the SEL when the log is filled to capacity.

Sensor Type Code: 0x10
Sensor Specific Offset: 0x04 [Log Full]

SEL Clear—A record is written to the SEL whenever the command Clear SEL is executed. This occurs only on the command Clear SEL; it does not occur if you delete the last SEL entry with the command Delete SEL Entry.

Sensor Type Code: 0x10
Sensor Specific Offset: 0x02 [Log AreaReset/Cleared]

System Firmware Progress

The system-firmware progress sensor is an event-only sensor. The BIOS Boot Success SEL entry can be logged against this sensor when the BIOS has successfully booted and has attempted to return control to the OS, or if the BIOS has been booted and you enter a BIOS Setup screen.

Sensor Type Code: 0x0F
Sensor Specific Offset: 0x02 [Firmware Progress]
Event Data 2: 0x13 [Starting operating system boot process]

Watchdog

The Watchdog 2 sensor is used to log watchdog timer expirations. These events are generated only for timers that do not have the “do not log” bit set. A timer-expiration event is logged when a watchdog timer expires.

Sensor Type Code: 0x23
Sensor Specific Offset: * all supported actions

Event Filters

Note – To ensure a graceful shutdown, the correct platform drivers must be installed on the server.

Platform Event Filtering (PEF) provides policy management that enables the BMC to act on particular events. The supported actions through PEF include:

- Power down
- Power cycle
- Reset
- Send Alert

TABLE 2-3 lists the event filters that are enabled by default.

TABLE 2-3 Event Filters Enabled by Default

Filter Match	Action
ambienttemp asserts upper critical threshold	Power down
cpu0.dietemp asserts upper critical threshold	Graceful power down
cpu1.dietemp asserts upper critical threshold	Graceful power down
cpu2.dietemp asserts upper critical threshold Note: This filter is ignored on systems with two CPUs.	Graceful power down
cpu3.dietemp asserts upper critical threshold Note: This filter is ignored on systems with two CPUs.	Graceful power down

Watchdog Timers

A watchdog timer allows a selected action to occur when the timer expires.

For timer actions, pre-timeout interrupts are currently not supported. The following actions are supported:

- System reset
- System power off
- System power cycle

Alerting

When you use platform event trap (PET) LAN alerts, the number of alert destinations is limited to 16 (1 nonvolatile, 15 volatile). The number of alert policies is limited to 32.

Note – Acknowledgement of PET LAN alerts and alert strings are unsupported.

Alert Policy Set Determination

When event filters are matched, the following occurs:

- All non-alert actions are scanned for the filters.
- The highest priority action associated with all filters is taken.
- All alert actions are scanned for the filters.
- The highest priority (based on lowest numeric policy number) alert policy set is chosen.

You can configure policies so that, if the previous alert was successful, an alert is not sent as a result of the execution of the alert policy.

Lights Out Management

On these servers, Lights Out Management (LOM) is performed through IPMItool, a utility for controlling IPMI-enabled devices.

Description

IPMItool is a simple command-line interface (CLI) to servers that supports the Intelligent Platform Management Interface (IPMI) v1.5 specification. It provides the ability to:

- Read the sensor data record (SDR) and print sensor values
- Display the contents of the system event log (SEL)
- Print information about field-replaceable units (FRUs)
- Read and set LAN configuration parameters
- Perform chassis power control

Originally written to take advantage of IPMI-over-LAN interfaces, IPMItool is also capable of using a system interface, as provided by a kernel device driver such as OpenIPMI.

Further Information

- For up-to-date information about IPMItool, visit:
<http://ipmitool.sourceforge.net/>
- For more information about the IPMI specification, visit:
<http://www.intel.com/design/servers/ipmi/spec.htm>
- For more information about the OpenIPMI project (MontaVista IPMI kernel driver), visit:
<http://openipmi.sourceforge.net/>

Syntax

The syntax used by IPMItool is as follows:

```
ipmitool [-ghcvV] -I lan -H address [-P password] expression
```

```
ipmitool [-ghcvV] -I open expression
```

IPMItool Options

TABLE 2-4 lists the options available for IPMItool.

TABLE 2-4 Options for IPMItool

Option	Description
-h	Provides help on basic usage from the command line.
-c	Makes the output suitable for parsing, where possible, by separating fields with commas instead of spaces.
-g	Attempts to make IPMI-over-LAN communication more robust.
-V	Displays the version information.
-v	Increases the amount of text output. This option can be specified more than once to increase the level of debug output. If given three times, you receive hexdumps of all incoming and outgoing packets.
-I <i>interface</i>	Selects the IPMI interface to use. The possible interfaces are LAN or open interface.
-H <i>address</i>	Displays the address of the remote server; it can be an IP address or host name. This option is required for the LAN interface connection.
-P <i>password</i>	Displays the password for the remote server. The password is limited to a maximum of 16 characters. The password is optional for the LAN interface; if a password is not provided, the session is not authenticated.

IPMItool Expressions

TABLE 2-5 lists the expressions and parameters available for IPMItool.

Note – For each of these expressions, the beginning command is always `ipmitool`, followed by the expression and parameter(s).

Note – The `sol` command is not supported in these servers, but you can enable a serial-over-LAN feature. See [“Serial-Over-LAN” on page 106](#).

TABLE 2-5 Expressions and Parameters for IPMItool

Expression	Parameter	Subparameter	Description and examples
help			<p>Can be used to get command-line help on IPMItool commands. You can also place this expression at the end of commands to get help on the use of options.</p> <p>EXAMPLES:</p> <pre>ipmitool -I open help Commands: chassis, fru, lan, sdr, sel</pre> <pre>ipmitool -I open chassis help Chassis Commands: status, power, identify, policy, restart_cause</pre> <pre>ipmitool -I open chassis power help Chassis Power Commands: status, on, off, cycle, reset, diag, soft</pre>
raw	netfn	cmd data	<p>Enables you to execute raw IPMI commands (for example, to query the POH counter with a raw command).</p> <p>EXAMPLE:</p> <pre>ipmitool -I open raw 0x0 0x1</pre> <pre>RAW REQ (netfn=0x0 cmd=0x1 data_len=0) RAW RSP (3 bytes) 60 00 00</pre>

TABLE 2-5 Expressions and Parameters for IPMItool (*Continued*)

Expression	Parameter	Subparameter	Description and examples
chaninfo	<i>channel</i>		<p>Displays information about the selected channel. If no channel is specified, the command displays information about the channel currently being used.</p> <p>EXAMPLES:</p> <pre>ipmitool -I open chaninfo Channel 0xf info: Channel Medium Type: System Interface Channel Protocol Type: KCS Session Support: session-less Active Session Count: 0 Protocol Vendor ID: 7154</pre> <pre>ipmitool -I open chaninfo 7 Channel 0x7 info: Channel Medium Type: 802.3 LAN Channel Protocol Type: IPMB-1.0 Session Support: multi-session Active Session Count: 1 Protocol Vendor ID: 7154 Alerting: enabled Per-message Auth: enabled User Level Auth: enabled Access Mode: always available</pre>
userinfo	<i>channel</i>		<p>Displays information about configured user information on a specific LAN channel.</p> <p>EXAMPLE:</p> <pre>ipmitool -I open userinfo 6 Maximum User IDs : 4 Enabled User IDs : 1 Fixed Name User IDs : 1 Access Available : call-in / callback Link Authentication : disabled IPMI Messaging : enabled</pre>
	<i>Note:</i> Channels 6 and 7 are not supported on Sun Fire V20z servers.		

TABLE 2-5 Expressions and Parameters for IPMItool (*Continued*)

Expression	Parameter	Subparameter	Description and examples
chassis	<i>status</i>		Returns information about the high-level status of the server chassis and main power subsystem.
	<i>identify</i>	<i>interval</i>	Controls the front panel identification light. The default value is 15 seconds. Enter "0" to turn the light off.
	<i>restart_cause</i>		Queries the chassis for the cause of the last server restart.
power			Performs a chassis control command to view and change the power state.
	<i>status</i>		Shows the current status of the chassis power.
	<i>on</i>		Powers on the chassis.
	<i>off</i>		Powers off chassis into the <i>soft off</i> state (S4/S5 state). Note: This command does not initiate a clean shutdown of the operating system prior to powering off the server.
	<i>cycle</i>		Provides a power-off interval of at least 1 second. No action should occur if chassis power is in S4/S5 state, but it is recommended to check the power state first and then only issue a power-cycle command if the server power is on or in a lower sleep state than S4/S5.
	<i>reset</i>		Performs a hard reset.

TABLE 2-5 Expressions and Parameters for IPMItool (*Continued*)

Expression	Parameter	Subparameter	Description and examples	
lan	<i>print</i>	<i>channel</i>	Prints the current configuration for the given channel.	
	<i>set</i>	<i>channel</i> <i>parameter</i>	Sets the given parameter on the given channel.	
		<i>ipaddr</i> <i>x.x.x.x</i>	Sets the IP address for this channel.	
		<i>netmask</i> <i>x.x.x.x</i>	Sets the netmask for this channel.	
		<i>macaddr</i> <i>xx:xx:xx:xx:xx:xx</i>	Sets the MAC address for this channel.	
		<i>defgw ipaddr</i> <i>x.x.x.x</i>	Sets the default gateway IP address.	
		<i>defgw macaddr</i> <i>xx:xx:xx:xx:xx:xx</i>	Sets the default gateway MAC address.	
		<i>bakgw ipaddr</i> <i>x.x.x.x</i>	Sets the backup gateway IP address.	
		<i>bakgw macaddr</i> <i>xx:xx:xx:xx:xx:xx</i>	Sets the backup gateway MAC address.	
		<i>password</i> <i>pass</i>	Sets the null user password.	
		<i>user</i>	Enables the user-access mode.	
		<i>access</i> [on off]	Sets the LAN-channel-access mode.	
		<i>ipsrc</i> <i>source</i>	Sets the IP address source. For <i>source</i> , you can indicate: none = unspecified static = manually configured static IP address dhcp = address obtained by BMC running DHCP bios = address loaded by BIOS or system software	
		<i>arp respond</i> [on off]	Sets the BMC-generated ARP responses.	
		<i>arp generate</i> [on off]	Sets the BMC-generated gratuitous ARPs.	
		<i>arp interval</i> [seconds] <i>s</i>	Sets the interval for the BMC-generated gratuitous ARPs.	
		<i>auth level,...</i> <i>type,...</i>	Sets the valid authtypes for a given auth level. Levels can be: <i>callback, user, operator, admin</i> Types can be: <i>none, md2, md5</i>	
	fru	<i>print</i>		Reads all inventory data for the customer-replaceable units (CRUs) and extracts such information as serial number, part number, asset tags, and short strings describing the chassis, board, or product.

TABLE 2-5 Expressions and Parameters for IPMItool (*Continued*)

Expression	Parameter	Subparameter	Description and examples
sdr	list		Reads the Sensor Data Record (SDR) and extracts sensor information, then queries each sensor and prints its name, reading and status.
sel	info		Queries the BMC for information about the system event log (SEL) and its contents.
	clear		Clears the contents of the SEL. The <code>clear</code> command cannot be undone.
	list		Lists the contents of the SEL.

IPMI Linux Kernel Device Driver

The IPMItool application utilizes a modified MontaVista OpenIPMI kernel device driver that is provided on the Sun Fire V20z and Sun Fire V40z Servers Documentation and Support Files CD. The driver has been modified to use an alternate base hardware address and modified device I/O registration.

This driver must be compiled and installed from the Documentation and Support Files CD.

The following kernel modules must be loaded in order for IPMItool to work:

1. `ipmi_msghandler`

The message handler for incoming and outgoing messages for the IPMI interfaces.

2. `ipmi_kcs_drv`

An IPMI Keyboard Controller Style (KCS) interface driver for the message handler.

3. `ipmi_devintf`

Linux character device interface for the message handler.

To force IPMItool to use the device interface, you can specify it on the command line:

```
# ipmitool -I open [option...]
```

To install and compile this kernel device driver, see [“Initial Setup of the SP” on page 12](#).

LAN Interface for the BMC

Note – In the Sun Fire V20z and Sun Fire V40z servers, the SP has software that emulates a BMC.

The IPMItool LAN interface communicates with the BMC over an Ethernet LAN connection using User Datagram Protocol (UDP) under IPv4. UDP datagrams are formatted to contain IPMI request/response messages with IPMI session headers and Remote Management Control Protocol (RMCP) headers.

Remote Management Control Protocol is a request-response protocol delivered using UDP datagrams to port 623. IPMI-over-LAN uses version 1 of the RMCP to support management both before installing the OS on the server, or if the server will not have an OS installed.

The LAN interface is an authenticated, multisession connection; messages delivered to the BMC can (and should) be authenticated with a challenge/response protocol with either a straight password/key or an MD5 message-digest algorithm. IPMItool attempts to connect with administrator privilege level as this is required to perform chassis power functions.

With the `-I` option, you can direct IPMItool to use the LAN interface:

```
# ipmitool -I lan [option...] address password
```

To use the LAN interface with IPMItool, you must provide a host name on the command line.

The password field is optional. If you do not provide a password on the command line, IPMItool attempts to connect without authentication. If you specify a password, it uses MD5 authentication, if supported by the BMC; otherwise, it will use straight password/key.

Files

The file `/dev/ipmi0` is a character-device file used by the OpenIPMI kernel driver.

Examples

If you want to remotely control the power of an IPMI-over-LAN-enabled server, you can use the following commands:

```
# ipmitool -I lan -H spipaddr -P sppasswd chassis power on
```

The result returned is:

```
Chassis Power Control: Up/On
```

```
# ipmitool -I lan -H spipaddr -P sppasswd chassis power status
```

The result returned is:

```
Chassis Power is on
```

Viewing the IPMI System Event Log

To view the system event log (SEL), use IPMITool.

The out-of-band command is:

```
# ipmitool -I lan -H spipaddr -P ipmipasswd sel list
```

The in-band command (using OpenIPMI on a Linux software-based server or LIPMI on a Solaris software-based server) is:

```
# ipmitool -I open sel list
```

Note – To receive more verbose logging messages, you can run the following command:

```
# ssh -l spuser spipaddr sp get events
```

Clearing the IPMI System Event Log

You can use commands to clear the contents of the IPMI SEL.

Use one of the following commands, depending on your OS:

- For Linux: **ipmitool -I open sel clear**
- For Solaris 9: **ipmitool -I lipmi sel clear**
- For Solaris 10: **ipmitool -I bmc sel clear**

IPMI Troubleshooting

TABLE 2-6 describes some potential issues with IPMI and provides solutions.

TABLE 2-6 IPMI Troubleshooting

Issue	Solution
You cannot connect to the management controller using IPMITool over LAN.	Verify the network connection to the management controller and its IP address and verify the channel is enabled using the <code>ipmi get channels</code> command.
You cannot authenticate to the management controller using IPMITool over LAN.	Ensure that you are using the password assigned when you enabled IPMI LAN access from the management-controller shell prompt.
You have forgotten the password for IPMI access over LAN.	<ol style="list-style-type: none">1. You can reset the IPMI setting, reset the SDRR and purge the SEL from the management-controller shell by running the command: <pre># ssh spipaddr -l spuser ipmi reset -a</pre>2. Now re-enable IPMI on LAN with the following commands: <pre># ssh spipaddr -l spuser # ipmi enable channel lan # exit</pre>
IPMITool fails when using the “open” interface.	Ensure that the Linux kernel module <code>ipmi_kcs_drv</code> is loaded by running the <code>lsmod</code> command.

SNMP Server Management

You can manage your server using the Simple Network Management Protocol (SNMP).

Simple Network Management Protocol

Simple Network Management Protocol (SNMP) is a network-management protocol used almost exclusively in TCP/IP networks. SNMP provides a means to monitor and control network devices, and to manage configurations, statistics collection, performance, and security on a network.

SNMP-based management allows for third-party solutions to be used. This includes products such as HP OpenView and CA Unicenter.

The base component of an SNMP solution is the Management Information Base (MIB). The MIB is included on the Sun Fire V20z and Sun Fire V40z Servers Network Share Volume CD.

This server management configuration is beneficial when, for example, you have a cluster of machines serving web content and the platform is connected to the Internet, but the SP is protected and accessible only on an internal network.

SNMP Integration

SNMP is an open network-management technology that enables the management of networks and entities connected to the network. Within the SNMP architecture is a collection of network-management stations and managed nodes.

Network-management stations execute management applications, which monitor and control managed nodes. Managed nodes are devices such as hosts, gateways and so on, which have management agents responsible for performing the management functions requested by the management stations.

SNMP is used to communicate management information between the management stations and the agents. In other words, SNMP is the protocol by which the agent and the management station communicate.

Monitoring status through SNMP at any significant level of detail is accomplished primarily by polling for appropriate information on the part of the management station. Managed nodes can also provide unsolicited status information to management stations in the form of traps, which are likely to guide the polling at the management station.

Communication of information between management entities in a network is accomplished through the exchange of SNMP messages, both in the form of queries (get/set) by the management station and in the form of unsolicited messages (traps) indicated by the agent.

The servers include SNMP agents that allow for out-of-band health and status monitoring. The SNMP agent runs on the SP and therefore all SNMP-based management of the server should occur through the SP.

The SNMP agent on these servers provides the following capabilities:

- Event management
- Inventory management
- Sensor and system state monitoring
- SP configuration monitoring

SNMP Management Information Base

The Management Information Base (MIB) is a text file that describes SNMP data as managed objects. These servers provide SNMP MIBs so that you can manage and monitor your server using any SNMP-capable network management system, such as HP OpenView Network Node Manager (NNM), Tivoli, CA Unicenter, IBM Director, and so on. The MIB data describes the information being managed, reflects current and recent server status, and provides server statistics.

Sun Fire V20z and Sun Fire V40z Servers MIB Tree

FIGURE 3-1 illustrates the MIB tree.

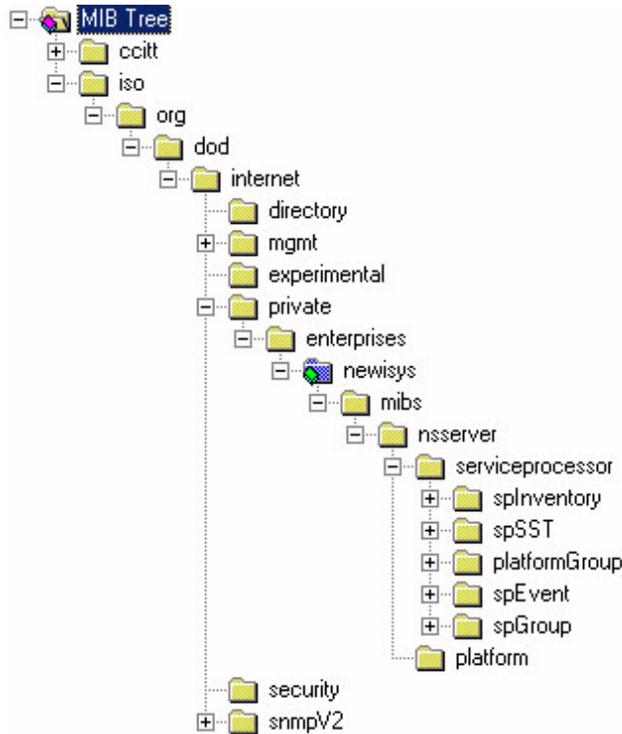


FIGURE 3-1 MIB Tree

Integrating MIBs With Third-Party Consoles

You use the server's MIBs to integrate the management and monitoring of the server into SNMP management consoles. The MIB branch is a private enterprise MIB, located at object identifier (OID) 1.3.6.1.2.1.9237. The standard SNMP port 161 is used by the SNMP agent on the SP.

Configuring SNMP on the Servers

Note – There are several services that are supplied by the SNMP agent on the servers. Depending on your business needs and the configuration of your current office network and management environment, you might want to take advantage of these services.

There are certain prerequisites and setup requirements on both the SP and the platform in order to enable and utilize each of these services:

- SNMP agent on the SP
- Proxy forwarder application/ProxyAgent [RFC 2271]
- Agent X [RFC 2741]

Customers can elect to manage a server out-of-band (OOB) through the SP. With OOB management, the SP is the target of the SNMP request. The SNMP agent on the SP is configured to provide proxy-request capability so that OID requests that are not related to the SP are forwarded, transparently, to the platform OS.

Out-of-Band Management Configuration

FIGURE 3-2 illustrates the SNMP architecture and communication paths between the SP and the platform.

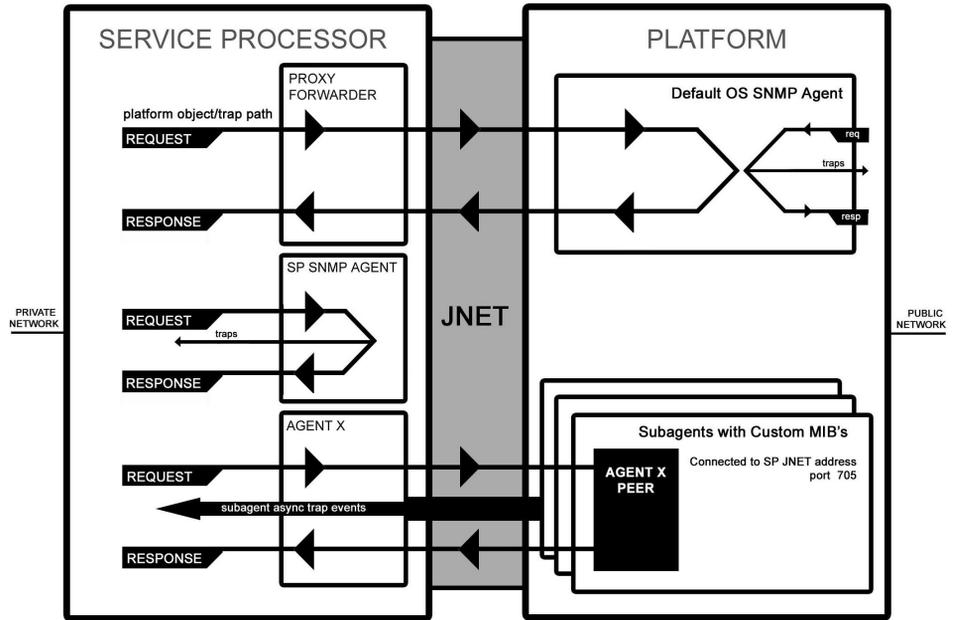


FIGURE 3-2 SNMP Architecture and Communication Paths

SNMP Agent on the Service Processor

The SNMP agent running on the SP facilitates the management and monitoring of the server. The SNMP agent can be used to query various types of SP information. Refer to FIGURE 3-1 for a list of the MIBs; refer to TABLE 3-3 for a detailed description of the MIBs.

There is no configuration required to use this functionality other than integrating the server MIBs with your desired management station.

Refer to the procedure for using the SNMP agent on the SP, as explained in “Integrating MIBs With Third-Party Consoles” on page 80.

Note – The SNMP agent on the Sun Fire V20z and Sun Fire V40z servers supports SNMP v1 and v2c. For security reasons, there are no settable attributes in this agent.

Proxy Agent

The SP acts as an SNMP proxy-agent intermediary for the platform. Queries made from a management station to the SNMP agent on the SP are intercepted by the proxy agent on the SP and forwarded to the platform; the SP proxy agent contacts the platform to retrieve the requested information. The proxy agent then receives the data from the platform and sends the request back to the management station. The management station never knows that the request was proxied. The SP and the platform communicate over an internal private network.

To enable this facility, you must first run an SNMP agent on your platform operating system (see your OS vendor to obtain this agent). This enables platform-level management transparently through the SP. Querying MIBs other than the server MIB (for example, the Host Resource MIB) and the MIBII System MIB on the SP obtains information from the platform by proxying the request to the platform SNMP agent.

Ensure that the SP can identify the read-only and read-write community names that are configured for your platform SNMP agent. Refer to [“Setting the Community Name” on page 82](#).

Setting the Community Name

The SNMP agent on the SP acts as a proxy for the SNMP agent running on the platform. (Refer to [“Configuring SNMP on the Servers” on page 80](#).) To proxy properly, you must use the community string. The community string needed to do so is the value defined when you configured the platform for SNMP.

If you find that your SNMP queries are not being proxied to the platform SNMP agents, validate that the community string on the SP matches the community string on the platform. The SP proxy community string can be changed to match the platform community string using the following command:

```
# sp set snmp proxy community
```

There are no restrictions on the length of the community strings; common names are *private* and *public*. The default name is *public*.

For more information, refer to [“SP Set SNMP Proxy Community Subcommand” on page 250](#).

Agent X

A subagent using SNMP Agent X protocol on the platform can connect to the SNMP agent on the SP (through a special port) and forward query responses or unsolicited traps through the SP. This allows server-management traffic to be kept secure from the production network connected to the platform, if required.

To properly enable this facility, you must identify the IP address and port number pair associated with the SP (as seen from the platform). The Agent X port is fixed at 705 (TCP). However, the private-network IP address is configurable and, by default, this address is 169.254.101.2.

Refer to your application documentation for instructions on configuring the subagents.

Note – You can use the subcommand `sp get jnet` on the SP to retrieve the JNET IP address of the SP. For more information, see [“SP JNET Address Subcommands” on page 213](#).

Using a Third-Party MIB Browser

The following procedure demonstrates how to integrate the server MIBs into an SNMP node manager.

1. **From the browser’s Manager Preferences menu, choose Load/Unload MIBS: SNMP.**
2. **Locate and select the `SP-MasterAgent-MIB.mib`.**
3. **Click Load.**
4. **Specify the directory in which the server MIBs are placed and click Open.**
5. **Repeat [Step 2](#) through [Step 4](#) to load other MIBs.**

For example, `SP-SST-MIB.mib`, `SP-INVENTORY-MIB.mib`, `SP-EVENT-MIB.mib`, `SP-PLATFORM-MIB.mib`, `SP-GROUP-MIB.mib`, and so on.

6. **Exit the Manager Preferences menu.**
7. **Open an SNMP MIB browser.**
The SNMP standard tree displays in the MIB Browser.
8. **Locate the `newisys` branch located under `private.enterprises`.**
Refer to [FIGURE 3-1](#) for a sample view of the MIB tree.

Setting Options for Logging Events

You can also easily integrate SP-generated traps and set logging options. The following procedure demonstrates the necessary steps when using HP OpenView Network Node Manager (NNM).

1. **Load the `SP-EVENT-MIB.mib` according to the previous procedure.**
2. **Choose `Options>EventConfiguration`**
3. **Select the `spEvent` module from the Enterprises list.**
4. **Double-click an event from the Events for Enterprise `spEvent` list.**
5. **Select the Event Message tab.**
6. **Select `Log and Display in Category` and choose a category from the corresponding list, or create your own event category.**
7. **Select the severity of the event from the Severity list.**
8. **Enter a message or `$*` to display all information in the Event Log Message field.**
9. **Click OK.**

SNMP Traps

SNMP traps are network-management notifications of an event occurring at a managed network node. These events can identify problems in the network, machines that are up or down, and so on. The Sun Fire V20z and Sun Fire V40z servers use traps to signal conditions related to the server's health, including critical conditions related to physical components, the return to a normal state for these components, and other situations related to the state of the software running on the SP (for example, network settings are being reconfigured).

Traps are defined in the MIB files and are generated, received, and processed by an SNMP management station. SNMP trap data is uniquely identified by the MIB. Each SNMP trap contains information identifying the server's name, IP address, and other relevant data about the event.

Within the server event MIB, each trap has variables and event bindings, as described in [TABLE 3-1](#).

TABLE 3-1 Server Event Traps

Event	Description
EventID	Uniquely identifies the event on the SP from where it came.
EventSource	Denotes the source module that generated the event.
EventComponent	Denotes the component ID to which the event refers.
EventDescription	The event message received from its source.
EventTimeStampInitial	The time at which this event ID was initially generated.
EventTimeStampLast	The most recent time at which this event ID was generated.

Configuring SNMP Trap Destinations

Although SNMP traps are generated for events that occur on the SP, you must configure where these traps are to be sent. There is no default destination for traps. You can use the server-management subcommands (see [TABLE 3-2](#)) on the SP to configure SNMP destinations.

For more information on these subcommands, refer to [Appendix H](#).

TABLE 3-2 Subcommands for Configuring SNMP Destinations

Subcommand	Description
<code>sp get snmp-destinations</code>	Displays all the available SNMP destination IP addresses and host names to which the SP will send traps.
<code>sp add snmp-destination</code>	Adds a new SNMP destination, one IP address or host name at a time.
<code>sp delete snmp-destination</code>	Removes an existing SNMP destination, one IP address or host name at a time.

Configuring SNMP Destinations

Admin-level and manager-level users can define SNMP destinations to which SNMP events (alerts) will be sent. All users can view the current destinations (using read-only access).

The number of destinations you can create is limited due to memory constraints.

You can configure SNMP destinations using the `sp snmp` subcommands. For more information about these subcommands, refer to [Appendix H](#).

Server MIB Details

SNMP uses object identifiers (OIDs) to provide name variables by which objects are grouped together for easier reference. These servers provide agents for the MIBs shown in [TABLE 3-3](#).

TABLE 3-3 SNMP MIBs

MIB	OID	Description
SP-MasterAgent-MIB.mib	.1.3.6.1.4.1.9237	Creates the main trunk of the server MIB tree. All other MIBs of the SP branch from this tree. To be loaded first while integrating with any third-party framework.
SP-INVENTORY-MIB.mib	.1.3.6.1.4.1.9237.2.1.1.1 .1.3.6.1.4.1.9237.2.1.1.1.2 .1.3.6.1.4.1.9237.2.1.1.1.3	Used for querying inventory information for all Sun Fire V20z and Sun Fire V40z servers hardware and software components. Hardware Inventory Table —Collects all hardware component inventory. Software Inventory Table —Collects all software component inventory.
SP-SST-MIB.mib	.1.3.6.1.4.1.9237.2.1.1.4	Defines objects for the System State Table in the SP. Contains all sensor readings, including the name of the sensor, its current value, maximum allowed value, measurement type, scale, and scanning interval.

TABLE 3-3 SNMP MIBs (Continued)

MIB	OID	Description
SP-PLATFORM-MIB.mib	.1.3.6.1.4.1.9237.2.1.1.5	Defines objects for the platform SNMP, which includes OS state, platform state, and platform IP table.
SP-EVENT-MIB.mib	.1.3.6.1.4.1.9237.2.1.1.6	Identifies the OIDs associated with all SNMP traps originated from the SP.
SP-GROUP-MIB.mib	.1.3.6.1.4.1.9237.2.1.1.7	Defines objects for the SP, including host name, DNS, a reboot node, a node to hold the last port 80 postcode, a clone tree, and an IP table.

The events listed in [TABLE 3-4](#) are sent to the SNMP destination by SP-EVENT-MIB.mib.

TABLE 3-4 SP Events

Enterprise Trap ID	Event
1	spGenericEventInformational
2	spGenericEventWarning
3	spGenericEventCritical
4	spTemperatureEventInformational
5	spTemperatureEventWarning
6	spTemperatureEventCritical
7	spVoltageEventInformational
8	spVoltageEventWarning
9	spVoltageEventCritical
10	spFanEventInformational
11	spFanEventWarning
12	spFanEventCritical
13	spPlatformMachineCheckEventInformational
14	spPlatformMachineCheckEventWarning
15	spPlatformMachineCheckEventCritical
16	spPlatformStateChangeEventInformational
17	spPlatformStateChangeEventWarning

TABLE 3-4 SP Events (*Continued*)

Enterprise Trap ID	Event
18	spPlatformStateChangeEventCritical
19	spPlatformBIOSEventInformational
20	spPlatformBIOSEventWarning
21	spPlatformBIOSEventCritical
22	spGenericEventInformational
23	spGenericEventWarning
24	spGenericEventCritical
25	spTemperatureEventInformational
26	spTemperatureEventWarning
27	spTemperatureEventCritical
28	spVoltageEventInformational
29	spVoltageEventWarning
30	spVoltageEventCritical
31	spFanEventInformational
32	spFanEventWarning
33	spFanEventCritical
37	spPlatformStateChangeEventInformational
38	spPlatformStateChangeEventWarning
39	spPlatformStateChangeEventCritical
40	spPlatformBIOSEventInformational
41	spPlatformBIOSEventWarning
42	spPlatformBIOSEventCritical

SNMP Troubleshooting

You might find that SNMP queries to the service processor (SP) are timed out. If so, note that the platform OS requires both the Newisys Platform Software (NPS) driver suite RPM and an active SNMP daemon sharing the SP's community string.

Further Management Information

This chapter provides additional server management information. The following topics are discussed:

- [“Configuring Scripting Capabilities” on page 89](#)
- [“Console Redirection Over the Serial Port” on page 98](#)
- [“Network Share Volume” on page 104](#)
- [“Serial-Over-LAN” on page 106](#)
- [“Escape Sequences for Remote Console Terminal” on page 108](#)

Configuring Scripting Capabilities

A system administrator can log in to the service processor (SP) using Secure Shell (SSH) and issue commands, or more commonly, write a shell script that remotely invokes these operations.

Note – You must create a valid initial manager account before using SSH. The SP includes a setup account that can be used to set up an initial manager account. This initial manager user can create additional users.

For more information on the initial manager account, see [“Task II: Securing the SP” on page 16](#).

The SP includes a suite of commands that enables management and monitoring of the server; this suite of commands is referred to as server management commands. From the command line, for instance, you can write data-driven scripts that automate configuration of multiple machines.

The Sun Fire V20z and Sun Fire V40z Servers Network Share Volume CD contains sample scripts for getting started, which you can access after you extract the files on the CD. See “[Network Share Volume](#)” on page 104 for more information about the script locations.

Using Shell Scripts

An administrator can make configuration changes for a single SP by using SSH to log in and run commands. For a multisystem environment in which configurations for all SPs must be synchronized, you can automate configuration changes.

As a UNIX, Linux, or Microsoft Windows administrator, you can use SSH, trusted-host relationships, or public-key authentication, and UNIX or Linux shell scripting to automate tasks that need to be performed on multiple SPs.

Here is an overview of the scripting process:

1. Set up your system for scripting.

Remote scripting solutions for the servers depend on SSH for authentication and data encryption. If you do not already have SSH, you can obtain a free implementation, OpenSSH, available at <http://www.openssh.org>. The SP allows the use of SSH v2 only. Refer to “[Remote Scripting Using SSH](#)” on page 91.

2. Create a trusted-host relationship or add your public key for SSH authentication.

In order to use SSH in a scripted environment such that you are not prompted for a password upon the execution of each command, you can establish a trusted-host relationship between the machine from which the commands are sent and the SP on which the commands are executed. (This requires the prior creation of a manager-level user on the SP.) Refer to “[Creating Trusted-Host Relationships](#)” on page 92.

You can also add a public key for SSH authentication, allowing you to log in via SSH and execute remote commands without being prompted for a password. Refer to “[Adding Public Keys](#)” on page 93.

3. Configure your client for scripting.

You must configure the client machine on which you will be running scripts. Since Microsoft Windows does not natively support the SP trusted-host relationship feature, scripting from a Windows client requires you to install a UNIX/Linux-on-Windows toolset that supports SSH. Refer to “[Configuring a Microsoft Windows Client for Scripting](#)” on page 94.

4. Create your scripts.

Remote Scripting Using SSH

Remote scripting to the SP is done by using a program called SSH. For example, as a user on the UNIX machine `client.company.com` with the SP name `sp.company.com`, you could execute a command on the SP from the UNIX client using the following format:

```
# ssh sp.company.com command
```

Because the SSH server must authenticate the remote user, the user must either enter a password, or a trusted-host relationship must exist, or the remote user's public key must be installed on the SP.

If using trusted-host relationships for passwordless access, the SP must have a local user of the same name as the remote user (or the remote user should be a member of a directory service group that is mapped to a local SP administrative group).

You can also add your public key file instead of creating a trusted-host relationship to be authenticated via SSH. Refer to [“Adding Public Keys” on page 93](#).

When configured for passwordless access, the SSH daemon on the SP allows the remote user access to `sp.company.com` without a password, either for logging in, or for issuing remote SSH commands from the command line or from a script.

Configuring Multiple Systems for Scripting

There are two ways to configure multiple SPs for scripting:

- Execute the procedure to configure the client machine on which you will be running scripts for each SP. Refer to [“Configuring a Microsoft Windows Client for Scripting” on page 94](#).
- Set up the trust relationship or add your public key file on an initial machine and use the autoconfiguration feature to duplicate the configuration on each of the additional machines. Refer to [“Creating Trusted-Host Relationships” on page 92](#) and [“Adding Public Keys” on page 93](#).

Generating Host Keys

To establish a trusted-host relationship, you must set up a host key that is used to authenticate one host to another. The host's SSH installation should generate the host keys. If it does not, follow these steps to generate a host-key pair:

1. Type the following command:

```
# ssh-keygen -q -t rsa -f rsa_key -C '' -N ''
```

2. Move `rsa_key` to `/etc/ssh/ssh_host_rsa_key`.
3. Move `rsa_key.pub` to `/etc/ssh/ssh_host_rsa_key.pub`.
4. Ensure that only the root user has read or write permissions to `/etc/ssh/ssh_host_rsa_key`.
The `ssh_host_rsa_key.pub` file is the file you will transfer to the SP.

Note – Only protocol version 2 key types (RSA or DSA) and 1024-bit key sizes (the default generated by `ssh-keygen`) are supported.

5. Copy the host's public key (the file `ssh_host_rsa_key.pub`) to the SP using `scp` (secure copy), or by copying the host key to an external file system that has been mounted on the SP.

Note – Use `scp` to copy the files to either the `/tmp` directory or to your home directory. The `sp` commands will then install the file specified on the command line.

6. Continue with [“Creating Trusted-Host Relationships” on page 92](#) for instructions on creating public keys that can be used for passwordless access.

Creating Trusted-Host Relationships

Adding a trusted-host relationship is one way to allow for passwordless access and thus is a means for one-to-many scripting. Once a host-equivalence relationship has been created with a client, users on that client can remotely execute commands on the SP without being prompted for a password, provided one of the following conditions is met:

- The user's login name on the client is the same as that of a local user on the SP.
- The user's login on the client belongs to a directory service group that is mapped to an SP administrative group. (In this case, the SSH command executes as a well-known auxiliary user on the SP: either *rmonitor*, *radmin*, or *rmanager*.)

Note – Support is available for SSH protocol version 2 key types (RSA or DSA) only. If DNS is enabled on the SP, the client machine must be specified with its DNS name, not an IP address.

Manager-level users can create a trusted-host relationship for the specified host from the command line using the `access add trust` command:

```
# access add trust {-c | --client} HOST {-k | --keyfile} \  
PUBLIC KEY FILE
```

Adding Public Keys

Adding a user's public key is another way to allow for passwordless access and thus provide one-to-many scripting. Once a public key for a specific user has been installed on the SP, that user can remotely execute commands on the SP without being prompted for a password, if that user has installed the associated private key on the client.

Note – Support is available for SSH protocol version 2 key types (RSA or DSA) only.

Only local users can add public keys. Users who obtain authorization from directory services group mappings are not able to add public keys.

Local admin-level or manager-level users can add public keys using the `access add public key` command:

```
# access add public key -l PUBLIC_KEY_FILE [-u user]
```

The public-key file is your RSA or DSA key. Up to 10 users can install public keys; only one key per user is allowed.

Admin-level users can only add their own public key. Manager-level users can add a public key for any local user. If the *user* is not specified in the command, the current user is the default.

Note – The maximum supported key length is 4096 bits.

Generating a Host-Key Pair

To establish a trusted-host relationship, you must set up a host key, which is used to authenticate one host to another. Follow these steps to generate a host-key pair by copying the public key to the SP to which you want passwordless access.

1. Type the following command:

```
# ssh-keygen -t rsa -N
```

2. Accept the default values.

By default, this command generates the public and private keys, `$HOME/.ssh/id_rsa.pub`, and `$HOME/.ssh/id_rsa`. In this example, the public key (`$HOME/.ssh/id_rsa.pub`) will be installed on the service processor. The private key (`$HOME/.ssh/id_rsa`) can be encrypted with a password.

You can install the generated public key in one of two ways:

- You can transfer it to the service processor using the SSH software suite's `scp` program, for example, and install the key from the command line. For more information, see [“Access Public Key Subcommands” on page 131](#).
- You can use the Systems Management Console.

Configuring a Microsoft Windows Client for Scripting

To configure the client machine on which you will be running scripts, follow these steps.

1. **Create a manager-level user on both the client machine and the SP.**
You can create any user name as long as the user name exists on both machines.
2. **Define a host name for the SP.**
3. **Define a host name for the client machine.**
4. **Verify that both the SP and the client machine can resolve each other's addresses.**

Installing the Toolset Cygwin

Since Microsoft Windows does not natively support the trusted-host relationship feature on the SP, scripting from a Microsoft Windows client requires you to install a UNIX or Linux-on-Windows toolset that supports SSH.

To install the toolset Cygwin, follow these steps.

1. **Navigate to <http://www.cygwin.com>.**
2. **To launch the installer, click on one of the many “Install Cygwin now” links.**
3. **Save the `setup.exe` program to a local folder by choosing Save from the Download dialog box.**
4. **Open the folder and execute the `setup.exe` program.**
5. **Follow the prompts for the Install Wizard.**
The following options are recommended:
 - Download Source: Install from Internet
 - Root Install Directory: File type - Unix
 - Internet Connection: Direct Connection
6. **Choose a download mirror site.**
7. **In the Select Packages dialog box, open the Net Category and check the OpenSSH and OpenSSL items.**
8. **Complete the installation.**

Enabling SSH Access Using Trusted Hosts

Follow these steps to add users to the `local /etc/passwd` file to attempt trusted-host access to the service processor.

1. **Enable access for clients by launching a Bash shell.**
 - If you want all network accounts added, execute `mkpasswd >> /etc/passwd`.
 - If you want just local accounts added, execute `mkpasswd -l >> /etc/passwd`.
2. **Create or modify the file `/etc/ssh_config` to ensure it contains the following entry:**

```
Host *
HostbasedAuthentication yes
```
3. **Set up your host keys by running the following command:**

```
# ssh-host-config
```
4. **As a manager-level user on the client, run the following commands to establish a trusted-host relationship (*manager1* is used in the example in this step):**
 - a. **Copy the client key to `/tmp` on the SP.**

```
# scp /etc/ssh_host_dsa_key.pub manager1@sp.test.com:/tmp
```
 - b. **Authenticate yourself for the `scp` command by entering the password for your manager-level user.**
 - c. **Add the client key to the set of trusted hosts for this SP.**

```
# ssh manager1@sp.test.com access add trust -c
client.test.com\ -k /tmp/ssh_host_dsa_key.pub
```
 - d. **Authenticate yourself for the SSH command.**

From this point, any user with the same login on both `sp.test.com` and `client.test.com` has access without requiring a password to the like-named account on `sp.test.com`.

Generating a Host-Key Pair on Microsoft Windows

To set up your host-key pair, follow these steps.

1. **Launch an SSH client.**

On Microsoft Windows, launch a Cygwin Bash shell. Scripting from a Microsoft Windows client requires you to install a UNIX or Linux-on-Windows toolset that supports SSH. Refer to [“Configuring a Microsoft Windows Client for Scripting” on page 94](#).

2. Execute `ssh_host_config` to create both DSA and RSA keypairs:

```
/etc/ssh_host_dsa_key  
/etc/ssh_host_dsa_key.pub  
/etc/ssh_host_rsa_key  
/etc/ssh_host_rsa_key.pub
```

The host-key pairs are created in `/etc` for Microsoft Windows machines and in `/etc/ssh` on UNIX or Linux machines.

Enabling SSH Access Using Public Keys

Follow these steps to install public keys to enable SSH access.

1. Set up your host keys.

Refer to [“Generating a Host-Key Pair” on page 93](#).

2. Install your public key using the `access add public key` command.

3. Run the following command on the client machine:

```
# ssh-keygen -t rsa -N
```

This command generates `~/.ssh/id_dsa` and `~/.ssh/id_dsa.pub`.

4. Run the following command on the client machine:

```
# scp ~/.ssh/id_rsa.pub SP_IP:/tmp
```

Where `SP_IP` is the SP IP address displayed on the LCD operator panel. Type your password when prompted.

5. Run the following command on the client machine:

```
# ssh SP_IP access add public key -k /tmp/id_rsa.pub
```

Type your password when prompted.

6. Run the following command:

```
# ssh SP_IP rm -f /tmp/id_rsa.pub
```

From this point, you have access without requiring a password.

Guidelines for Writing Server Management Command Scripts

This section describes some basic guidelines for managing your systems by writing scripts for remote execution on one or more SPs.

- **Shell Scripts**—You should be familiar with standard shell scripting. Refer to [“Using Shell Scripts” on page 90](#).
- **SSH**—You must currently use an SSH (Secure Shell) client to execute automated command scripts. Refer to [“Remote Scripting Using SSH” on page 91](#).
- **Authentication**—To avoid being prompted each time you run a script on an SP, upload a public key or trusted-host key to each SP. Refer to [“Creating Trusted-Host Relationships” on page 92](#) and [“Adding Public Keys” on page 93](#).
- **Authorization Levels**—Access changes (such as adding users or uploading keys) typically require manager-level access while most other management tasks can be performed by an admin-level user.
- **Return Codes**—Every subcommand returns one or more return codes upon completion.
- **Nowait Argument**—Most commands complete their execution fairly quickly and are therefore performed synchronously. For some longer operations (such as rebooting the platform), a `--nowait` option is provided so that a script can initiate the operation without waiting for the prompt to return.
- **Quiet Argument**—The `delete` and `update` operations (such as `access delete user, sp delete event`) accept multiple targets. To ensure that a certain set of targets is deleted on a set of SPs, you can use the `--quiet` argument to suppress errors if one of the targets is not found, or to suppress interactive warning messages from the platform command.

Command Output

The following list defines common general output:

- Commands that complete successfully return 0 with no success return string. Some exceptions are commands that also return vital information.
- Table output, interactive warnings, and any other non-error messages are directed to standard output.
- Commands that return errors display the return codes and a descriptive error string.

Following are common characteristics of table output from a `get` command:

- Heading columns are provided by default for output with more than one column.
- Single-column output does not include a heading.
- To suppress headings, use the `-H` argument.
- Data for each column is left-aligned with at least one space between columns. Numeric data is right-aligned.
- The `-D` argument enables you to specify a delimiter character when scripting. This is very useful in parsing fields with white space.

- If all lines have the same number and type of data values, each row is printed to a separate line so that variable data can be parsed easily. For example, executing `access get users -g monitor` returns a list of monitor users each on a separate line.
- Commands that return multiple columns (such as `inventory get hardware`) might have a minimal default set of columns and a `--verbose` argument to display all columns. Some commands include arguments that enable you to select specific columns to output.

Other Best Practices for Using Server Management Command Scripts

- Externalize the set of SP IP addresses into a file to be shared across all of your scripts.
- Consider using a script to create the initial manager account and upload its public key on your SPs.
- Test the output and return codes of each command manually by using SSH to log in to the SP and run the commands individually.
- Test your scripts on a single staging machine before applying them to your remaining machines.
- To configure all of your SPs identically, consider configuring a single SP and then using the `sp load settings` command to synchronize that configuration on the remaining machines.

Note – If running the script from the SP, there is a limited number of commands (not a full Bash environment).

Console Redirection Over the Serial Port

Redirecting the console interaction over the serial port provides the user with another method to monitor the server.

The BIOS redirects console output to the serial port by default (9600, 8N1, no handshake).

This section describes how to configure these options for servers based both on the Linux OS and SolarisOS.

Linux-based Server



Caution – Redirecting the console over the serial port is a procedure intended for advanced users of Linux only. You can seriously disrupt the proper functioning of the server or render the server unbootable if you introduce a problem in the configuration files.

The goal of these configurations is to configure the bootloader to redirect its output, pass the kernel the proper parameters, and configure a login session on the serial port.

The BIOS redirects console output to the serial port by default (9600, 8N1, no handshake) until a bootloader program is run from the hard disk drive. The bootloader must be configured to support the serial console in addition to the keyboard, video, and mouse (KVM) console.

Two common bootloaders are `grub` and Linux Loader (LILO).



Caution – Do not edit the working-image section of your configuration files directly. Copy the working-image section and paste it within the configuration file. Make your editing changes to this copied section.

Using the grub Bootloader

If you use `grub`, there are three steps to enable console redirection over the serial port. These steps all involve editing the `grub` configuration file:

- If you are using Red Hat Linux, the `grub` file is `/etc/grub.conf`.
- If you are using SUSE Linux, the `grub` file is `/boot/grub/menu.lst`.

Note – On Red Hat Linux systems, the file `/etc/grub.conf` might be a symbolic link to the file `/boot/grub/grub.conf`.

The procedure in this section involves:

- Passing the proper console parameters to the kernel
- Configuring the `grub` menu system to redirect to the proper console
- Removing any splash images that would prevent the proper serial-console display

For more information on the parameters, refer to the file `kernel-parameters.txt` in your kernel documentation.

For more information on `grub`, run the command `info grub`.

Note – If the arrow keys do not work through your remote serial concentrator, you can use the keystroke combinations of CTRL+P and CTRL+N to highlight the Previous and Next entry, respectively. Pressing Enter then boots that entry.

The parameter `console=ttyS0` tells the system to send the data to the serial port first. The parameter `console=tty0` tells the system to send the data to the KVM second.

A working-image section in your `grub` configuration file should have an entry for the kernel image to boot. The stock kernel entry looks like this:

```
kernel /vmlinuz-kernel_revision ro root=/dev/sda5
```

Where `kernel_revision` is the kernel version that you are using.

Follow these steps to enable console redirection over the serial port using `grub`.

1. **Change the stock kernel entry of your image to include the console-kernel parameters, as follows:**

```
kernel /vmlinuz-kernel_revision ro root=/dev/sda5
console=ttyS0,9600 console=tty0
```

Note – These options should all be on one line with no wrap to a second line.

2. **Add the following two lines to the top of your `grub` configuration file:**

```
serial --unit=0 --speed=9600
terminal serial console
```

Adding these two lines at the beginning of the file sets up your serial port or your KVM as your `grub` console so that you can remotely or locally select a boot image from the `grub` menu.

3. **Comment out or remove the following line from your `grub` configuration file:**

```
splashimage=(hd0,1)/boot/grub/splash.xpm.gz
```

Removing the `splashimage` line allows for greater compatibility during your serial connection; with this line removed, the splash image does not prevent the proper `grub` menu from displaying.

Using the LILO Bootloader

Note – When you enable the BIOS option “Console Redirection After POST,” and LILO is used as a bootloader, the system might hang with an “L” printed on the screen.

This issue arises because there is not enough lower memory available to load the second-stage boot file that LILO uses. If you turn off the option “Console Redirection After POST” in the BIOS, the system will boot normally. See “[Enabling and Disabling BIOS Console Redirection](#)” on page 103.

If you require the option “Console Redirection After POST,” use `grub` or upgrade to a newer version of LILO. The current version of LILO is 22.5.9; to access the LILO pages, visit <http://lilo.go.dyndns.org/> and click on the link.

Before you upgrade, it is a best practice to verify with your OS vendor that the OS supports the updated version of LILO.

LILO uses the `append` feature in an image section in order to pass to the kernel the proper parameters for using the serial console.

Follow these steps to pass the proper parameters to the kernel.

1. **In the file `/etc/lilo.conf` on your Sun Fire V20z or V40z server, type the `consoles` in the `append` statement:**

```
append="console=ttyS0,9600 console=tty0"
```

2. **After modifying the file `/etc/lilo.conf`, run `lilo` from the command line to activate the change.**

For more information on LILO, run the commands `man lilo` or `man lilo.conf`.

Using `getty`

You can run a service called `getty` to enable login on the serial interface.

To enable `getty`:

- **Append the following line to the list of `gettys` in the `/etc/inittab` file:**

```
7:12345:respawn:/sbin/agetty 9600 ttyS0
```

Note – It does not matter where you append this line in the list.

Note – Make certain that the first number is unique within the `inittab` file.

The list of `gettys` currently looks like the following:

```
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Using `securetty`

To add the serial-console device `/dev/ttyS0` to the file `/etc/securetty`:

- **Run the following command:**

```
# echo ttyS0 >> /etc/securetty
```

Solaris-based Server



Caution – Redirecting the console over the serial port is a procedure intended only for advanced users of Solaris software. You can seriously disrupt the proper functioning of the server or render the server unbootable if you introduce a problem in the `bootenv.rc` file.

Note – The default setting for the output device is `screen` and the default setting for the input device is `keyboard`.

Changing the Settings

To enable console redirection over the serial port on a server running Solaris software:

- **In a terminal window, run the `eeprom` command to change the settings for the output and input devices, as shown here.**

```
eeprom output-device=ttya
eeprom input-device=ttya
```

Verifying the Settings

To verify that the changes were made:

1. **In a terminal window, run the `eeeprom` command with no arguments.**
The contents of the `bootenv.rc` file is displayed in the terminal window.
2. **Locate the following lines and verify that they display the correct values.**
`output-device=ttya`
`input-device=ttya`

Restoring the Default Settings

- **To reset the output and input devices to the default settings, run the `eeeprom` command with the following arguments.**
`eeeprom output-device=screen`
`eeeprom input-device=keyboard`

Enabling and Disabling BIOS Console Redirection

Note – Console redirection is enabled by default in the BIOS.

If the default settings have been changed in the BIOS, the following procedure explains how to change the console-redirection settings.

1. **Boot or reboot the server.**
2. **When prompted, press the F2 key to enter the BIOS Setup Utility.**
3. **Select the Advanced menu from the category selections along the top.**
4. **Select Console Redirection.**

Note – Make note of all settings in this menu, as they are required for configuring the remote-console access and the serial-over-LAN (SOL) feature.

- To disable Console Redirection to Serial, select Disabled from the Com Port Address option.
- To change the baud rate, select the desired bit rate from the Baud Rate option.
- To disable Continue Console Redirection after POST, toggle the setting to OFF.

5. Save the changes to the BIOS settings.
6. Press the F10 key to exit the BIOS Setup Utility.

For the new settings to take effect, you must reboot the server.

Network Share Volume

A network share volume (NSV) structure is included with the server on the Sun Fire V20z and Sun Fire V40z Servers Network Share Volume CD.

Although the SP functions normally without access to an external file system, a file system is required to enable several features, including event log files, software updates, diagnostics, and the Troubleshooting Dump Utility. You can configure the NSV to be shared among multiple SPs. Admin-level and manager-level users can configure the external file system; regular users can only view the current configuration.

The following software components are included with the server:

- Platform BIOS
- SP base software
- SP value-add software
- Update file for downloading Java Runtime Environment (JRE) packages
- Network share volume software, which includes diagnostics
- Platform software
- Motherboard platform drivers

All of these software packages are packaged with the NSV and are installed on the file server when the external file system is installed and configured.

For instructions on extracting and installing the NSV software, refer to the *Sun Fire V20z and Sun Fire V40z Servers—Installation Guide*.

Network Share Volume Structure

[TABLE 4-1](#) lists the compressed packages that are included on the Sun Fire V20z and Sun Fire V40z Servers Network Share Volume CD.

TABLE 4-1 Compressed Packages on the Network Share Volume CD

File Name	File Contents
nsv_V2_2_0_x.zip	SP and platform diagnostics with some support for the SP software
nsv-redhat_V2_2_0_x.zip	Drivers for Red Hat Linux OS
nsv-solaris_V2_2_0_x.zip	Drivers for Solaris 9 OS and Solaris 10 OS
nsv-suse_V2_2_0_x.zip	Drivers for SUSE Linux OS

When extracted, the compressed packages in [TABLE 4-1](#) populate the following directories on the NSV:

```
/mnt/nsv/  
diags  
logs  
scripts  
snmp  
update_server  
sw_images (this folder appears after you extract one of the OS-specific .zip files)
```

TABLE 4-2 Extracted Files on the Network Share Volume

File Name	Description
diags	Offline location of the server diagnostics.
logs	Offline location of the log files for the SP.
scripts	Sample scripts that can be used for scripting commands.
snmp	SNMP MIBs. Refer to Chapter 3 for details.
update_server	Application for updating the SP software and BIOS. Refer to Chapter 1 for details.
sw_images	Contains a directory hierarchy of platform and SP components, including subdirectories for each version.

Serial-Over-LAN

The serial-over-LAN (SOL) feature lets servers transparently redirect the serial character stream from the baseboard universal asynchronous receiver/transmitter (UART) to and from the remote-client system over a LAN. Serial-over-LAN has the following benefits compared to a serial interface:

- Eliminates the need for a serial concentrator
- Reduces the amount of cabling
- Allows remote management of servers without video, mouse, or keyboard (headless servers)

Serial-over-LAN requires a properly configured LAN connection and a console from which an `ssh` session can be established.

In a Linux environment, you can use a shell such as `csh` or `ksh` as your console. This console works well in a scripting environment in which you might want to monitor many servers.

Enabling or Disabling the SOL Feature on the Server

Note – When the SOL feature is enabled, you cannot access the server through the external DB9 serial port (COM A).

Note – The variable `spuser` is the user account created when securing the SP. The variable `spipaddr` is the IP address assigned to the SP. For more information, see [“Initial Setup of the SP” on page 12](#).

You can enable or disable the SOL feature through the SP.

Enabling the SOL Feature

- **To enable the SOL feature, run the following command:**

```
# ssh -l spuser spipaddr platform set console -s sp -e -S 9600
```

Note – Ensure that the baud rate value passed to the `-S` argument matches the speed that was specified for the serial-redirection feature of the BIOS and the speed used for your boot loader and OS configuration. The default baud rate in the BIOS settings is 9600.

Disabling the SOL Feature

- To disable the SOL feature, run the following command:

```
# ssh -l spuser spipaddr platform set console -s platform
```

Starting and Stopping an SOL Session

You can start or stop an SOL session through the SP.

Starting an SOL Session

- To launch an SOL session, run the following command:

```
# ssh spipaddr -l spuser platform console
```

Stopping an SOL Session

To stop an SOL session:

1. Press **Ctrl-E**.
2. Press the **'c'** key.
3. Press the **period key (.)**.

You can also stop an SOL session by terminating the `ssh` session:

1. Press **Enter**.
2. Press the **tilde key (~)**.
3. Press the **period key (.)**.

Escape Sequences for Remote Console Terminal

If you are accessing your server using a remote console terminal, you might need to use the escape sequences shown in [TABLE 4-3](#). If a regular function key is not working properly, use the escape sequence listed next to it in the table.

You will most likely need to use the escape sequences if you are using a Linux or Solaris OS.

TABLE 4-3 Special Keys for Remote Console Terminal

Function Key	Escape Sequence
HOME	<ESC> h
END	<ESC> k
INSERT	<ESC> +
DELETE	<ESC> -
PAGE UP	<ESC> ?
PAGE DOWN	<ESC> /
ALT	<ESC> ^A
CTRL	<ESC> ^C
F1	<ESC> 1
F2	<ESC> 2
F3	<ESC> 3
F4	<ESC> 4
F5	<ESC> 5
F6	<ESC> 6
F7	<ESC> 7
F8	<ESC> 8
F9	<ESC> 9
F10	<ESC> 0
F11	<ESC> !
F12	<ESC> @

Server Management Commands Summary

The service processor (SP) includes a suite of commands that enables management and monitoring of the server. This suite of commands is referred to as the server-management commands.

Note – This appendix provides an overview of the server-management command groups that are available on the SP. For a detailed description of the subcommands, arguments, and return codes for each command type, refer to the appendices in this guide, as described in [TABLE A-1](#).

Using the ssh Protocol

You must use `ssh` to execute these commands on the service processor (SP). There are two ways to do this:

- Use the interactive shell on the SP.
- Preface each command with a set piece of text.

Interactive Shell on the SP

To use the interactive shell:

- **Log in to the interactive shell by running the command:**

```
# ssh -l spipaddr spuser
```

Preface Text

- Preface each command with the following text:

```
# ssh -l spipaddr spuser
```

Commands

The server-management commands take arguments, perform one or more actions, and display the result or text to the standard output device. Commands are grouped by similar function; each command has numerous subcommands supporting functions within that grouping.

Note – Every subcommand (except `help`) returns a return code upon completion. See “Return Codes” on page 111 for a summary.

TABLE A-1 lists the server-management command groups.

TABLE A-1 Server-Management Commands

Command Group	Description
<code>access</code>	Allows the authorized user to manage and monitor access control and security features of the SP, such as users, groups, SSL, and so on. See Appendix B .
<code>diags</code>	Manages diagnostics tests that are included with your server. See Appendix C .
<code>inventory</code>	Allows the authorized user to monitor hardware and software inventory information. See Appendix D .
<code>ipmi</code>	Manages IPMI functions. See Appendix E .
<code>platform</code>	Allows the authorized user to manage and monitor platform activities, such as rebooting the platform operating system, gathering system status and so on. See Appendix F .

TABLE A-1 Server-Management Commands (*Continued*)

Command Group	Description
sensor	Reports or sets the value of an environmental sensor or control. See Appendix G .
sp	Allows the authorized user to manage and monitor the SP configurations, such as networking, external file system, SNMP, SMTP, SSL, event logs, and so on. See Appendix H .
help	Returns the following text: Available Commands: platform, access, sp, sensor, inventory, ipmi. Each of these commands includes a help option (--help).

Return Codes

Every subcommand returns one or more of the following return codes upon completion. Refer to the following appendices in this user guide for each subcommand and the corresponding return codes for that subcommand.

[TABLE A-2](#) lists the return codes for the server-management commands.

TABLE A-2 Return Codes

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NotFound	5	Entity (user, service, file, path, or other) was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_MissingArgument	7	Missing argument(s).
NWSE_NoMemory	8	Insufficient memory.
NWSE_Busy	9	Device or resource is busy.

TABLE A-2 Return Codes (*Continued*)

Return Code	ID	Description
NWSE_NotImplemented	10	Function not implemented.
NWSE_RPCConnected	11	RPC client already connected.
NWSE_RPCConnRefused	12	RPC connection refused.
NWSE_NoRouteToHost	13	No route to host (network down).
NWSE_HostDown	14	Host is down.
NWSE_UnknownError	15	Miscellaneous error not captured by other errors.
NWSE_GatewayOffNet	16	Gateway address is not on the network.
NWSE_NetMaskIncorrect	17	An inappropriate netmask was specified.
NWSE_FileError	18	File open, file missing, or a read or write error occurred.
NWSE_Exist	19	Entity (user, service, or other) already exists.
NWSE_NotRecognized	20	Request not understood.
NWSE_NotMounted	21	File system is not mounted.
NWSE_InvalidOpForState	22	Invalid operation for current state.
NWSE_TimedOut	23	Operation timed out.
NWSE_ServiceNotAvailable	24	Requested service is not available.
NWSE_DeviceError	25	Unable to read or write to the device.
NWSE_LimitExceeded	26	Limit has been exceeded.

Access Commands

The `access` command validates a user's authority or controls authorization services. Using the `access` command, you can retrieve information about user groups, add a user to or delete a user from a group, and specify a mapping between site-defined administrative groups and the administrative groups that are used to authorize actions on the service processor.

Note – [TABLE B-1](#) lists the groups of `access` subcommands. Every subcommand returns a return code upon completion.

TABLE B-1 Access Subcommand Groups

Subcommand Group	Description
<code>access config-sharing</code>	Controls configuration sharing in order to perform autoconfiguration.
<code>access groups</code>	Returns the authorization group for a specific user or a list of defined groups.
<code>access map</code>	Maps, unmaps, and returns a list of existing site-specified group names (the directory service group) mapped to one of the standard administrative groups.
<code>access public key</code>	Manages public keys and public key users.
<code>access services</code>	Enables, disables, or defines a directory services mechanism that determines a user's group memberships.
<code>access trust</code>	Creates a host-based trust relationship for the specified host.
<code>access user</code>	Manages local users or a group of users.

Access Config-Sharing Subcommands

The subcommands in [TABLE B-2](#) control the configuration-sharing feature. This feature is required for autoconfiguration.

TABLE B-2 Access Config-Sharing Subcommands

Subcommand	Description
<code>access enable config-sharing</code>	Allows the SP to be a source for configuration settings for other SPs.
<code>access disable config-sharing</code>	Prevents the SP from being a source for configuration settings for other SPs.
<code>access get config-sharing</code>	Returns the value of the configuration sharing setting.

Access Enable Config-Sharing Subcommand

Description: This command is run on the SP. It enables one SP to be a source of configuration settings for other SPs. After you enable the `config-sharing` setting on one SP, any other SP with network access to the first server can replicate the configuration settings of the first server.

Format

```
access enable config-sharing
```

Return Codes

[TABLE B-3](#) lists the return codes for this subcommand.

TABLE B-3 Return Codes for Subcommand `access enable config-sharing`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.

TABLE B-3 Return Codes for Subcommand `access enable config-sharing`

Return Code	ID	Description
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoPermission	6	Not authorized to perform this operation.

Access Disable Config-Sharing Subcommand

Description: This command is run on the SP. It prevents an SP from being a source of configuration settings for other SPs.

Format

```
access disable config-sharing
```

Return Codes

[TABLE B-4](#) lists the return codes for this subcommand.

TABLE B-4 Return Codes for Subcommand `access disable config-sharing`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoPermission	6	Not authorized to perform this operation.

Access Get Config-Sharing Subcommand

Description: This command returns the value of the configuration-sharing setting.

Format

```
access get config-sharing
```

Values

TABLE B-5 lists the values for this subcommand.

TABLE B-5 Values for Subcommand `access get config-sharing`

Value	Description
Enabled	Allows configuration-settings sharing. The SP is a source of configuration settings for other SPs.
Disabled	Prevents configuration-settings sharing. The SP is blocked from being a source of configuration settings for other SPs.

Return Codes

TABLE B-6 lists the return codes for this subcommand.

TABLE B-6 Return Codes for Subcommand `access get config-sharing`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoPermission	6	Not authorized to perform this operation.

Access Group Subcommands

The subcommands in [TABLE B-7](#) return the authorization group for a specific user or for a list of defined groups.

TABLE B-7 Access Group Subcommands

Subcommand	Description
<code>access get group</code>	Returns the authorization group for the specified user.
<code>access get groups</code>	Returns a list of the groups defined, including the standard groups.

Access Get Group Subcommand

Description: Returns the authorization group for the specified user.

Format

```
access get group
```

Return Codes

[TABLE B-8](#) lists the return codes for this subcommand.

TABLE B-8 Return Codes for Subcommand `access get group`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.
<code>NWSE_RPCTimeout</code>	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
<code>NWSE_RPCNotConnected</code>	3	Unable to connect to the RPC server.
<code>NWSE_NotFound</code>	5	Entity (user, service, file, path, or other) was not found.

Access Get Groups Subcommand

Description: Returns a list of the groups defined, including the standard groups.

Format

```
access get groups
```

Return Codes

[TABLE B-9](#) lists the return codes for this subcommand.

TABLE B-9 Return Codes for Subcommand `access get groups`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NotFound	5	Entity (user, service, file, path, or other) was not found.

Access Map Subcommands

The subcommands in [TABLE B-10](#) manage mappings between existing site-specified groups and one of the standard administrative groups.

TABLE B-10 Access Map Subcommands

Subcommand	Description
<code>access get map</code>	Returns the names of all the site-specified groups mapped to a specific administrative group.
<code>access map</code>	Maps an existing site-specified group name (the directory-service group) to one of the standard administrative groups.
<code>access unmap</code>	Removes the directory-service group and administrative group mapping.

Access Get Map Subcommand

Description: Returns the names of all the site-specified groups mapped to a specific administrative group.

Format

```
access get map LOGICAL_GROUP_NAME [{-D | --Delim}]  
[{-H | --noheader}]
```

Note – To return mappings for all groups, omit the group name from the command line.

[TABLE B-11](#) lists the arguments for this subcommand.

TABLE B-11 Arguments for Subcommand `access get map`

Argument	Description
{ -H --noheader }	Suppresses column headings.
{ -D --Delim }	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

[TABLE B-12](#) lists the return codes for this subcommand.

TABLE B-12 Return Codes for Subcommand `access get map`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.

Access Map Subcommand

Description: Maps an existing site-specified group name (the directory-services group) to one of the standard administrative groups.

Format

```
access map {-d | --dsgroup} DIRECTORY-SERVICES-GROUP  
{-g | --group} LOCAL-GROUP {-v | --verify}
```

[TABLE B-13](#) lists the arguments for this subcommand.

TABLE B-13 Arguments for Subcommand `access map`

Argument	Description
{-d --dsgroup}	The name of the directory-services group for which you want to map to a standard administrative group.
{-g --group}	The name of the standard administrative group to which you want to map to the directory-services group.
{-v --verify}	Verifies the group existence.

Return Codes

[TABLE B-14](#) lists the return codes for this subcommand.

TABLE B-14 Return Codes for Subcommand `access map`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NotFound	5	Entity (user, service, file, path, or other) was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_InvalidOpForState	22	Invalid operation for current state.

Access Unmap Subcommand

Description: Removes the directory service group and administrative group mapping.

Format

```
access unmap [-a | --all] DIRECTORY-SERVICES-GROUP
```

[TABLE B-15](#) lists the arguments for this subcommand.

TABLE B-15 Arguments for Subcommand `access unmap`

Argument	Description
<i>DIRECTORY-SERVICES-GROUP</i>	The name of the directory services group for which you want to remove a mapping.
<code>[-a --all]</code>	Removes mappings for all of the directory services groups.

Return Codes

[TABLE B-16](#) lists the return codes for this subcommand.

TABLE B-16 Return Codes for Subcommand `access unmap`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.
<code>NWSE_RPCTimeout</code>	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
<code>NWSE_RPCNotConnected</code>	3	Unable to connect to the RPC server.
<code>NWSE_InvalidArgument</code>	4	One or more arguments were incorrect or invalid.
<code>NWSE_NoPermission</code>	6	Not authorized to perform this operation.

Access Directory Services Subcommands

Services defines a directory-services mechanism that determines the group memberships for a user. Remote users gain access to the SP features only through these group mappings that relate a directory-services group to a local SP administrative group.

Therefore, using the command `access map`, the administrator must set up the appropriate directory-services configuration and create mappings from the directory-services groups to local SP administrative groups.

[TABLE B-17](#) lists the Access Directory Services subcommands.

TABLE B-17 Access Directory Services Subcommands

Subcommand	Description
<code>access disable service</code>	Disables a directory service.
<code>access enable service</code>	Enables a directory service.
<code>access get services</code>	Defines a directory-services mechanism that determines the group memberships for a user.

Access Disable Service Subcommand

Description: Disables a directory service (either NIS or ADS) from the name-service lookup system on the SP.

Format

```
access disable service {nis | ads}
```

[TABLE B-18](#) lists the argument for this subcommand.

TABLE B-18 Argument for Subcommand `access disable service`

Argument	Description
<code>{nis ads }</code>	Specifies the service type: NIS or ADS.

Return Codes

[TABLE B-19](#) lists the return codes for this subcommand.

TABLE B-19 Return Codes for Subcommand `access disable service`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_InvalidOpForState	22	Invalid operation for current state.

Access Enable Service Subcommand

Description: Enables a directory service (either NIS or ADS) to name-service lookup system on the SP.

Format

```
access enable service NIS {-d | --domain} DOMAIN NAME {-s | --server } SERVER
```

```
access enable service ADS {-d | --domain} DOMAIN NAME {-s | --server } SERVER {-k | --keytab} KEYTAB FILENAME {-o | --ou} ORGANIZATIONAL UNIT {-l|--logon} LOGON
```

[TABLE B-20](#) lists the arguments for this subcommand.

TABLE B-20 Arguments for Subcommand `access enable service`

Argument	Description
{-d --domain}	Specifies the domain name.
{-s --server}	Specifies the server.
{-k --keytab}	For ADS only: Specifies the ADS keytab file name.
{-o --ou}	For ADS only: Specifies the organizational unit under which the name-service library looks for group data.
{-l --logon}	For ADS only: Specifies the login ID for the active directory account.

To use ADS as a directory service on the SP, you must create an active directory account. The name-service library on the SP uses this account to authenticate itself to the LDAP interface of the active directory server. A Microsoft Windows administrator can create the keytab for this account using the following command:

```
ktpass -princ <logon>@<domain> -pass <password> -mapuser <logon> -out <output filename>
```

The keytab file must then be securely transferred to the SP using an encrypted file-transfer mechanism.

The clock on the SP must be accurate and DNS must be set up (meaning that the SP must have a DNS record).

If a directory service has been previously enabled, you can specify the following command and options; the saved settings are then used to re-enable the service.

```
access enable service -t <nis | ads>
```

Return Codes

[TABLE B-21](#) lists the return codes for this subcommand.

TABLE B-21 Return Codes for Subcommand `access enable service`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.

TABLE B-21 Return Codes for Subcommand `access enable service` (Continued)

Return Code	ID	Description
NWSE_NotFound	5	Entity (user, service, file, path, etc.) was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_FileError	18	File open, file missing, or a read or write error occurred.
NWSE_InvalidOpForState	22	Invalid operation for current state.

Access Get Services Subcommand

Description: Returns a string containing the current naming services option (NIS or ADS).

Format

```
access get services [ {-t | --type } NIS
[{-d | --domain} | {-s | --server}]
[-H | --noheader]] [{-D | --Delim <DELIMITER>}]
```

```
access get services [ {-t | --type } ADS
[{-d | --domain} | {-s | --server} |
{-l | --logonID} | {-o | --ou}]
[-H | --noheader]] [{-D | --Delim <DELIMITER>}]
```

[TABLE B-22](#) lists the arguments for this subcommand.

TABLE B-22 Arguments for Subcommand `access get services`

Argument	Description
<code>{-t --type }</code>	Returns information about the configuration of either the NIS or ADS service. You must specify <code>-t</code> to return a list of enabled services.
<code>{-d --domain}</code>	Returns domain information. Only one of the parameters <code>-d</code> and <code>-s</code> are permitted at a time.
<code>{-s --server}</code>	Returns server information. Only one of the parameters <code>-d</code> and <code>-s</code> are permitted at a time.
<code>{-l --ID}</code>	For ADS only: Returns the ADS login ID. Only one of the parameters <code>-o</code> and <code>-l</code> are permitted at a time.

TABLE B-22 Arguments for Subcommand `access get services` (*Continued*)

Argument	Description
{-o --ou}	For ADS only: Returns the organization unit information. Only one of the parameters -o and -l are permitted at a time.
[-H --noheader]	Suppresses header output.
[-D --Delim <DELIMITER>]	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

[TABLE B-23](#) lists the return codes for this subcommand.

TABLE B-23 Return Codes for Subcommand `access get services`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.

Access Trust Subcommands

Adding host-based trusts provides many-to-one scripting solutions. Once a host-equivalence relationship has been created with a client, users on that client can remotely execute commands on the SP without being prompted for a password.

[TABLE B-24](#) lists the commands related to trusted-host relationships.

TABLE B-24 Access Trust Subcommands

Subcommand	Description
<code>access add trust</code>	Creates a host-based trust relationship for the specified host.
<code>access delete trust</code>	Removes a host-based trust relationship for the specified host.
<code>access get trusts</code>	Requests a list of hosts involved in trust relationships with the SP.

Access Add Trust Subcommand

Description: Creates a host-based trust relationship for the specified host. Adding host-based trusts provides many-to-one scripting solutions. Once a host-equivalence relationship has been created with a client, users on that client can remotely execute commands on the SP without being prompted for a password, provided one of the following conditions is met:

- Their login on the client has the same user name as a local user on the SP.
- Their login on the client is in a directory-service group that is mapped to an SP administrative group.

Format

```
access add trust {-c | --client} HOST {-k | --keyfile} PUBLIC KEY FILE
```

[TABLE B-25](#) lists the arguments for this subcommand.

TABLE B-25 Arguments for Subcommand `access add trust`

Arguments	Description
<code>{-c --client}</code>	Specifies the host for which to create the relationship.
<code>{-k --keyfile}</code>	Specifies the public key file.

If the login is authorized through a mapping of a directory-service group, the `ssh` command is executed as the proxy user on the SP: either *rmonitor*, *radmin*, or *rmanager*.

Support is available for SSH protocol version 2 key types (RSA or DSA) only.

If DNS is enabled on the SP, the client machine must be specified with its DNS name, (and not the IP address).

Generating Host Keys

The host's `ssh` installation should generate the host keys. If it does not, follow these steps to manually generate the key pair:

1. Type the following command:

```
ssh-keygen -q -t rsa -f rsa_key -C '' -N ''
```

2. Copy `rsa_key` to `/etc/ssh/ssh_host_rsa_key`.

3. Ensure that only the root user has read or write permission to this file. The `rsa_key.pub` file is the file you will transfer to the SP.

Note – Only protocol version 2 key types and 1024-bit key sizes (the default generated by `ssh-keygen`) are supported.

4. Copy the host's public key (the `rsa_key.pub` file) to the SP using `scp` (secure copy), or by copying the host key to an external file system that has been mounted on the SP.

Note – Use `scp` to copy the files to either `/tmp` or to your home directory. The SP commands will then install the file specified on the command line to `/pstore`.

Note – If DNS is enabled on the SP, you must specify the client that is used in the trust commands with its DNS name (and not the IP address).

Return Codes

[TABLE B-26](#) lists the return codes for this subcommand.

TABLE B-26 Return Codes for Subcommand `access add trust`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_FileError	18	File open, file missing, or a read or write error occurred.
NWSE_Exist	19	Entity (user, service, or other) already exists.

Access Delete Trust Subcommand

Description: Removes a host-based trust relationship for the specified host.

Format

```
access delete trust CLIENT HOSTNAME [-a | --all] [-q | --quiet]
```

[TABLE B-27](#) lists the arguments for this subcommand.

TABLE B-27 Arguments for Subcommand `access delete trust`

Argument	Description
<i>CLIENT HOSTNAME</i>	Specifies the name of the client to remove.
[-a --all]	Removes all trust relationships.
[-q --quiet]	If the trust relationship to delete is not found, this argument specifies that no error be returned.

Return Codes

[TABLE B-28](#) lists the return codes for this subcommand.

TABLE B-28 Return Codes for Subcommand `access delete trust`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NotFound	5	Entity (user, service, file, path, or other) was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_DeviceError	25	Error deleting trusted host. Insufficient space in /tmp.

Access Get Trusts Subcommand

Description: Requests a list of hosts involved in trust relationships with the SP.

Format

```
access get trusts
```

Return Codes

[TABLE B-29](#) lists the return codes for this subcommand.

TABLE B-29 Return Codes for Subcommand `access get trusts`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.

TABLE B-29 Return Codes for Subcommand `access get trusts` (Continued)

Return Code	ID	Description
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoPermission	6	Not authorized to perform this operation.

Access Public Key Subcommands

The subcommands listed in [TABLE B-30](#) allow you to manage public keys and public-key users.

TABLE B-30 Access Public Key Subcommands

Subcommand	Description
<code>access add public key</code>	Installs a public key for SSH authentication.
<code>access get public key users</code>	Determines which users have public keys installed.
<code>access delete public key</code>	Removes a user's public key.

Access Add Public Key Subcommand

Description: Installs a public key for SSH authentication, which enables SSH logins and remote command execution without being prompted for a password. You must first generate a key pair (RSA or DSA), which you can generate using the `ssh-keygen` command included with OpenSSH.

- Only local users can install public keys (not users who gain authorization through a mapping of a directory-services group).
- Manager-level users can add keys for any local user.
- Admin-level users can add only themselves.
- Service-level users can not add anyone.
- Up to 10 users can install public keys; each user can install only one key.
- The maximum key length supported is 4096 bits.

Format

```
access add public key {-k | --keyfile} PUBLIC_KEY_FILE [-u | --user] USER
```

TABLE B-31 lists the arguments for this subcommand.

TABLE B-31 Arguments for Subcommand `access add public key`

Arguments	Description
{-k --keyfile}	Specifies the user's public RSA or DSA key.
{-u --user}	Specifies the user for which this key will be installed. The default is the current user if no user is specified.

Return Codes

TABLE B-32 lists the return codes for this subcommand.

TABLE B-32 Return Codes for Subcommand `access add public key`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid. The group specified with <code>-g</code> is an invalid local SP administrative group or the length of the user name or password exceeds the maximum length.
NWSE_NotFound	5	Entity (user, service, file, path, or other) was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_Exist	19	The user already exists.
NWSE_LimitExceeded	26	Limit has been exceeded.

Access Get Public Key Users Subcommand

Description: Determines which users have public keys installed.

Format

```
access get public key users
```

Return Codes

[TABLE B-33](#) lists the return codes for this subcommand.

TABLE B-33 Return Codes for Subcommand `access get public key users`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoPermission	6	Not authorized to perform this operation.

Access Delete Public Key Subcommand

Description: All users can execute this command to remove their own individual public key. Manager-level users can execute this command to remove the public key for any user.

Format

```
access delete public key [-u | --user] USER [-a | --all] [-q | --quiet]
```

TABLE B-34 lists the arguments for this subcommand.

TABLE B-34 Arguments for Subcommand `access delete public key`

Arguments	Description
[-u --user]	The user whose public key will be removed. Defaults to the current user. If <i>USER</i> is not specified, this argument is repeatable to remove multiple public keys at one time.
[-a --all]	Removes all public keys.
[-q --quiet]	If the user to delete is not found, this argument specifies that no error be returned.

Return Codes

TABLE B-35 lists the return codes for this subcommand.

TABLE B-35 Return Codes for Subcommand `access delete public key`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NotFound	5	Entity (user, service, file, path, or other) was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.

Access User Subcommands

The subcommands listed in [TABLE B-36](#) allow you to manage a single user or group of users.

TABLE B-36 Access User Subcommands

Subcommand	Description
<code>access add user</code>	Adds the specified local user to the specified group.
<code>access delete user</code>	Deletes the specified user.
<code>access get users</code>	Retrieves all the users in an administrative group or all users in all groups.
<code>access update password</code>	Updates the password of the specified user.
<code>access update user</code>	Updates the login information for the specified user.

Access Add User Subcommand

Description: Adds the specified local user to the specified group with the specified user name and password.

Format

```
access add user {-p | --password} PASSWORD {-g | --group} GROUP  
{-u | --user} USERNAME
```

[TABLE B-37](#) lists the arguments for this subcommand.

TABLE B-37 Arguments for Subcommand `access add user`

Arguments	Description
<code>{-p --password}</code>	Specifies the password for the new user. The password is optional and if not specified, a prompt displays requesting confirmation.
<code>{-g --group}</code>	Specifies the group to which the new user will belong.
<code>{-u --user}</code>	Specifies the name of the new user to add. This argument is repeatable to add multiple users at one time.

Return Codes

[TABLE B-38](#) lists the return codes for this subcommand.

TABLE B-38 Return Codes for Subcommand `access add user`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid. The group specified with <code>-g</code> is an invalid local SP administrative group or the length of the user name or password exceeds the maximum length.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_Exist	19	The user already exists.

Access Delete User Subcommand

Description: Deletes a user.

Format

```
access delete user USERNAME [-a | --all] [-q | --quiet]
```

[TABLE B-39](#) lists the arguments for this subcommand.

TABLE B-39 Arguments for Subcommand `access delete user`

Argument	Description
<i>USERNAME</i>	Specifies the name of the user to remove. This argument is repeatable to remove multiple users at one time.
[-a --all]	Removes all user accounts. The manager-level user executing the command is not removed.
[-q --quiet]	If the user to delete is not found, this argument specifies that no error be returned.

Return Codes

TABLE B-40 lists the return codes for this subcommand.

TABLE B-40 Return Codes for Subcommand `access delete user`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NotFound	5	Specified user was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.

Access Get Users Subcommand

Description: Retrieves all the local users in an administrative group.

Format

```
access get users {-g | --group} [{-H | noheader}][{-D | --Delim  
<DELIMITER>}]
```

TABLE B-41 lists the arguments for this subcommand.

TABLE B-41 Arguments for Subcommand `access get users`

Argument	Description
<code>{-g --group}</code>	Specifies that group from which to retrieve all users.
<code>{ -H --noheader }</code>	Specifies that column headings should be suppressed.
<code>{ -D --Delim }</code>	Specifies to delimit columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

[TABLE B-42](#) lists the return codes for this subcommand.

TABLE B-42 Return Codes for Subcommand `access get users`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.

Access Update Password Subcommand

Note – This command is for managers to change other users' passwords; all users can change their own passwords.

Description: Changes the password of an existing user.

Format

```
access update password {-p | --password} PASSWORD {u | --user} USER
```

[TABLE B-43](#) lists the arguments for this subcommand.

TABLE B-43 Arguments for Subcommand `access update password`

Argument	Description
{-u --user}	The name of the user whose password you want to update. If a user name is not specified, the current user is implied. You must have manager-level access to change another user's password. This argument is repeatable to update multiple user's passwords at one time.
{-p --password}	The user's new password. If a password is not specified, a prompt appears to enter the password and again to confirm the password.

Return Codes

[TABLE B-44](#) lists the return codes for this subcommand.

TABLE B-44 Return Codes for Subcommand `access update password`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NotFound	5	Entity (user, service, file, path, or other) was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.

Access Update User Subcommand

Description: Updates the login information (password or group) for the user.

Format

```
access update user {-u | --user} USER {-p | --password} PASSWORD  
{-g | --group} GROUP
```

[TABLE B-45](#) lists the arguments for this subcommand.

Note – The `-p` and `-g` arguments are optional but you must specify at least one.

TABLE B-45 Arguments for Subcommand `access update user`

Argument	Description
<code>{-u --user}</code>	The name of the user to update.
<code>{-p --password}</code>	The user's new password. The <code>-p</code> and <code>-g</code> options are optional but you must specify at least one.
<code>{-g --group}</code>	The new group to which to reassign to the user. The <code>-p</code> and <code>-g</code> options are optional but you must specify at least one.

Return Codes

[TABLE B-45](#) lists the return codes for this subcommand.

TABLE B-46 Return Codes for Subcommand `access update user`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NotFound	5	Entity (user, service, file, path or other) was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.

Diagnostics Commands

The `diags` commands allow you to manage the diagnostics tests.

[TABLE C-1](#) lists the `diags` subcommands.

Note – Every subcommand returns a return code upon completion.

Note – The diagnostics commands are also provided in the *Sun Fire V20z and Sun Fire V40z—Troubleshooting Techniques and Diagnostics Guide*.

TABLE C-1 Diagnostics Subcommands

Subcommand	Description
<code>diags cancel tests</code>	Cancels one or more diagnostics tests, resulting in the deletion of the results data.
<code>diags get modules</code>	Returns a list of test modules that are available. Queries the framework to obtain the test module information.
<code>diags get state</code>	Returns the state of the platform-diagnostics control server.
<code>diags get tests</code>	Returns data describing the diagnostics tests that are available and their requirements and parameters.
<code>diags run tests</code>	Submits one or more diagnostics tests for execution.
<code>diags start</code>	Starts the service processor (SP) and platform-diagnostics framework.
<code>diags terminate</code>	Terminates all diagnostics tests and terminates the diagnostics subsystem.

Diags Cancel Tests Subcommand

Description: Cancels one or more diagnostics tests, resulting in the deletion of results data.

Format

```
diags cancel tests [[{-t | --test} TEST HANDLE] | [{-a|--all}]
[{-H | --noheader}]]
```

[TABLE C-2](#) lists the arguments for this subcommand.

Note – Specifying no arguments cancels all tests for each device in the server.

TABLE C-2 Arguments for Subcommand `diags cancel tests`

Arguments	Description
{ -t --test }	Specifies the test to cancel. Note: The <code>TEST HANDLE</code> is the same <code>TEST HANDLE</code> that is output to the screen when you submit the test.
{-a --all}	Cancels all tests.
{-H --noheader}	Suppresses header output.

Return Codes

[TABLE C-3](#) lists the return codes for this subcommand.

TABLE C-3 Return Codes for Subcommand `diags cancel tests`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.

TABLE C-3 Return Codes for Subcommand `diags cancel tests` (Continued)

Return Code	ID	Description
NWSE_NoPermission	6	Not authorized to perform operation.
NWSE_MissingArgument	7	Missing argument(s).
NWSE_ServiceNotAvailable	24	Requested service is not available.

Diags Get Modules Subcommand

Description: Returns a list of test modules that are available. The `diags get modules` command queries the framework to obtain the test module information.

```
diags get modules [{-v|--verbose}]
```

Examples of successful output are:

```
diags get modules
Module
fan
flash
memory
```

```
diags get modules -v
Module Host Type
fan      SP
flash    SP
memory   PF
```

TABLE C-4 Arguments for `diags get modules`

Arguments	Description
[{ -v --verbose}]	Displays all columns in output.

TABLE C-5 Return Codes for `diags get modules`

Return Code	ID	Description
NWSE_RPCTimeout	0	Request was issued but not serviced by the server.
NWSE_RPCNotConnected	1	Unable to connect to the RPC server.

TABLE C-5 Return Codes for `diags get modules` (Continued)

Return Code	ID	Description
NWSE_NoPermission	6	Not authorized to perform operation.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_ServiceNotAvailable	24	Requested service is not available. CD diagnostics are running, user is trying to run diagnostics on the service processor.

Diags Get State Subcommand

Description: Returns the state of the platform-diagnostics control server.

Format

```
diags get state
```

If the result returned from the command is that the platform is up and ready for diagnostics, then you can submit platform diagnostics tests for execution.

The following success text messages might return:

```
SP Diagnostics is ready to accept tests. Run 'diags get state' to determine availability of Platform Diagnostics.
```

```
SP Diagnostics (in no-platform mode) is ready to accept tests.
```

```
Platform and SP Diagnostics are ready to accept tests.
```

The following error text messages might return:

```
Error. Verify that the platform state is 'off' and retry or use the 'diags start --forced' option to ignore the current state.
```

```
Error. Platform CD Diagnostics is currently running.
```

```
Error. Diagnostics is currently running. Run 'diags terminate' and try again.
```

```
Error. Unable to load Platform Diagnostics. Diagnostics terminated.
```

```
Error. Unable to load SP Diagnostics. Diagnostics terminated.
```

```
Error. SP no-platform Diagnostics is already running.
```

Return Codes

TABLE C-6 lists the return codes for this subcommand.

TABLE C-6 Return Codes for Subcommand `diags get state`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_ServiceNotAvailable	24	Requested service is not available.
NWSE_DeviceError	25	Unable to read or write to the device.

Diags Get Tests Subcommand

Description: Returns data describing the diagnostic tests that are available. This data includes the specific test name and the module to which the test applies.

Format

```
diags get tests [{-H | --noheader}] [{-D | --Delim} <DELIMITER>]  
[-v | --verbose]
```

Note – If the output for certain tests wraps, you can redirect the output to a file and view it with an editor for better readability.

TABLE C-7 lists the arguments for this subcommand.

TABLE C-7 Arguments for Subcommand `diags get tests`

Arguments	Description
<code>{-H --noheader}</code>	Suppresses header output.
<code>{-D --Delim}</code> <code><DELIMITER></code>	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.
<code>[-v --verbose]</code>	If specified, <code>Host Type, Services, and Devices</code> are displayed in addition to <code>Module and Testname</code> .

Return Codes

TABLE C-8 lists the return codes for this subcommand.

TABLE C-8 Return Codes for Subcommand `diags get tests`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_MissingArgument	7	Missing argument(s).
NWSE_ServiceNotAvailable	24	Requested service is not available.

Diags Run Tests Subcommand

Description: Submits one or more diagnostics tests for execution.

Format

```
diags run tests [ [{-n | --name} TEST_NAME ] [{-a | --all}]  
[-H | --noheader] [-P | --noprogess] [{-m | --module} MODULE_NAME]  
[-v | --verbose]
```

Note – If the output for certain tests wraps, you can redirect the output to a file and view it with an editor for better readability.

TABLE C-9 lists the arguments for this subcommand.

TABLE C-9 Arguments for Subcommand `diags run tests`

Arguments	Description
{ -n --name }	Specifies the specific test(s) to execute. Run <code>diags get tests</code> for a list of individual test names.
{ -a --all }	Specifies that all tests are to be executed. Run <code>diags get tests</code> for a list of all available tests. Specifying no arguments also runs all tests for each device in the server.
{ -H --noheader }	Suppresses header output.
{ -P --noprogess }	Suppresses progress dots when waiting for test results.
{ -m --module }	Specifies that only tests for the specified module are to be executed. Run <code>diags get tests</code> for a list of modules.
{ -v --verbose }	If specified, the <code>Test Details</code> is displayed following the test result line.

The following data displays after a test is run:

- Submitted Test Name
- Test Handle
- Test Result (for example: Passed, Failed)
- Details. If you specify the `-v` option, the `Test Details` are displayed, indicating detailed information about the test, such as high, low and nominal values, actual values, and so on. Upon failure, the `Failure Details` are displayed with a text message indicating the cause of failure.

Return Codes

[TABLE C-10](#) lists the return codes for this subcommand.

TABLE C-10 Return Codes for Subcommand `diags run tests`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.

TABLE C-10 Return Codes for Subcommand `diags run tests` (Continued)

Return Code	ID	Description
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_MissingArgument	7	Missing argument(s).
NWSE_ServiceNotAvailable	24	Requested service is not available.

Diags Start Subcommand

Description: Starts the SP and platform-diagnostics framework. You must execute this command before running any tests.

Format

```
diags start [--noplatform] [{-f|--forced}]
```

TABLE C-11 Arguments for Subcommand `diags start`

Arguments	Description
{--noplatform}	Specifies to start diagnostics on the SP from the NFS mount, without rebooting the platform into <code>diags</code> mode. Note: This option is not available when running diagnostics from CD-ROM.
{-f --forced}	Forces diagnostics to start.

After running this command, you can immediately run the SP tests.

This command reboots the platform into diagnostics mode. This process might take between two and three minutes to complete. You can begin to run diagnostics on the SP while the platform diagnostics are loading. However, before you run the platform diagnostics, wait approximately one minute for the diagnostic kernel to completely load on the platform.

To verify whether the diagnostics tests are available to run, run the subcommand `diags get state`. Refer to [“Diags Get State Subcommand” on page 144](#).

This subcommand returns one of the following states:

- **Success Text Message**—The Platform Diagnostics are up and are available to receive test requests.
- **Error Text Message**—The Platform Diagnostics are not up.

If the state returned from the command is that the platform is up and ready for diagnostics, then you can submit platform diagnostic tests for execution. You can optionally start diagnostics on the SP from the NFS mount without rebooting the platform into `diags` mode. This enables you to continue to run the production OS while you simultaneously perform SP diagnostics testing.

To do so, run the `diags start` subcommand with the following option:

```
diags start --no platform
```

The platform state must be either off or OS Communicating. Refer to the `platform get os state` subcommand for details about these states.

Return Codes

[TABLE C-12](#) lists the return codes for this subcommand.

TABLE C-12 Return Codes for Subcommand `diags start`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoRouteToHost	13	No route to host (network down). Occurs when CD Diagnostics is unable to send a packet to the SP on a specific port.
NWSE_InvalidOpForState	22	Invalid operation for current state.
NWSE_ServiceNotAvailable	24	Requested service is not available.

Diags Terminate Subcommand

Description: Terminates all diagnostics tests and the diagnostics session.

Format

```
diags terminate
```

Return Codes

[TABLE C-13](#) lists the return codes for this subcommand.

TABLE C-13 Return Codes for Subcommand `diags terminate`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_MissingArgument	7	Missing argument(s).
NWSE_NoRouteToHost	13	No route to host (network down). Occurs when CD Diagnostics is unable to send a packet to the SP on a specific port.
NWSE_ServiceNotAvailable	24	Requested service is not available.

Inventory Commands

The `inventory` command reports on the inventory of hardware and software for a Sun Fire V20z or Sun Fire V40z server.

[TABLE D-1](#) lists the `inventory` subcommands that you can use to retrieve specific information about hardware or software.

Note – Every subcommand returns a return code upon completion.

TABLE D-1 Inventory Subcommands

Subcommand	Description
<code>inventory compare versions</code>	Returns a list of all installed software packages and the version differences with those listed in a release manifest.
<code>inventory get hardware</code>	Returns detailed information for all field-replaceable hardware components.
<code>inventory get software</code>	Returns inventory information for all installed or uninstalled software.
<code>inventory get remote-software</code>	Returns a list of package versions available for download or installation from a running update server.
<code>inventory get all</code>	Returns detailed information for all hardware and software components.

Inventory Compare Versions Subcommand

Description: Returns a list of all installed software packages and the version differences with those listed in a release manifest or those available on a running update server. You can use this command to verify that your installation is consistent with a supported release and to determine the packages that have been updated in a new release.

Format

```
inventory compare versions
[{-f | --file} RELEASE_MANIFEST_FILE |
[{-i | --ipaddress} REMOTE_SERVER_IP]
[{-p | --port} REMOTE_SERVER_PORT]
[{-v | --verbose} [{-H | --noheader}]]
[{-D | --Delim <DELIMITER>}]
```

TABLE D-2 lists the arguments for this subcommand.

TABLE D-2 Arguments for Subcommand `inventory compare versions`

Argument	Description
{-f --file}	The file describing all of the packages and versions within a release of software. These files are at the root directory of an unzipped NSV file and are usually accessed via the share point at /mnt.
{-i --ipaddress}	The IP address of a running update server.
{-p --port}	The port number used by the update server.
{-v --verbose}	Displays additional information, including the path to the matching package on the NSV, the installed package description, and the matching manifest package description. This option is ignored when comparing to the update server (using the -i and -p options).
{ -H --noheader }	Suppresses column headings.
{ -D --Delim }	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

TABLE D-3 lists the return codes for this subcommand.

TABLE D-3 Return Codes for Subcommand `inventory compare versions`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NotFound	5	Error. The remote software inventory is not available.
NWSE_ServiceNotAvailable	24	Requested service is not available.

Inventory Get Hardware Subcommand

Description: Returns information for field-replaceable hardware components. Information returned includes name, device type, attributes, OEM, manufacture date, hardware revision, serial number, and part number for the component.

Format

```
inventory get hardware [{-v|--verbose}]  
[{-H|--noheader}] [{-D|--Delim}]
```

To obtain the board revision, run this command:

```
inventory get hardware -D : |grep Motherboard|awk -F : '{print $5}'
```

To obtain the PRS revision, run this command:

```
inventory get hardware -D : |grep PRS|awk -F : '{print $5}'
```

Note – You also can use the `sensor get` command to find this information.

The command output, without the `-v` argument, includes the following information, in columns from left to right:

- Name (of device)
- Type (of device)
- Attributes (miscellaneous information about the component such as CPU speed)
- OEM (manufacturer that distributes the part)
- Manufacture Date
- HW Revision (number)
- Serial #
- Part #

Below is an example of successful output for the command, without the `-v` argument. (Because of space limitations here, the Hardware Revision, Part #, Serial #, and Attributes columns are omitted.)

```
localhost # inventory get hardware
Name           Type           OEM           Manufacture Date
CPU 0 DIMM 0   memory        127f000000000000 2000-01-01
CPU 0 DIMM 1   memory        127f000000000000 2000-01-01
CPU 0 DIMM 2   memory        127f000000000000 2000-01-01
CPU 0 DIMM 3   memory        127f000000000000 2000-01-01
DDR 0 VRM      memvrm        NA
CPU 0          cpu           AuthenticAMD     NA
Family 15 Model 5 Stepping 1
CPU 0 VRM      vrm           NA
CPU 1 DIMM 0   memory        127f000000000000 2000-01-01
CPU 1 DIMM 1   memory        127f000000000000 2000-01-01
CPU 1 DIMM 2   memory        127f000000000000 2000-01-01
CPU 1 DIMM 3   memory        127f000000000000 2000-01-01
DDR 1 VRM      memvrm        NA
CPU 1          cpu           AuthenticAMD     NA
Family 15 Model 5 Stepping 1
CPU 1 VRM      vrm           NA
Motherbrd     planarS-SCI14312004-10-31
PRS Software  os            2005-03-16
SCSI backplane scsi_backplane NA
```

Additional column headings that appear in the output, if you use the `-v` argument, are:

- MfgAssy#
- MfgAssyRev
- FirmwareID
- FirmwareRev
- SoftwareRev
- SoftwareID
- Identifier

[TABLE D-4](#) lists the arguments for this subcommand.

TABLE D-4 Arguments for Subcommand `inventory get hardware`

Argument	Description
{ <code>-v</code> <code>--verbose</code> }	Displays all columns.
{ <code>-H</code> <code>--noheader</code> }	Suppresses column headings.
{ <code>-D</code> <code>--Delim</code> }	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

[TABLE D-5](#) lists the return codes for this subcommand.

TABLE D-5 Return Codes for Subcommand `inventory get hardware`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.

Inventory Get Software Subcommand

Description: Returns the inventory information for all installed or uninstalled software (located on the optional external file system).

Format

```
inventory get software [{-a | --all}][{-H | --noheader}]  
[{-D | --Delim <DELIMITER>}]
```

TABLE D-6 lists the arguments for this subcommand.

TABLE D-6 Arguments for Subcommand `inventory get software`

Argument	Description
{-a --all}	(Optional) Looks in the directory <code>/sw_images</code> on the service processor for software packages and uninstalled software.
{-H --noheader }	Suppresses column headings.
{-D --Delim }	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

TABLE D-7 lists the return codes for this subcommand.

TABLE D-7 Return Codes for Subcommand `inventory get software`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.

Inventory Get Remote-Software Subcommand

Description: Returns a list of package versions that are available for download or for installation from a running update server.

Format

```
inventory get remote-software [{-D|--Delim} DELIMITER]
[{-H|--noheader}] [{-i|--ipaddress} REMOTE_ADDRESS]
[{-p|--port} REMOTE_PORT]
```

[TABLE D-8](#) lists the arguments for this subcommand.

TABLE D-8 Arguments for Subcommand `inventory get remote-software`

Argument	Description
{-i --ipaddress}	The IP address of a running update server.
{-p --port}	The port number used by the update server.
{ -H --noheader }	Suppresses column headings.
{ -D --Delim }	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

[TABLE D-9](#) lists the return codes for this subcommand.

TABLE D-9 Return Codes for Subcommand `inventory get remote-software`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NotFound	5	Error. The remote software inventory is not available.
NWSE_ServiceNotAvailable	24	Requested service is not available.

Inventory Get All Subcommand

Description: Returns detailed information for all field-replaceable hardware components and all installed or uninstalled software.

Format

```
inventory get all {-a | --all} {-v | --verbose} [{-H | --noheader}]  
[{-D | --Delim <DELIMITER>}]
```

TABLE D-10 lists the arguments for this subcommand.

TABLE D-10 Arguments for Subcommand `inventory get all`

Argument	Description
{-a --all}	(Optional) Looks in the directory <code>/sw_images</code> on the Service Processor for software packages and uninstalled software.
{ -v --verbose }	Displays all columns.
{ -H --noheader }	Suppresses column headings.
{ -D --Delim }	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

TABLE D-11 lists the return codes for this subcommand.

TABLE D-11 Return Codes for Subcommand `inventory get all`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.

IPMI Commands

The `ipmi` command manages the Intelligent Platform Management Interface (IPMI) functions.

[TABLE E-1](#) lists the `ipmi` subcommands.

Note – Every subcommand returns a return code upon completion.

TABLE E-1 IPMI Subcommands

Subcommand	Description
<code>ipmi disable channel</code>	Disables one of two IPMI channels.
<code>ipmi enable channel</code>	Enables one of two IPMI channels.
<code>ipmi disable pef</code>	Disables platform-event filtering.
<code>ipmi enable pef</code>	Enables platform-event filtering.
<code>ipmi get channels</code>	Displays the list of IPMI channels and whether they are enabled or disabled.
<code>ipmi get global enables</code>	Displays the list of IPMI global enables and their current value.
<code>ipmi set global enable</code>	Sets the value of several IPMI global enable variables.
<code>ipmi get sel</code>	Displays the system event log (SEL) items in raw format.
<code>ipmi clear sel</code>	Clears the system event log (SEL).
<code>ipmi reset</code>	Resets IPMI information back to default factory settings.

IPMI Disable Channel Subcommand

Description: Allows you to disable one of two IPMI channels.

Format

```
ipmi disable channel {sms | lan}
```

[TABLE E-2](#) lists the arguments for this subcommand.

TABLE E-2 Arguments for Subcommand `ipmi disable channel`

Argument	Description
sms	The ID of the channel to disable for the system interface; not case-sensitive.
lan	The ID of the channel to disable for the LAN interface; not case-sensitive.

Return Codes

[TABLE E-3](#) lists the arguments for this subcommand.

TABLE E-3 Return Codes for Subcommand `ipmi disable channel`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NoPermission	6	Not authorized to perform this operation.

IPMI Enable Channel Subcommand

Description: Allows you to enable one of two IPMI channels.

Format

```
ipmi enable channel {sms | lan}
```

[TABLE E-4](#) lists the arguments for this subcommand.

TABLE E-4 Arguments for Subcommand `ipmi enable channel`

Argument	Description
sms	The ID of the channel to enable for the system interface; not case-sensitive.
lan	The ID of the channel to enable for the LAN interface; not case-sensitive. If you are activating the LAN channel for the first time, you are prompted for a password to associate with the null user.

Return Codes

[TABLE E-5](#) lists the return codes for this subcommand

TABLE E-5 Return Codes for Subcommand `ipmi enable channel`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_ServiceNotAvailable	24	Requested service is not available.

IPMI Disable PEF Subcommand

Description: Allows you to disable platform-event filtering (PEF).

Format

```
ipmi disable pef
```

Return Codes

[TABLE E-6](#) lists the return codes for this subcommand.

TABLE E-6 Return Codes for Subcommand `ipmi disable pef`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_NoPermission	6	Not authorized to perform this operation.

IPMI Enable PEF Subcommand

Description: Allows you to enable platform-event filtering (PEF).

Format

```
ipmi enable pef
```

Return Codes

[TABLE E-7](#) lists the return codes for this subcommand.

TABLE E-7 Return Codes for Subcommand `ipmi enable pef`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_ServiceNotAvailable	24	Requested service is not available.

IPMI Get Channels Subcommand

Description: Displays the list of IPMI channels and whether each channel is enabled or disabled.

Format

```
ipmi get channels
```

Return Codes

[TABLE E-8](#) lists the return codes for this subcommand.

TABLE E-8 Return Codes for Subcommand `ipmi get channels`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.

IPMI Get Global Enables Subcommand

Description: Displays the list of IPMI global enables and the current value of each enable.

Format

```
ipmi get global enables
```

Return Codes

[TABLE E-9](#) lists the return codes for this subcommand.

TABLE E-9 Return Codes for Subcommand `ipmi get global enables`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.

IPMI Get Sel Subcommand

Description: Displays the list of system event log items, in raw format.

Format

```
ipmi get sel
```

The command output returns the following information, in columns from left to right:

- Record ID
- Record Type
- Timestamp
- Generator ID
- Format Version
- Sensor Type
- Sensor #

- Event Dir/Type Event
- Data

Below is an example of successful output for the command. (Because of space limitations here, the Timestamp column, which is the third column from the left, is omitted.)

```
localhost # ipmi get sel
Record          RecordGeneratorFormatSensorSensorEventData
ID              TypeID Ver Type#   Dir/
0001            0x020x00200x040x100xfc0x6f02ffff
0002            0x020x00000x040x510x060x0708090a
0003            0x020x00000x040x510x060x0708090a
0004            0x020x00000x040x510x060x0708090a
0005            0x020x00200x040x120x270x6fc500ff
0006            0x020x00200x040x120x270x6fc580ff
0007            0x020x00200x040x040x310x01520121

localhost # ipmi get sel -H
0001            0x020x00200x040x100xfc0x6f02ffff
0002            0x020x00000x040x510x060x0708090a
0003            0x020x00000x040x510x060x0708090a
0004            0x020x00000x040x510x060x0708090a
0005            0x020x00200x040x120x270x6fc500ff
0006            0x020x00200x040x120x270x6fc580ff
0007            0x020x00200x040x040x310x01520121

localhost # ipmi get sel -H -D ,
0001,0x02,02/28/2005 18:53:17,0x0020,0x04,0x10,0xfc,0x6f,02ffff
0002,0x02,02/28/2005 18:53:17,0x0000,0x04,0x51,0x06,0x07,08090a
0003,0x02,02/28/2005 18:53:17,0x0000,0x04,0x51,0x06,0x07,08090a
0004,0x02,02/28/2005 18:53:17,0x0000,0x04,0x51,0x06,0x07,08090a
0005,0x02,02/28/2005 18:53:17,0x0020,0x04,0x12,0x27,0x6f,c500ff
0006,0x02,06/14/1906 21:02:57,0x0020,0x04,0x12,0x27,0x6f,c580ff
0007,0x02,06/15/1906 00:00:05,0x0020,0x04,0x04,0x31,0x01,520121
```

TABLE E-10 Arguments for the Subcommand `ipmi get sel`

Argument	Description
<code>[-H --noheader]</code>	Suppresses column headings
<code>[-D --Delim]</code>	Specifies a different field separator.

TABLE E-11 Return Codes for Subcommand `ipmi get sel`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command completed successfully.
<code>NWSE_Busy</code>	9	Device or resource is busy.

IPMI Clear Sel Subcommand

Description: Clears the system event log.

```
ipmi clear sel
```

Successful output is:

```
localhost # ipmi clear sel
```

TABLE E-12 Return Codes for Subcommand `ipmi clear sel`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command completed successfully.
<code>NWSE_Busy</code>	9	Unable to reserve SEL.
<code>NWSE_NotRecognized</code>	20	Request not recognized or understood.
<code>NWSE_DeviceError</code>	25	Unable to access SEL information.

IPMI Set Global Enable Subcommand

Description: Allows you to set the value of several IPMI global-enable variables.

Format

```
ipmi set global enable {-n |--name} GLOBAL_NAME {{-e|--enabled} |  
{-d|--disabled}}
```

[TABLE E-13](#) lists the arguments for this subcommand.

[TABLE E-14](#) provides information about the aliases.

TABLE E-13 Arguments for Subcommand `ipmi set global enable`

Argument	Description
{-n --name}	The name of one of the IPMI global enable variables; see TABLE E-14 . You can use either a quoted long string or an alias without quotes for the list of global enables.
{-e --enabled}	Turns the channel on.
{-d --disabled}	Turns the channel off.

TABLE E-14 IPMI Global-Enabled Aliases

Alias	Name String	Values	Default
oem0	OEM0 Enable	Enabled/ Disabled	Disabled
oem1	OEM1 Enable	Enabled/ Disabled	Disabled
oem2	OEM 2 Enable	Enabled/ Disabled	
logging	Enable System Event Logging	Enabled/ Disabled	Enabled

TABLE E-14 IPMI Global-Enabled Aliases (*Continued*)

Alias	Name String	Values	Default
msg_buf	Enable Event Message Buffer	Enabled/ Disabled	
msg_buf_interrupt	Enable the Event Message Buffer Full	Enabled/ Disabled	
msg_queue_interrupt	Enable Receive Message Queue Interrupt	Enabled/ Disabled	Enabled

Return Codes

[TABLE E-15](#) lists the return codes for this subcommand.

TABLE E-15 Return Codes for Subcommand `ipmi set global enable`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NoPermission	6	Not authorized to perform this operation.

IPMI Reset Subcommand

Description: Resets IPMI information back to default factory settings.

Format

```
ipmi reset {-s | --sdr} {-c | --config} {-p | --password} {-a | --all}
```

[TABLE E-16](#) lists the arguments for this subcommand.

TABLE E-16 Arguments for Subcommand `ipmi reset`

Argument	Description
{-s --sdr} }	Copies the original database file to <code>pstore</code> .
{-c --config} }	Deletes the configuration file and global enables.
{-p --password} }	Deletes the password file.
{-a --all} }	Performs the functions of all the parameters.

Return Codes

[TABLE E-17](#) lists the return codes for this subcommand.

TABLE E-17 Return Codes for Subcommand `ipmi reset`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.
<code>NWSE_NoPermission</code>	6	Not authorized to perform this operation.

Platform Commands

The `platform` command reports or changes some aspect of the state of the platform.

[TABLE F-1](#) lists the platform subcommands.

Note – Every subcommand returns a return code upon completion.

TABLE F-1 Platform Subcommand

Subcommand	Description
<code>platform console</code>	Manages access to the platform serial console.
<code>platform os state</code>	Manages the current state of the operating system (OS).
<code>platform power state</code>	Manages the state of the platform power.
<code>platform get hostname</code>	Displays the host name of the current primary platform.
<code>platform get mac</code>	Returns the MAC addresses for the two on-board platform NICs.
<code>platform get product id</code>	Displays the product ID for the current system.

Platform Console Subcommands

The subcommands listed in [TABLE F-2](#) allow you to manage access to the platform serial console.

TABLE F-2 Platform Console Subcommands

Subcommand	Description
<code>platform console</code>	Provides access to the platform serial console.
<code>platform get console</code>	Retrieves the configuration of the service processor (SP) access to the platform serial console.
<code>platform set console</code>	Configures the SP access to the platform serial console.

Platform Console Subcommand

Description: For remote-management capability, this command provides access to the platform serial console. Used in conjunction with the subcommand `platform set console` and the appropriate BIOS and platform OS settings, this command enables you to view the platform serial console while logged in to the SP.

Format

```
platform console
```

You must configure the BIOS settings using the BIOS Setup utility. To refresh the BIOS Setup screen, press the Control-R keys. Choose the Advanced tab to set the configuration.

[TABLE F-3](#) lists common COM1 values. [TABLE F-4](#) lists common values for console redirection.

TABLE F-3 Common COM1 Values

I/O Device Configuration	Value
Serial port A	Enabled
Base I/O address	3F8
Interrupt	IRQ 4

TABLE F-4 Common Values for Console Redirection

Console Redirection	Value
Com Port Address	On-board COM A
Console connection	Direct
Baud Rate	19.2K
Flow Control	None
Console Type	ANSI

Note – You can change these values, as long as they are the same as serial-port values for the operating system (OS). If your operating system supports the COM2-4 values, you can set these for the BIOS settings.

The serial-console settings in the platform OS should be set to match the BIOS settings.

Enter the following while you are connected to the console:

```
^Ec character
```

Where ^E represents the Control-E keys and *character* is one of the entries in [TABLE F-5](#).

TABLE F-5 Serial-Console Values

Character	Function
.	Disconnects an attach read/write.
b	Sends a broadcast message.
c	Toggles flow control.
d	Takes down a console.
e	Changes the escape sequence.
f	Forces an attach read/write.
g	Groups information.
i	Dumps information.
L	Toggles logging on/off.
l?	Breaks the sequence list.

Note: The first character is a lowercase L.

TABLE F-5 Serial-Console Values (*Continued*)

Character	Function
l0	Sends a break per configuration file. Note: The first character is a lowercase L.
l1-9	Sends a specific break sequence. Note: The first character is a lowercase L.
o	Re-opens the tty and log file.
p	Replays the last 60 lines.
r	Replays the last 20 lines.
s	Activates spy read only.
u	Shows the host status.
v	Shows the version information.
w	Shows who is logged on to this console.
x	Shows the console baud information.
z	Suspends the connection.
<CR>	Ignores/aborts the command.
?	Prints this message.
^R	Replays the last line.
\ooo	Sends the character by octal code.

Under certain circumstances, it might be necessary to send a serial-break sequence to the platform OS (for example, to simulate the SysRq key when CONFIG_MAGIC_SYSRQ is defined and enabled in a Linux kernel).

To perform this operation, use the following sequence:

```
^Ec10
```

Where ^Ec10 represents the Control-E keys, followed by the lowercase letter "C," the lowercase letter "L," and the numeral "0".)

The platform console command responds by displaying the string [halt sent], confirming that the break sequence has been generated.

In the event that console output becomes corrupted, ^Ecd ^Eco usually restores proper operation. This problem is normally due to flow-control issues.

Example

The following example lists the steps you would perform to enable and run the `platform console` command:

1. Check or set the BIOS settings.

2. Run the command:

```
platform set console -s sp -S 19200 -e
```

3. Run the command:

```
platform set console
```

Return Codes

[TABLE F-6](#) lists the return codes for this subcommand.

TABLE F-6 Return Codes for Subcommand `platform console`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.

Platform Get Console Subcommand

Description: Retrieves the configuration information regarding the SP access to the platform serial console.

Format

```
platform get console [{-H|--noheader}] [{-D | --Delim <DELIMITER>}]
```

[TABLE F-7](#) lists the arguments for this subcommand.

TABLE F-7 Arguments for Subcommand `platform get console`

Argument	Description
<code>{-H --noheader}</code>	Suppresses column headers.
<code>{ -D --Delim }</code>	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

The following list shows examples of the output from the successful execution of this command.

- Platform serial port directed to rear panel:

```
Rear Panel
Platform COMA
```

- Platform serial port directed to SP; SP serial port directed to rear panel, platform console disabled:

```
Rear PanelConsole Redirection
SP ConsoleDisabled
```

- Platform console enabled:

```
Rear PanelConsole Redirection
SpeedPruningLog Trigger
19200No244 KB
SP ConsoleEnabled
```

If the external serial port is not connected to the platform and is connected to the SP console, you can access the platform serial console using the `platform console` subcommand.

[TABLE F-8](#) lists the information that is displayed, depending on whether the rear-panel serial port is connected to the platform or to the SP.

TABLE F-8 Displayed Data for Subcommand `platform get console`

Column	Description
Enabled	Displays <code>No</code> if the external serial port is connected to the platform. Otherwise, the external serial port is connected to the SP console; you can access the platform serial console through the SP command line by running the subcommand <code>platform console</code> .
Speed	Indicates the communications speed of the link.
Prune	Indicates whether ANSI escape code and duplicate information pruning is enabled.
Log Trigger	Indicates the approximate size at which log rotation occurs (for example, when the file <code>console.0</code> is removed, the current log is moved to <code>console.0</code> and a new log file is opened). Pruning of log-file contents happens only when rotation occurs. The minimum size for a log file is 64 KB; the maximum size is 1024 KB.

Return Codes

TABLE F-9 lists the return codes for this subcommand.

TABLE F-9 Return Codes for Subcommand `platform get console`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoPermission	6	Not authorized to perform this operation.

Platform Set Console Subcommand

Description: Enables you to configure access of the SP to the platform serial console, set the speed of the connection, and limit the size of the log files created.

Format

Two options are available:

- Configure the external serial port so that it is connected to the platform serial console (default configuration).
- Configure the external serial port so that it is connected to the Service Processor serial console.

The default configuration of the external serial port connects it to the platform serial console. Use the following syntax for the default configuration:

```
platform set console [--serial|-s] platform
```

Use the following syntax to configure the external serial port so that it connects to the SP serial console. With this configuration, you can access the platform serial console through the command line of the SP by running the `platform console` subcommand.

```
platform set console [--serial|-s] sp
[ [--enable|-e] | [--disable|-d] ]
[ [ --prune|-p ] | [ --noprune|-n ] ]
[ --speed|-S ] {1200|2400|4800|9600|19200|38400|115200}
[ --log|-l ] size
```

TABLE F-10 lists the arguments for this subcommand.

Note – If `-s` is set to `platform`, none of the following arguments can be used.

TABLE F-10 Arguments for Subcommand `platform set console`

Argument	Description
{-S --speed} {1200 2400 4800 9600 19200 38400 115200}	Selects the port speed for the platform console. BIOS, the platform OS, and the console must all be configured for the same speed.
{-d --disable}	Indicates that the platform console monitor is inactive. Cannot be used with: <code>-e</code> .
{-e --enable}	Indicates that the platform console monitor is active. Cannot be used with: <code>-d</code> .
{-l --log} size	Select the trigger size in KB for console log rotation. The acceptable values for log size are between 64 and 1024 inclusive.
{-n --noprune}	Indicates that the platform console log should be the raw console data. Cannot be used with: <code>-p</code> .
{-p --prune}	Indicates that the platform console log is to be cleaned of ANSI sequences and pruned of duplicated information. Cannot be used with: <code>-n</code> .
{-s --serial} {sp platform}	Specifies whether the serial port is connected to the platform COMA port, or the SP serial console. Cannot be used with: <code>-e [platform]</code> <code>-d [platform]</code> <code>-p [platform]</code> <code>-n [platform]</code> <code>-S [platform]</code> <code>-l [platform]</code> .

Return Codes

[TABLE F-11](#) lists the return codes for this subcommand.

TABLE F-11 Return Codes for Subcommand `platform set console`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoMemory	8	Insufficient memory.
NWSE_DeviceError	25	Unable to read or write to the device.

Platform OS State Subcommands

The subcommands listed in [TABLE F-12](#) enable you to manage the operating system (OS).

TABLE F-12 Platform OS State Subcommands

Subcommand	Description
<code>platform get os state</code>	Retrieves the current state of the platform OS (for example, running, booting, off and so on).
<code>platform set os state reboot</code>	Reboots the platform into the default OS, BIOS Setup or BIOS Update, or shuts down the platform.
<code>platform set os state boot</code>	Serves as an alias for the subcommand <code>platform set os state reboot</code> and only functions when the platform power state is off.
<code>platform set os state shutdown</code>	Shuts down the platform.
<code>platform set os state update bios</code>	Allows updating the platform BIOS with a new local or remote BIOS image file.

If the platform is off, the subcommand `platform set os state reboot` causes the platform to turn on and boot the OS. If the platform is already running, this command reboots the OS. The subcommand `platform set os state reboot` waits for the platform to boot.

The subcommand `platform set power state` ensures that the platform is running. It will not affect the platform if it is running; if the platform is off, it will power on and boot the OS. The subcommand `platform set power state` waits only for the power to come on. (Refer to [“Platform Power State Subcommands” on page 187.](#))

Platform Get OS State Subcommand

Description: Retrieves the current state of the platform OS.

Format

```
platform get os state
```

The values for the current state include:

- Off
- On
- Communicating
- Diagnostics
- Sleeping
- BIOS booting
- BIOS setup
- OS booting
- OS shutting down

When the platform is in the Communicating state (in which the OS is communicating with the SP), if the platform drivers are uninstalled, the SP remains in the Communicating state even though it can no longer communicate with the platform.

Refer to [“Platform Set OS State Subcommands” on page 181](#) for more information about setting the state.

Return Codes

TABLE F-13 lists the return codes for this subcommand.

TABLE F-13 Return Codes for Subcommand `platform get os state`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request might or might not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoPermission	6	Not authorized to perform this operation.

Platform Set OS State Subcommands

This group of subcommands provides the ability to reboot the platform into the default OS, BIOS Setup, or BIOS Update, or to shut down the platform. Rebooting to BIOS Setup enables you to configure the BIOS parameters while BIOS Update enables you to reflash the BIOS image.

In order to shut down the platform, the platform drivers must be installed (unless you use the `-f` argument). Using this subcommand, you can smoothly shut down the platform and allow the OS to shut off power.

Platform Set OS State Reboot

Description: Enables you to reboot the platform. If the platform is running, this subcommand reboots the OS.

Format

```
platform set os state reboot [{-W | --nowait}]  
[{-b | --bios}] [{-d | --device}][{-f|--forced}] [-q | --quiet]
```

TABLE F-14 lists the arguments for this subcommand.

TABLE F-14 Arguments for Subcommand `platform set os state reboot`

Argument	Description
<code>[-W --nowait]</code>	If specified, the subcommand returns immediately instead of waiting for the operation to complete.
<code>[-b --bios]</code>	Returns to BIOS Setup. Allows you to change BIOS settings. Cannot be used with <code>-d</code> .
<code>[-d --device]</code>	Causes the BIOS to first attempt to use the specified device as a boot device, before returning to the configured BIOS boot order. Currently, the only supported device argument is <code>network</code> . Specifying <code>-device network</code> will cause the BIOS to attempt a network boot via the Preboot Execution Environment (PXE). Cannot be used with the <code>-b</code> argument.
<code>{-f --forced}</code>	Results in a hard power off. It either forces the power off, or resets the server: <ul style="list-style-type: none"> • After a time-out of several minutes, if the platform has not responded, or • Immediately, if the platform is not in the running OS state (no drivers have been installed or the server has crashed).
<code>[-q --quiet]</code>	Suppresses interactive warning messages. No error messages are blocked.

Return Codes

TABLE F-15 lists the return codes for this subcommand.

TABLE F-15 Return Codes for Subcommand `platform set os state reboot`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.
<code>NWSE_RPCTimeout</code>	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
<code>NWSE_RPCNotConnected</code>	3	Unable to connect to the RPC server.
<code>NWSE_NoPermission</code>	6	Not authorized to perform this operation.
<code>NWSE_Busy</code>	9	Device or resource is busy.
<code>NWSE_FileError</code>	18	File open, file missing, or a read or write error occurred.
<code>NWSE_InvalidOpForState</code>	22	Invalid operation for current state.

Platform Set OS State Boot

Description: Serves as an alias for the subcommand `platform set os state reboot`. It functions only when the platform power state is off.

Format

```
platform set os state boot [{-W | --nowait}]  
[{-b | --bios}] [{-d|--device}] [{-f|--forced}] [-q | --quiet]
```

TABLE F-16 lists the arguments for this subcommand.

TABLE F-16 Arguments for Subcommand `platform set os state boot`

Argument	Description
<code>[-W --nowait]</code>	If specified, the subcommand returns immediately instead of waiting for the operation to complete.
<code>[-b --bios]</code>	Boots to BIOS Setup instead of the OS. Cannot be used with <code>-d</code> .
<code>[-d --device]</code>	Causes the BIOS to first attempt to use the specified device as a boot device, before returning to the configured BIOS boot order. Currently, the only supported device argument is network. Specifying “ <code>-device network</code> ” will cause the BIOS to attempt a network boot via the Preboot Execution Environment (PXE). Cannot be used with the <code>-b</code> argument.
<code>{-f --forced}</code>	Results in a hard power off. This option is ignored
<code>[-q --quiet]</code>	Suppresses interactive warning messages. No error messages are blocked.

Return Codes

TABLE F-17 lists the return codes for this subcommand.

TABLE F-17 Return Codes for Subcommand `platform set os state boot`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.
<code>NWSE_RPCTimeout</code>	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
<code>NWSE_RPCNotConnected</code>	3	Unable to connect to the RPC server.

TABLE F-17 Return Codes for Subcommand `platform set os state boot`

Return Code	ID	Description
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_Busy	9	Device or resource is busy.
NWSE_InvalidOpForState	22	Invalid operation for current state.

Platform Set OS State Shutdown

Description: Enables you to shut down the platform. This action requires that the platform drivers be installed, unless you execute this subcommand with the argument `-f`.

Format

```
platform set os state shutdown [{-W | --nowait}]
[{-f|--forced}] [-q | --quiet]
```

[TABLE F-18](#) lists the arguments for this subcommand.

TABLE F-18 Arguments for Subcommand `platform set os state shutdown`

Argument	Description
<code>[-W --nowait]</code>	If specified, the subcommand returns immediately instead of waiting for the operation to complete.
<code>{-f --forced}</code>	Results in a hard power off. It either forces the power off, or resets the server: <ul style="list-style-type: none"> • After a time-out of several minutes, if the platform has not responded, or • Immediately, if the platform is not in the running OS state (no drivers have been installed or the server has crashed).
<code>[-q --quiet]</code>	Suppresses interactive warning messages. No error messages are blocked.

Return Codes

TABLE F-19 lists the return codes for this subcommand.

TABLE F-19 Return Codes for Subcommand `platform set os state shutdown`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_Busy	9	Device or resource is busy.
NWSE_FileError	18	File open, file missing, or a read or write error occurred.
NWSE_InvalidOpForState	22	Invalid operation for current state.

Platform Set OS State Update-BIOS

Description: Enables you to update the platform BIOS with a new local or remote BIOS image file.

Format

```
platform set os state update-bios {-f | --filename} BIOS IMAGE  
{-i | --ipaddress} REMOTE_ADDRESS {-r | --remote} REMOTE_VERSION  
[{-p | --port} REMOTE_PORT] [{-W | --nowait}] [{-q | --quiet}]
```

TABLE F-20 lists the arguments for this subcommand.

TABLE F-20 Arguments for Subcommand `platform set os state update-bios`

Argument	Description
<code>{-f --filename}</code>	Indicates the name of the file containing the new BIOS image to be used for updating the BIOS.
<code>{-i --ipaddress}</code>	The IP address of the server on which the update server (Java application) is running.
<code>{-r --remote}</code>	Specifies a version (for example, V1.2.3.4) or use LATEST to update using the latest version available on the update server.

TABLE F-20 Arguments for Subcommand `platform set os state update-bios`

Argument	Description
<code>{-p --port}</code>	Optional. The port number on the remote server on which the <code>java sp update</code> program is listening for SP flash update requests. If the port number is not provided, the command tries to connect to the default port. The default port number is 52708.
<code>[-W --nowait]</code>	If specified, the subcommand returns immediately instead of waiting for the operation to complete.
<code>[-q --quiet]</code>	Suppresses interactive warning messages. No error messages are blocked.

If the platform is off, the subcommand `platform set os state reboot` causes the platform to turn on and boot the OS. If the platform is already running, this command reboots the OS. The subcommand `platform set os state reboot` waits for the platform to boot.

The subcommand `platform set power state` ensures that the platform is running. It will not affect the platform if it is running; if the platform is off, it will power on and boot the OS. The subcommand `platform set power state` waits only for the power to come on. (Refer to [“Platform Power State Subcommands” on page 187.](#))

Return Codes

[TABLE F-21](#) lists the return codes for this subcommand.

TABLE F-21 Return Codes for Subcommand `platform set os state update-bios`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.
<code>NWSE_RPCTimeout</code>	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
<code>NWSE_RPCNotConnected</code>	3	Unable to connect to the RPC server.
<code>NWSE_NotFound</code>	5	Entity (user, service, file, path or other) was not found.
<code>NWSE_NoPermission</code>	6	Not authorized to perform this operation.

TABLE F-21 Return Codes for Subcommand `platform set os state update-bios`

Return Code	ID	Description
<code>NWSE_FileError</code>	18	File open, file missing, or a read or write error occurred.
<code>NWSE_InvalidOpForState</code>	22	Invalid operation for current state.
<code>NWSE_ServiceNotAvailable</code>	24	Requested service is not available.

Platform Power State Subcommands

The subcommands listed in [TABLE F-22](#) enable you to manage the platform power.

TABLE F-22 Platform Power State Subcommands

Subcommand	Description
<code>platform get power state</code>	Provides the ability to determine the platform power state (for example, whether it is on or off).
<code>platform set power state</code>	Provides the ability to turn the platform power on or off.

The subcommand `platform set power state` does not affect the platform if the platform is already on; if the platform is off, it powers on and boots the OS. In other words, the subcommand `platform set power state` ensures that the platform is on, but does not reboot the platform if it is not on.

The subcommand `platform set os state` waits for the platform to boot; the subcommand `platform set power state` only waits for the power to come on.

Platform Get Power State Subcommand

Description: Provides the ability to determine the platform power state from within a script (whether the platform is on or off).

Format

```
platform get power state
```

Return Codes

[TABLE F-23](#) lists the return codes for this subcommand.

TABLE F-23 Return codes for Subcommand `platform get power state`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoPermission	6	Not authorized to perform this operation.

Platform Set Power State Subcommand

Description: Enables you to turn the platform power on or off from within a script. This subcommand does not notify the platform OS of the request through the supplied channels.

The subcommand `platform set power state` either removes power abruptly, or forces the platform into a panic shutdown.

It is the same as pressing the Power button for less than one second or for more than five seconds (`-f` argument).

Note – Equally effective, less-destructive commands are available. If the platform drivers are installed, use the subcommand `platform set os state shutdown` to shut down the server gracefully. For more information, see [“Platform Set OS State Shutdown”](#) on page 184.

Format

```
platform set power state [{-W|--nowait}] [{-f|--forced}]  
[{-t|--timeout} TIME] {off|on|cycle}
```

[TABLE F-24](#) lists the arguments for this subcommand.

TABLE F-24 Arguments for Subcommand `platform set power state`

Argument	Description
{-W --nowait}	If specified, the command returns immediately instead of waiting for the operation to complete.
{-f --forced}	Results in a hard power off.
{-t --timeout}	Specifies the maximum time to wait for the operation to complete (in seconds).
{off on cycle}	Specifies whether to turn the platform power on or off or to cycle. Specifying the <code>cycle</code> argument causes platform power to be turned off, then on.

If the platform is off, the subcommand `platform set os state reboot` causes the platform to turn on and boot the OS. If the platform is already running, this command reboots the OS. The subcommand `platform set os state reboot` waits for the platform to boot. (Refer to [“Platform Set OS State Subcommands” on page 181](#).)

The subcommand `platform set power state` ensures that the platform is running. It will not affect the platform if the platform is running; if the platform is off, it will power on and boot the OS. The subcommand `platform set power state` waits only for the power to come on.

Return Codes

[TABLE F-25](#) lists the return codes for this subcommand.

TABLE F-25 Return Codes for Subcommand `platform set power state`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.

TABLE F-25 Return Codes for Subcommand `platform set power state` (Continued)

Return Code	ID	Description
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_MissingArgument	7	Missing argument(s).
NWSE_TimedOut	23	Operation timed out.

Platform Get Hostname Subcommand

Description: Displays the host name of the current primary platform. The data is refreshed only when the platform is rebooted.

Format

```
platform get hostname [{-H|--noheader}]
```

[TABLE F-26](#) lists the argument for this subcommand.

TABLE F-26 Argument for Subcommand `platform get hostname`

Argument	Description
{-H --noheader}	Suppresses column headers.

Return Codes

[TABLE F-27](#) lists the return codes for this subcommand.

TABLE F-27 Return Codes for Subcommand `platform get hostname`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_NoMemory	8	Insufficient memory.
NWSE_Busy	9	Device or resource is busy.
NWSE_RPCConnected	11	RPC client already connected.

TABLE F-27 Return Codes for Subcommand `platform get hostname` (Continued)

Return Code	ID	Description
NWSE_RPCConnRefused	12	RPC connection refused.
NWSE_NoRouteToHost	13	No route to host (network down).
NWSE_HostDown	14	Host is down.

Platform Get MAC Subcommand

Description: Returns the MAC addresses for the two on-board platform network interface cards (NICs).

Format

```
platform get mac
```

Return Codes

[TABLE F-27](#) lists the return codes for this subcommand.

TABLE F-28 Return Codes for Subcommand `platform get mac`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_ServiceNotAvailable	24	Requested service is not available.

Platform Get Product-ID Subcommand

Description: Displays the product ID for the current system.

Format

```
platform get product-id
```

Note – You can also retrieve the product ID, board revision number, and Power and Reset Sequencer (PRS) revision number by running the subcommands `sensor get` and `inventory get hardware`.

Return Codes

[TABLE F-29](#) lists the return codes for this subcommand.

TABLE F-29 Return Codes for Subcommand `platform get product-id`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_NotFound	5	Entity (user, service, file, path or other) was not found.
NWSE_FileError	18	File open, file missing, or a read or write error occurred.
NWSE_ServiceNotAvailable	24	Requested service is not available.

Sensor Commands

The `sensor` command reports or sets the value of an environmental sensor or control.

[TABLE G-1](#) lists the `sensor` subcommands.

Note – Every subcommand returns a return code upon completion.

Sensor Subcommands

The subcommands listed in [TABLE G-1](#) allow you to monitor components and to facilitate the proper operation of the software.

TABLE G-1 Sensor Subcommands

Subcommand	Description
<code>sensor get</code>	Returns all data associated with a sensor.
<code>sensor set</code>	Sets some of the data associated with a specific sensor or a class of sensors.

Note – There are some sensors whose values do not change, some that are there to provide information in the event of a problem, and others to facilitate the proper operation of the software.

Many of these sensors do not have a related component (parent) associated with them. For example, the die-temperature sensor for a CPU has the CPU as its parent component and a fan-speed sensor has the fan as its parent component; the product-id sensor, however, only reports a static value and has no parent relationship.

This relationship establishes the component(s) that is affected by changes in the value of the sensor. You cannot modify the thresholds for sensors that do not have a parent relationship, since an event will never occur for these threshold crossings.

Sensor Get Subcommand

Description: Returns all data associated with a sensor.

By default, only the sensor ID and its current value are displayed. You can specify on the command line the order of the data output.

Note – The *identifier* field is always displayed first, unless you suppress it with the `-I` option.

Format

```
sensor get [{"-i | --id} ID | {"-t | --type} TYPE_ID}]
[{"-v | --value}] [{"-n | --nominal}]
[{"-C | --crithigh}] [{"-c | --critlow}]
[{"-W | --warnhigh}] [{"-w | --warnlow}]
[{"-N | --name}] [{"-d | --description}]
[{"-S | --sensor-type}] [{"-p | --parent-comp}]
[{"-s | --severity}] | [{"--verbose}]
[{"-I | --noid}] [{"-H | noheader}]
[{"-D | --Delim <DELIMITER>}]
```

TABLE G-2 lists the arguments for this subcommand.

TABLE G-2 Arguments for Subcommand `sensor get`

Argument	Description
{-i --id}	<p>SENSOR_ID, PRODUCT-ID, BOARD-REVISION, PRS-REVISION</p> <p>Specifies the sensor for which the data is desired. You can specify this argument multiple times, in which case the sensor data is reported in the order specified.</p> <p>You can also retrieve the product ID, board-revision number, and Power and Reset Sequencer (PRS) revision number using this flag. Specify [-vIH] following the ID to convert the output to the appropriate product ID.</p> <p>For example, product ID 255 indicates the 2100 server and product ID 239 indicates the 4300 server. You can also obtain this information using the <code>inventory get hardware</code> command.</p>
{-t --type}	<p>Specifies the sensor class for which the data is desired. You can specify this argument multiple times, in which case the sensor output is grouped by type in the order specified. Current sensor classes are voltage, fan, temperature, current, power, and switch.</p>
{-v --value}	<p>Displays the current value of the sensor.</p>
{-n --nominal}	<p>Displays the nominal value of the sensor.</p>
{-C --crithigh}	<p>Displays the <i>critical high</i> threshold value for the sensor. Thresholds configured to a value other than the factory value display with a trailing asterisk (*) character.</p>
{-c --critlow}	<p>Displays the <i>critical low</i> threshold value for the sensor.</p>
{-W --warnhigh}	<p>Displays the <i>warning high</i> threshold value for the sensor.</p>
{-w --warnlow}	<p>Displays the <i>warning low</i> threshold value for the sensor.</p>
{-N --name}	<p>Displays the name of the sensor.</p>
{-d --description}	<p>Displays a description of the sensor.</p>
{-S --sensor-type}	<p>Displays the type of sensor (for use with --type).</p>
{-p --parent-comp}	<p>Displays the parent component list for the sensor. These are the components that are affected by changes in the value of a sensor (for example, the components that change severity as the sensor changes severity).</p>
{-s --severity}	<p>Displays the current severity lever of the sensor (nominal, warning, or critical).</p>
{--verbose}	<p>Displays all columns; you cannot use this argument with any of the other column addition options.</p>

TABLE G-2 Arguments for Subcommand `sensor get` (Continued)

Argument	Description
{-I --noid}	Suppresses the display of the sensor ID column. By default, this column always displays when more than one sensor is selected.
[-H --noheader]	Suppresses the column headings.
{ -D --Delim }	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

[TABLE G-3](#) lists the return codes for this subcommand.

Note – There are some sensors whose values do not change, some that are there to provide information in the event of a problem, and others to facilitate the proper operation of the software.

Many of these sensors do not have a related component (parent) associated with them. For example, the die-temperature sensor for a CPU has the CPU as its parent component and a fan-speed sensor has the fan as its parent component; the product-id sensor, however, only reports a static value and has no parent relationship.

This relationship establishes the component(s) that is affected by changes in the value of the sensor. You cannot modify the thresholds for sensors that do not have a parent relationship, since an event will never occur for these threshold crossings.

TABLE G-3 Return Codes for Subcommand `sensor get`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NotFound	5	Entity (user, service, file, path, or other) was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.

Sensor Set Subcommand

Description: Enables you to set some of the data associated with a specific sensor or a class of sensors.

Format

```
sensor set [{-i | --id} SENSOR_ID [{-i | --id} SENSOR_ID] ...]
[{{-C | --crithigh} VALUE} [{{-c | --critlow} VALUE}
[{-W | --warnhigh} VALUE} [{-w | --warnlow} VALUE} [{-v | --value}
{on|off}]] | {-r | --reset}}
sensor set [{-t | --type} TYPE_ID] [{{-C | --crithigh} VALUE}
[{{-c | --critlow} VALUE} [{-W | --warnhigh} VALUE} [{-w | --warnlow}
VALUE} [{-v | --value} {on|off}]] | {-r | --reset}}]
sensor set [{-R | --resetall}]
```

TABLE G-4 lists the arguments for this subcommand.

TABLE G-4 Arguments for Subcommand `sensor set`

Argument	Description
<code>{-i --id}</code>	Specifies the specific sensor on which to operate. You can specify multiple sensors by repeating <code>--id</code> .
<code>{-t --type}</code>	Specifies the specific sensor class on which to operate (for example, fan, voltage, and so on).
<code>{-C --crithigh}</code>	Specifies the <i>critical high</i> threshold value for the sensor. <ul style="list-style-type: none">Setting the string to <code>clear</code> disables the threshold.Setting the string to <code>reset</code> sets the value to the original factory-specified value.If the value specified ends in a percent sign (%), the threshold is set to that percentage of the nominal value for the sensor.Any other value is interpreted as the actual value to which to set the threshold.
<code>{-c --critlow}</code>	Specifies the <i>critical low</i> threshold value for the sensor. Setting the string to <code>clear</code> disables the threshold.
<code>{-W --warnhigh}</code>	Specifies the <i>warning high</i> threshold value for the sensor. Setting the string to <code>clear</code> disables the threshold.
<code>{-w --warnlow}</code>	Specifies the <i>warning low</i> threshold value for the sensor. Setting the string to <code>clear</code> disables the threshold.

TABLE G-4 Arguments for Subcommand `sensor set` (*Continued*)

Argument	Description
{-v --value}	Sets the value of the sensor.
{-r --reset}	Resets all thresholds for the specified sensor(s) to the factory defaults.
{-R --resetall}	Resets all thresholds for all sensors to the factory defaults.

Return Codes

[TABLE G-5](#) lists the return codes for this subcommand.

Note – There are some sensors whose values do not change, some that are there to provide information in the event of a problem, and others to facilitate the proper operation of the software.

Many of these sensors do not have a related component (parent) associated with them. For example, the die-temperature sensor for a CPU has the CPU as its parent component and a fan-speed sensor has the fan as its parent component; the product-id sensor, however, only reports a static value and has no parent relationship.

This relationship establishes the component(s) that is affected by changes in the value of the sensor. You cannot modify the thresholds for sensors that do not have a parent relationship, since an event will never occur for these threshold crossings.

TABLE G-5 Return Codes for Subcommand `sensor set`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NotFound	5	Entity (user, service, file, path, or other) was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.

Service Processor Commands

The `sp` command gets or sets the configuration values for the service processor (SP); generates or manages events and notices; or adds or modifies subscribers, event routes, and email-notification groups for the SP event manager.

[TABLE H-1](#) lists the groups of `sp` subcommands.

Note – Every subcommand returns a return code upon completion.

TABLE H-1 Service Processor Subcommand Groups

Subcommand	Description
Date	Sets or retrieves the date and time on the SP real-time clock (RTC).
DNS	Displays or configures the DNS client configuration on the SP.
Events	Returns detailed information or clears an event.
Hostname	Displays or resets the host name or domain name of the SP.
IP	Sets, modifies, or retrieves the SP network configuration.
JNET Address	Sets or retrieves the JNET address.
Locate Light	Sets the state or reads the value of the Locatelight switch.
Logfile	Retrieves or configures the event log file.
MAC address	Retrieves the MAC address for the SP.
Miscellaneous	Reads status for a component, retrieves the last port 80 postcode, restores settings to defaults, stores data in tar zipped format, or captures debug data.
Mount	Displays, creates, resets, or deletes a mount point.
SMTP	Manages information about SMTP email delivery.

TABLE H-1 Service Processor Subcommand Groups (*Continued*)

Subcommand	Description
SNMP	Manages SNMP functions.
SSL	Manages SSL capabilities.
Update Flash	Sets the update flag to start the full flash update or copies the Value-Add file to the Value-Add component of the SP flash.

SP Date Subcommands

The subcommands in [TABLE H-2](#) manage the date and time on the SP.

TABLE H-2 SP Date Subcommands

Subcommand	Description
<code>sp get date</code>	Retrieves the date and time from the SP real-time clock (RTC).
<code>sp set date</code>	Sets the date and time on the SP RTC.

SP Get Date Subcommand

Description: Retrieves the date and time from the SP real-time clock (RTC).

Format

```
sp get date
```

Return Codes

[TABLE H-3](#) lists the return codes for this subcommand.

TABLE H-3 Return Codes for Subcommand `sp get date`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoMemory	8	Insufficient memory.

SP Set Date Subcommand

Description: Sets the date and time on the SP real-time clock (RTC).

Format

```
sp set date DATE STRING
```

[TABLE H-4](#) lists the argument for this subcommand.

TABLE H-4 Argument for Subcommand `sp set date`

Argument	Description
<code>DATE STRING</code>	Specifies the date and time on the service processor RTC. The date string is a Coordinated Universal Time (UTC) date of the form YYYY-MM-DD HH:MM:SS.

You can use this command to initially set the platform RTC after the platform has lost CMOS backup power. If the platform is in the state in which the operating system (OS) is communicating with the SP, the platform time will override the SP time, which allows the platform and SP event times to be synchronized in the event log.

Return Codes

[TABLE H-5](#) lists the return codes for this command.

TABLE H-5 Return Codes for Subcommand `sp set date`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoMemory	8	Insufficient memory.
NWSE_FileError	18	File open, file missing, or a read or write error occurred.

SP DNS Subcommands

The subcommands in [TABLE H-6](#) manage the DNS configuration on the SP.

TABLE H-6 SP DNS Subcommands

Subcommand	Description
<code>sp disable dns</code>	Disables the DNS configuration on the SP.
<code>sp enable dns</code>	Configures the DNS configuration on the SP.
<code>sp get dns</code>	Displays the current DNS configuration on the SP.

SP Disable DNS Subcommand

Description: Disables the DNS configuration on the SP.

```
sp disable dns
```

When the SP is configured to use Dynamic Host Configuration Protocol (DHCP), DHCP automatically configures DNS settings. Changes to the DNS settings in this configuration can be replaced with the DHCP client.

Return Codes

TABLE H-7 lists the return codes for this command:

TABLE H-7 Return Codes for Subcommand `sp disable dns`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_NoPermission	6	Not authorized to perform this operation.

SP Enable DNS Subcommand

Description: Configures the DNS configuration on the SP.

Because applications do not see updated DNS resolver configurations (in `/etc/resolv.conf`) until they are restarted, this command restarts server processes that depend on DNS. This currently includes the `sshd` daemon and the Security Manager.

Format

```
sp enable dns { -n | --nameserver } NAMESERVER IP..  
{ -s | --searchdomain } SEARCH DOMAIN..
```

TABLE H-8 lists the arguments for this subcommand.

TABLE H-8 Arguments for Subcommand `sp enable dns`

Argument	Description
{ -n --nameserver }	Displays the name server IP addresses. If there is more than one, the addresses print on separate lines.
{ -s --searchdomain }	Displays the search domain(s). If there is more than one, the search domains print on separate lines.

Return Codes

TABLE H-9 lists the return codes for this subcommand.

TABLE H-9 Return Codes for Subcommand `sp enable dns`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_NoPermission	6	Not authorized to perform this operation.

SP Get DNS Subcommand

Description: Displays the current DNS configuration on the SP.

Format

```
sp get dns [{-n | --nameserver } | -s | --searchdomain } |  
{-H | --noheader }] [{-D | --Delim <DELIMITER>}]
```

TABLE H-10 lists the arguments for this subcommand.

TABLE H-10 Arguments for Subcommand `sp get dns`

Argument	Description
{ -n --nameserver }	Displays the name server(s). If there is more than one name server, they print on separate lines.
{ -s --searchdomain }	Displays the search domain(s). If there is more than one search domain, they print on separate lines.
{ -H --noheader }	Suppresses column headings.
[-D --Delim <DELIMITER>]	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

[TABLE H-11](#) lists the return codes for this subcommand.

TABLE H-11 Return Codes for Subcommand `sp get dns`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.

SP Events Subcommands

The subcommands in [TABLE H-12](#) manage events on the SP.

TABLE H-12 SP Events Subcommands

Subcommand	Description
<code>sp delete event</code>	Clears an existing event using the event ID.
<code>sp get events</code>	Returns detailed information about all active SP events.

SP Delete Event Subcommand

Description: Clears an existing event using the event ID.

Format

```
sp delete event { EVENT ID | {-a | --all}} [-q | --quiet]
```

TABLE H-13 lists the arguments for this subcommand.

TABLE H-13 Arguments for Subcommand `sp delete event`

Argument	Description
EVENT ID	Specifies the existing event to clear. This argument is repeatable to clear multiple events at one time.
[-a --all]	Removes all events.
[-q --quiet]	If the event to delete is not found, this argument specifies that no error be returned.

Return Codes

TABLE H-14 lists the return codes for this subcommand.

TABLE H-14 Return Codes for Subcommand `sp delete event`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NotFound	5	Entity (user, service, file, path, or other) was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoMemory	8	Insufficient memory.
NWSE_InvalidOpForState	22	Invalid operation for current state.

SP Get Events Subcommand

Description: Returns detailed information about all active SP events. By default, event ID, last update, component, severity, and a message are displayed.

Administrators can view detailed information about all the currently active system events and perform various actions related to each event.

Administrators can view this information using this command. For a list of all possible events, refer to [TABLE 3-4](#).

Format

```
sp get events [ {-i | --id} <EVENT ID> ] [{-d | --detail} ]
[{-v | --verbose}] [{-H | noheader}][{-D | --Delim <DELIMITER>}]
```

[TABLE H-15](#) lists the arguments for this subcommand.

TABLE H-15 Arguments for Subcommand `sp get events`

Argument	Description
{-i --id}	Specifies to display only information about this event; otherwise, information for all existing events is returned.
{-d --detail}	Specifies to display the history of either one or all events.
{ -v --verbose}	Specifies to display all columns.
{ -H --noheader }	Suppresses column headings.
{ -D --Delim }	Specifies to delimit columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

[TABLE H-16](#) lists the return codes for this subcommand.

TABLE H-16 Return Codes for Subcommand `sp get events`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NotFound	5	Entity (user, service, file, path, or other) not found.
NWSE_NoMemory	8	Insufficient memory.

SP Hostname Subcommands

The subcommands in [TABLE H-17](#) manage the SP host and domain.

TABLE H-17 SP Hostname Subcommands

Subcommand	Description
<code>sp get hostname</code>	Displays the current host name and optionally the domain name of the SP.
<code>sp set hostname</code>	Resets the host name or domain name of the SP to the specified name.

SP Get Hostname Subcommand

Description: Displays the current host name and optionally the domain name of the SP. This name is used by many of the networking programs to identify the machine. It is also used to identify a logging subdirectory for event logs.

Format

```
sp get hostname [-f | --fqdn]
```

[TABLE H-18](#) lists the argument for this subcommand.

TABLE H-18 Argument for Subcommand `sp get hostname`

Argument	Description
<code>[-f --fqdn]</code>	Causes the fully qualified host name to display.

Return Codes

[TABLE H-19](#) lists the return codes for this subcommand.

TABLE H-19 Return Codes for Subcommand `sp get hostname`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_NoMemory	8	Insufficient memory.
NWSE_Busy	9	Device or resource is busy.
NWSE_RPCConnected	11	RPC client already connected.
NWSE_RPCConnRefused	12	RPC connection refused.
NWSE_NoRouteToHost	13	No route to host (network down).
NWSE_HostDown	14	Host is down.

SP Set Hostname Subcommand

Description: Resets the host name or domain name of the SP to the specified name. This name is used by many of the networking programs to identify the machine.

Format

```
sp set hostname HOSTNAME
```

[TABLE H-20](#) lists the argument for this subcommand.

TABLE H-20 Argument for Subcommand `sp set hostname`

Argument	Description
<i>HOSTNAME</i>	Specifies the name of the host to set.

Return Codes

[TABLE H-21](#) lists the return codes for this subcommand.

TABLE H-21 Return Codes for Subcommand `sp set hostname`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoMemory	8	Insufficient memory.
NWSE_Busy	9	Device or resource is busy.
NWSE_RPCConnected	11	RPC client already connected.
NWSE_RPCConnRefused	12	RPC connection refused.
NWSE_NoRouteToHost	13	No route to host (network down).
NWSE_HostDown	14	Host is down.

SP IP Subcommands

The subcommands in [TABLE H-22](#) manage the SP network configuration.

TABLE H-22 SP IP Subcommands

Subcommand	Description
<code>sp get ip</code>	Retrieves the Ethernet-based network configuration information for the SP.
<code>sp set ip</code>	Sets or modifies the SP network configuration.

SP Get IP Subcommand

Description: Retrieves the Ethernet-based network configuration information for the SP, including IP address, network mask, and gateway. In addition, it indicates whether the SP is configured to use DHCP or a static IP address.

Format

```
sp get ip [-H | noheader] [{-D | --Delim <DELIMITER>}]
```

[TABLE H-23](#) lists the arguments for this subcommand.

TABLE H-23 Arguments for Subcommand `sp get ip`

Argument	Description
{ -H --noheader }	Suppresses column headings.
{ -D --Delim }	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

[TABLE H-24](#) lists the return codes for this subcommand.

TABLE H-24 Return Codes for Subcommand `sp get ip`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_NoMemory	8	Insufficient memory.
NWSE_Busy	9	Device or resource is busy.
NWSE_RPCConnected	11	RPC client already connected.
NWSE_RPCConnRefused	12	RPC connection refused.
NWSE_NoRouteToHost	13	No route to host (network down).
NWSE_HostDown	14	Host is down.

SP Set IP Subcommand

Description: Sets or modifies the SP network configuration.

Format

```
sp set ip dhcp [--nowait]
```

```
sp set ip static {-i | --ipaddress} IP_ADDRESS  
[{-n | --netmask} NETMASK] [{-g | --gateway} GATEWAY]] [-w | --nowait]
```

[TABLE H-25](#) lists the arguments for this subcommand.

TABLE H-25 Arguments for Subcommand `sp set ip`

Argument	Description
<code>{-i --ipaddress}</code>	Specifies the IP address you wish to set.
<code>{-n --netmask}</code>	Specifies the netmask; the default value is 255.255.255.0.
<code>{-g --gateway}</code>	Specifies the gateway; the default value is the existing gateway.
<code>{-w --nowait}</code>	If you specify the <code>-nowait</code> option, loss of connectivity will occur some time after the command returns. If you do not specify the <code>-nowait</code> option, your connections to the SP will be lost before the command returns.

Return Codes

[TABLE H-26](#) lists the return codes for this subcommand.

TABLE H-26 Return Codes for Subcommand `sp set ip`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.
<code>NWSE_InvalidArgument</code>	4	One or more arguments were incorrect or invalid.
<code>NWSE_NoPermission</code>	6	Not authorized to perform this operation.
<code>NWSE_NoMemory</code>	8	Insufficient memory.
<code>NWSE_Busy</code>	9	Device or resource is busy.
<code>NWSE_RPCConnected</code>	11	RPC client already connected.
<code>NWSE_RPCConnRefused</code>	12	RPC connection refused.

TABLE H-26 Return Codes for Subcommand `sp set ip` (Continued)

Return Code	ID	Description
NWSE_NoRouteToHost	13	No route to host (network down).
NWSE_HostDown	14	Host is down.
NWSE_UnknownError	15	Miscellaneous error not captured by other errors.
NWSE_GatewayOffNet	16	Gateway address is not on network.
NWSE_NetMaskIncorrect	17	An inappropriate netmask was specified.

SP JNET Address Subcommands

The JNET address is used for communications between the SP and the platform. The subcommands in [TABLE H-27](#) manage the SP JNET address.

TABLE H-27 SP JNET Subcommands

Subcommand	Description
<code>sp get jnet</code>	Retrieves the JNET address.
<code>sp set jnet</code>	Sets the JNET address.

SP Get JNET Subcommand

Description: Retrieves the IP address of the platform JNET driver.

Format

```
sp get jnet [{-H | --noheader}] [{-D | --Delim <DELIMITER>}]
```

[TABLE H-28](#) lists the arguments for this subcommand.

TABLE H-28 Arguments for Subcommand `sp get jnet`

Argument	Description
{ -H --noheader }	Suppresses column headings.
{ -D --Delim }	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

[TABLE H-29](#) lists the return codes for this subcommand.

TABLE H-29 Return Codes for Subcommand `sp get jnet`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_NoMemory	8	Insufficient memory.
NWSE_Busy	9	Device or resource is busy.
NWSE_HostDown	14	Host is down.

SP Set JNET Subcommand

Description: Sets or modifies the SP and platform network addresses for JNET. Because of the firewall between these drivers, you must specify both addresses at the same time.

Both the SP and platform JNET addresses must be on the same Class C subnet.

Format

```
sp set jnet {-p | --platform} IP ADDRESS {-s | --sp} IP ADDRESS
```

[TABLE H-30](#) lists the arguments for this subcommand.

TABLE H-30 Arguments for Subcommand `sp set jnet`

Argument	Description
<code>{-p --platform}</code>	Specifies the IP address for the platform.
<code>{-s --sp}</code>	Specifies the IP address for the SP.

Note – If you change the default addresses of JNET using this command and then reinstall the platform OS or reset the SP through the subcommand `sp reset to default-settings`, you must reissue the subcommand `sp set jnet` to re-establish the JNET connection.

Otherwise, the connection will be out-of-sync (one address will be modified and one will be reset to the default address).

Return Codes

[TABLE H-31](#) lists the return codes for this subcommand.

TABLE H-31 Return Codes for Subcommand `sp set jnet`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoMemory	8	Insufficient memory.
NWSE_Busy	9	Device or resource is busy.
NWSE_HostDown	14	Host is down.

SP Locatelight Subcommands

The subcommands in [TABLE H-32](#) manage the Locatelight switch.

TABLE H-32 SP Locatelight Subcommands

Subcommand	Description
<code>sp get locatelight</code>	Reads the value of the Locatelight switch (which represents the state of the front-panel and rear-panel identification lights).
<code>sp set locatelight</code>	Sets the state of the Locatelight switch.

SP Get Locatelight Subcommand

Description: Reads the value of the Locatelight switch (which represents the state of the front-panel and rear-panel identification lights). The possible states are blinking or off.

Format

```
sp get locatelight
```

Return Codes

TABLE H-33 lists the return codes for this subcommand.

TABLE H-33 Return Codes for Subcommand `sp get locatelight`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoPermission	6	Not authorized to perform this operation.

SP Set Locatelight Subcommand

Description: Sets the state of the Locatelight switch (which describes the state of the front-panel and rear-panel identification lights).

Format

```
sp set locatelight {BLINK | OFF}
```

Return Codes

[TABLE H-34](#) lists the return codes for this subcommand.

TABLE H-34 Return Codes for Subcommand `sp set locatelight`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoPermission	6	Not authorized to perform this operation.

SP Logfile Subcommands

The subcommands in [TABLE H-35](#) manage the SP log files.

TABLE H-35 SP Logfile Subcommands

Subcommand	Description
<code>sp get logfile</code>	Retrieves the eventlog file configuration.
<code>sp set logfile</code>	Configures the eventlog file that is the destination of all Event Manager events and notices.

SP Get Logfile Subcommand

Description: Retrieves the event log file configuration.

Format

```
sp get logfile [-H | --noheader] [{-D | --Delim <DELIMITER>}]
```

TABLE H-36 lists the arguments for this subcommand.

TABLE H-36 Arguments for Subcommand `sp get logfile`

Argument	Description
{ <code>-H</code> <code>--noheader</code> }	Suppresses column headings.
{ <code>-D</code> <code>--Delim</code> }	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

TABLE H-37 lists the return codes for this subcommand.

TABLE H-37 Return Codes for Subcommand `sp get logfile`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.
<code>NWSE_RPCTimeout</code>	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
<code>NWSE_RPCNotConnected</code>	3	Unable to connect to the RPC server.
<code>NWSE_NoMemory</code>	8	Insufficient memory.

SP Set Logfile Subcommand

Description: Configures the event log file that is the destination of all Event Manager events and notices.

Format

```
sp set logfile [ {-f | --file} FILENAME] [ {-s | --size} SIZE]
```

You must specify the name of the file to which the Event Manager sends logs. When setting the log file using this command, specify only the name of the log file without the path. File names cannot contain the forward slash character (/), backward relative-path reference (..) or the less-than symbol (<).

[TABLE H-38](#) lists the arguments for this subcommand.

TABLE H-38 Arguments for Subcommand `sp set logfile`

Argument	Description
<code>{-f --file}</code>	Specifies the name of the file within the directory to which the Event Manager sends logs.
<code>{-s --size}</code>	Specifies the size of the file in megabytes. A minimum size of 0.01 MB is required for this log file.

Return Codes

[TABLE H-39](#) lists the arguments for this subcommand.

TABLE H-39 Return Codes for Subcommand `sp set logfile`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.
<code>NWSE_RPCTimeout</code>	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
<code>NWSE_RPCNotConnected</code>	3	Unable to connect to the RPC server.
<code>NWSE_NoPermission</code>	6	Not authorized to perform this operation.
<code>NWSE_NoMemory</code>	8	Insufficient memory.

SP Miscellaneous Subcommands

The subcommands in [TABLE H-40](#) manage miscellaneous SP functions.

TABLE H-40 Miscellaneous SP Subcommands

Subcommand	Description
<code>sp create test events</code>	Tests and validates different types of configurations you might be considering for the SP.
<code>sp get mac address</code>	Retrieves the MAC address for the SP.
<code>sp get port 80</code>	Retrieves the last port 80 POST code from the PRS Port80 register.
<code>sp get status</code>	Returns the status of the overall system.
<code>sp get tdulog</code>	Captures data and stores it on the SP in compressed format.
<code>sp autoconfigure</code>	Configures an SP with the same configuration as that of another service processor.
<code>sp reboot</code>	Restarts the SP.
<code>sp reset</code>	Restores selected settings of the SP to the default factory configuration.

SP Create Test Events Subcommand

Description: This command helps you to test and validate different types of configurations that you might be considering for the SP (for example, configurations involving event forwarding, such as SNMP, SMTP, log files, or directory services).

When you execute this command, the SP generates three new events, each with a different level of severity. To view these events, run the subcommand `sp get events`. Then retrieve the configuration settings for SNMP, SMTP, and log files and validate that the events reached the configured destinations.

SNMP

To configure destinations for SNMP traps, run the subcommand `sp add snmp-destination`. To view your current settings, run the subcommand `sp get snmp-destinations`. All SP events are translated into SNMP traps and are sent to all configured destinations.

SMTP

You can configure destinations for events as SMTP addresses. You must first run the subcommand `sp set smtp server` to configure your SMTP *server* and *from* address. Then run the subcommand `sp update smtp subscriber` to configure the destination addresses and formats to be used for different severity events. This command enables you to configure the format of the events that are delivered (long or short) and the recipient(s) of events of different severities.

All SP events are translated into alerts and are sent to all configured SMTP destinations according to the event severity (the subscribers). You can run the subcommands `sp get smtp server` and `sp get smtp subscribers` to view your current settings.

Log Files

All events that are generated on the SP are written to either a default or a user-specified log file if a mount point is configured. To add a mount point, run the subcommand `sp add mount`. You can then run the subcommand `sp set logfile` to configure the name and size of the target file on the mounted file system. To view your current settings, run the subcommands `sp get mounts` and `sp get logfile`.

Format

```
sp create test events
```

Return Codes

[TABLE H-41](#) lists the return codes for this command.

TABLE H-41 Return Codes for Subcommand `sp create test events`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoMemory	8	Insufficient memory.

SP Get MAC Address Subcommand

Description: Retrieves the MAC address for the SP.

Format

```
sp get mac
```

Return Codes

[TABLE H-42](#) lists the return codes for this subcommand.

TABLE H-42 Return Codes for Subcommand `sp get mac`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_ServiceNotAvailable	24	Requested service is not available.

SP Get Port 80 Subcommand

Description: Retrieves the last port 80 POST code from the PRS Port80 register. The register is written by platform BIOS during platform boot. You can use this subcommand to debug platform-boot problems.

Format

```
sp get port80 {-m | --monitor}
```

[TABLE H-43](#) lists the arguments for this subcommand.

TABLE H-43 Argument for Subcommand `sp get port80`

Argument	Description
<code>{-m --monitor}</code>	Allows for continuous monitoring of the port 80 traffic.

You can also retrieve the last ten port 80 POST codes using the operator panel.

For more details about using the operator-panel menus, refer to the *Sun Fire V20z and Sun Fire V40z Servers—User Guide*.

See [TABLE H-45](#) for a list of the Power On Self Test (POST) codes for the Phoenix BIOS.

See [TABLE H-46](#) for a list of the boot block codes on Flash ROM.

Return Codes

[TABLE H-44](#) lists the return codes for this subcommand.

TABLE H-44 Return Codes for Subcommand `sp get port80`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoMemory	8	Insufficient memory.
NWSE_ServiceNotAvailable	24	Requested service is not available.

BIOS POST Codes

[TABLE H-45](#) lists the POST codes for the Phoenix BIOS.

TABLE H-45 BIOS POST Codes

POST Code	Description
02	Verify real mode
03	Disable non-maskable interrupt (NMI)
04	Get CPU type
06	Initialize system hardware
07	Disable shadow and execute code from the ROM
08	Initialize chipset with initial POST values
09	Set IN POST flag
0A	Initialize CPU registers
0B	Enable CPU cache
0C	Initialize caches to initial POST values
0E	Initialize I/O component
0F	Initialize the local bus IDE

TABLE H-45 BIOS POST Codes (*Continued*)

POST Code	Description
10	Initialize power management
11	Load alternate registers with initial POST values
12	Restore CPU control word during warm boot
13	Initialize PCI bus mastering devices
14	Initialize keyboard controller
16	BIOS ROM checksum
17	Initialize cache before memory autosize
18	8254 programmable interrupt timer initialization
1A	8237 DMA controller initialization
1C	Reset programmable interrupt controller
20	Test DRAM refresh
22	Test 8742 keyboard controller
24	Set ES segment register to 4GB
26	Enable gate A20 line
28	Autosize DRAM
29	Initialize POST memory manager
2A	Clear 512-KB base RAM
2C	RAM failure on address line xxxx
2E	RAM failure on data bits xxxx of low byte of memory bus
2F	Enable cache before system BIOS shadow
30	RAM failure on data bits xxxx of high byte of memory bus
32	Test CPU bus clock frequency
33	Initialize Phoenix Dispatch Manager
36	Warm start shut down
38	Shadow system BIOS ROM
3A	Autosize cache
3C	Advanced configuration of chipset registers
3D	Load alternate registers with CMOS values
41	Initialize extended memory for RomPilot
42	Initialize interrupt vectors

TABLE H-45 BIOS POST Codes (*Continued*)

POST Code	Description
45	POST device initialization
46	Check ROM copyright notice
47	Initialize I20 support
48	Check video configuration against CMOS
49	Initialize PCI bus and devices
4A	Initialize all video adapters in system
4B	QuietBoot start (optional)
4C	Shadow video BIOS ROM
4E	Display BIOS copyright notice
4F	Initialize MultiBoot
50	Display CPU type and speed
51	Initialize EISA board
52	Test keyboard
54	Set key click if enabled
55	Enable USB devices
58	Test for unexpected interrupts
59	Initialize POST display service
5A	Display prompt "Press F2 to enter SETUP"
5B	Disable CPU cache
5C	Test RAM between 512KB and 640KB
60	Test extended memory
62	Test extended memory address lines
64	Jump to UserPatch1
66	Configure advanced cache registers
67	Initialize multi processor APIC
68	Enable external and CPU caches
69	Setup system management mode (SMM) area
6A	Display external L2 cache size
6B	Load custom defaults (optional)
6C	Display shadow area message

TABLE H-45 BIOS POST Codes (*Continued*)

POST Code	Description
6E	Display possible high address for Upper Memory Block (UMB) recovery
70	Display error messages
72	Check for configuration errors
76	Check for keyboard errors
7C	Set up hardware interrupt vectors
7D	Initialize Intelligent System Monitoring
7E	Initialize coprocessor if present
80	Disable on-board super I/O ports and IRQs
81	Late POST device initialization
82	Detect and install external RS-232 ports
83	Configure non-MCD IDE controllers
84	Detect and install external parallel ports
85	Initialize PC compatible PnP ISA devices
86	Reinitialize on-board I/O ports
87	Configure motherboard configurable devices (optional)
88	Initialize BIOS data area
89	Enable non-maskable interrupts (NMIs)
8A	Initialize extended BIOS data area
8B	Test and initialize PS/2 mouse
8C	Initialize floppy controller
8F	Determine number of ATA drives (optional)
90	Initialize hard disk controllers
91	Initialize local bus hard disk controllers
92	Jump to UserPatch2
93	Build MPTABLE for multiprocessor boards
95	Install CD-ROM for boot
96	Clear huge ES segment register
97	Fix up multiprocessor table
98	Search for option ROMs
99	Check for SMART drive (optional)

TABLE H-45 BIOS POST Codes (*Continued*)

POST Code	Description
9A	Shadow option ROMs
9C	Set up power management
9D	Initialize security engine (optional)
9E	Enable hardware interrupts
9F	Determine number of ATA and SCSI drives
A0	Set time of day
A2	Check key lock
A4	Initialize typematic rate
A8	Erase F2 prompt
AA	Scan for F2 keystroke
AC	Enter setup
AE	Clear boot flag
B0	Check for errors
B1	Inform RomPilot about the end of POST
B2	POST done - prepare to boot operating system
B4	One short beep
B5	Terminate QuietBoot (optional)
B6	Check password
B7	Initialize ACPI BIOS
B9	Prepare boot
BA	Initialize DMI parameters
BB	Initialize PnP option ROMs
BC	Clear parity checkers
BD	Display multiboot menu
BE	Clear screen
BF	Check virus and backup reminders
C0	Try to boot with interrupt 19
C1	Initialize POST Error Manager (PEM)
C2	Initialize error logging
C3	Initialize error display function

TABLE H-45 BIOS POST Codes (*Continued*)

POST Code	Description
C4	Initialize system error handler
C5	PnP dual CMOS (optional)
C6	Initialize notebook docking (optional)
C7	Initialize notebook docking rate
C8	Force check (optional)
C9	Extended checksum (optional)
CA	Redirect Int 15h to enable remote keyboard
CB	Redirect Int 13 to Memory Technologies Devices such as ROM, RAM, PCMCIA, and serial disk
CC	Redirect Int 10h to enable remote serial video
CD	Re-map I/O and memory for PCMCIA
CE	Initialize digitizer and display message
D2	Unknown interrupt

Boot Block Codes for Flash ROM

[TABLE H-46](#) lists the boot block codes in Flash ROM.

TABLE H-46 Boot Block Codes in Flash ROM

Post Code	Description
E0	Initialize the chipset
E1	Initialize the bridge
E2	Initialize the CPU
E3	Initialize the system timer
E4	Initialize system I/O
E5	Check force recovery boot
E6	Checksum BIOS ROM
E7	Go to BIOS
E8	Set Huge Segment
E9	Initialize Multiprocessor
EA	Initialize OEM special code

TABLE H-46 Boot Block Codes in Flash ROM (*Continued*)

Post Code	Description
EB	Initialize PIC and DMA
EC	Initialize memory type
ED	Initialize memory size
EE	Shadow Boot Block
EF	System memory test
F0	Initialize interrupt vectors
F1	Initialize run time clock
F2	Initialize video
F3	Initialize System Management Manager
F4	Output one beep
F5	Clear Huge Segment
F6	Boot to mini DOS
F7	Boot to full DOS

SP Autoconfigure Subcommand

Description: Copies the configuration settings from one service processor to another service processor.

You can also perform autoconfiguration from the operator panel to perform this same function. For more information, see [“Autoconfiguring the SP” on page 36](#).

The command invokes a series of HTTPS requests for all of the configuration files from the source machine and then loads configuration data to the second SP. The uploaded files are copied to the local `pstore` file and the second SP is rebooted. While this operation is running, you cannot execute any other configuration changes.

By default, configuration sharing is disabled and you must enable it on the source machine.

- To identify the status of configuration sharing on a server, refer to [“Access Get Config-Sharing Subcommand” on page 116](#).
- To enable configuration sharing, refer to [“Access Enable Config-Sharing Subcommand” on page 114](#).
- To disable configuration sharing, refer to [“Access Disable Config-Sharing Subcommand” on page 115](#).

After the command is executed, a message is displayed indicating that the SP will be rebooted. The SSH connection then terminates.

Format

```
sp autoconfigure {{ -s | --sp } SP_IP_OR_HOST [-H | --noheader]
```

[TABLE H-47](#) lists the arguments for this subcommand.

TABLE H-47 Arguments for Subcommand `sp autoconfigure`

Argument	Description
{ -s --sp }	The DNS host name or IP address of the machine from which to copy the configuration information.
[-H --noheader]	Suppresses header output.

Return Codes

[TABLE H-48](#) lists the return codes for this subcommand.

TABLE H-48 Return Codes for Subcommand `sp autoconfigure`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoMemory	8	Insufficient memory.
NWSE_HostDown	14	Host is down.
NWSE_TimedOut	23	Operation timed out.
NWSE_ServiceNotAvailable	24	Requested service is not available.

SP Get Status Subcommand

Description: Returns the status of the overall system.

Format

```
sp get status
```

[TABLE H-49](#) lists the arguments for this subcommand.

TABLE H-49 Arguments for Subcommand `sp get status`

Argument	Description
Nominal	All components are operating within normal parameters.
Warning	One or more components are operating at warning levels.
Critical	One or more components are operating out of specification or have failed.

Return Codes

[TABLE H-50](#) lists the return codes for this subcommand.

TABLE H-50 Return Codes for Subcommand `sp get status`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoPermission	6	Not authorized to perform this operation.

SP Get TDULog Subcommand

Description: The Troubleshooting Dump Utility (TDU) captures debug data. When you execute this command, this data is gathered and stored on the SP in a compressed tar file.

Format

```
sp get tdulog [{-f | --filename} FILENAME or STDOUT ]
[{-c | --cpuregs} CPU REGISTERS]
[{-p | --pciregs} PCI REGISTERS]
[{-r | --reset} RESET PLATFORM]
```

[TABLE H-51](#) lists the arguments for this subcommand.

TABLE H-51 Arguments for Subcommand `sp get tdulog`

Argument	Description
<code>{-f --filename}</code>	<p>Optional. The name of the output file to which the log files are copied, or the fully qualified path name. File names cannot contain the backward relative path reference (..) or the less-than symbol (<). The following log files are created by default:</p> <ul style="list-style-type: none">• <code>envLog</code>: contains the environment variables• <code>vpdLog</code>: contains raw VPD data <p>Additional log files are created for CPU2 and CPU3 registers. The TDU adds the <code>.tar</code> to the filename. If you do not provide a filename, <code>tdulog</code> creates a file named <code>tdulog.tar</code> in the current directory, if the current directory is NFS mounted. An NFS-mounted file share must be used to store the output file. The TDU data can also be redirected to <code>stdout</code>. If the file name is <code>"stdout"</code>, the output is sent to <code>stdout</code> and the log files are not created.</p>
<code>{-c --cpuregs}</code>	Reads the K-8 registers (GPRs, MSRs, TCB and machine check) from up to four CPUs.
<code>{-p --pciregs}</code>	Reads all PCI registers on the system.
<code>{-r --reset}</code>	Resets the platform if unable to access HDT mode.

The register name, address, and data are logged to a file. For example, the information for CPU 0 is shown in [TABLE H-52](#).

TABLE H-52 Sample Information for Subcommand `sp get tdulog` on CPU 0

Reg Name	Reg Addr	Reg Data
<code>MSR_MCG_CAP_MSR</code>	0xc0020179	0x0000000000000105
<code>MSR_MCG_STAT_MSR</code>	0xc002017a	0x0000000000000000
<code>MSR_MCG_CTL_MSR</code>	0xc002017b	0x000000000000001F
<code>MSR_MC0_CTL</code>	0xc0020400	0x000000000000007F

Return Codes

TABLE H-53 lists the return codes for this subcommand.

TABLE H-53 Return Codes for Subcommand `sp get tdulog`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_NotFound	5	Entity (user, service, file, path or other) was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_MissingArgument	7	Missing argument(s).
NWSE_UnknownError	15	Miscellaneous error not captured by other errors.
NWSE_FileError	18	File open, file missing, or a read or write error occurred.
NWSE_NotMounted	21	File system is not mounted.
NWSE_ServiceNotAvailable	24	Requested service is not available.

SP Reboot Subcommand

Description: Restarts the SP. This command is useful in emergency situations in which you might not have physical access to a machine.

Format

```
sp reboot [ {-f | --forced} ]
```

TABLE H-54 lists the argument for this command.

TABLE H-54 Argument for Subcommand `sp reboot`

Argument	Description
<code>{-f --forced}</code>	Results in a hard power off.

Return Codes

TABLE H-55 lists the return codes for this command.

TABLE H-55 Return Codes for Subcommand `sp reboot`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoMemory	8	Insufficient memory.

SP Reset Subcommand

Description: Restores selected settings of the SP to the default factory configuration.

The SP configuration files are stored in the directory `/pstore`. When you boot the system, the SP copies these configuration files from `/pstore` to `/etc` whenever the files are missing from `/etc`. Resetting the SP to its default configuration is accomplished by deleting the configuration files in the directory `/pstore`. A reboot of the SP is necessary for the SP reset to take effect.

By default, the SP reboots 60 seconds after the subcommand `sp reset to default-settings` executes, unless you specify the `--nowait` option, in which case the reboot occurs immediately.

A message is displayed every 20 seconds to indicate that the reboot will occur.

Format

```
sp reset to default-settings {-a | --all} {-c | --config}
{-n | --network} {-s | --ssh} {-u | --users} {-W | --nowait}
```

TABLE H-56 lists the arguments for this command.

TABLE H-56 Arguments for Subcommand `sp reset`

Argument	Description
{-a --all}	Resets all SP settings to their default configuration. When the SP reboots, the settings are reset to their default values. This option also includes events and IPMI settings.
{-c --config}	Resets other system configuration settings to their default configuration. When the SP reboots, the system settings are reset to their default values.
{-n --network}	Resets network settings to their default configuration. When the SP reboots, it has no network capabilities or host name. Its NFS mounts fail and you cannot log on to the SP remotely through SSH. Set up the network configuration for the SP through the operator panel to restore network functions. Set the host name for the SP in order to refer to the SP by name and set up the file <code>resolv.conf</code> in order to refer to other systems by name instead of by dot-quad IP addresses. This option deletes all the network files in the directory <code>/pstore</code> .
{-s --ssh}	Resets SSH settings to their default configuration. When the SP reboots, new SSH public and private keys are generated. Using SSH to access the SP from a remote system that had previously logged in to the SP causes a failure with a message about the “Remote Host Identification” changing because the SSH key on the SP has changed. The remote system must delete its SSH public key entry for the SP in order to SSH into the SP successfully. This option deletes all the files in the directory <code>/pstore/ssh/</code> .
{-u --users}	Resets user authentication settings to their default configuration. When the SP reboots, all user accounts will have been deleted and you cannot log into the SP remotely through SSH.
[-W --nowait]	Reboots the SP immediately.

Note – If you change the default addresses of JNET using this command and then reinstall the platform OS or reset the SP by running the subcommand `sp reset to default-settings`, you must reissue the subcommand `sp set jnet` to re-establish the JNET connection.

Otherwise, the connection will be out-of-sync (one address will be modified and one will be reset to the default address).

Return Codes

[TABLE H-57](#) list the return codes for this command.

TABLE H-57 Return Codes for Subcommand `sp reset`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoMemory	8	Insufficient memory.
NWSE_Busy	9	Device or resource is busy.
NWSE_RPCConnected	11	RPC client already connected.
NWSE_RPCConnRefused	12	RPC connection refused.
NWSE_NoRouteToHost	13	No route to host (network down).
NWSE_HostDown	14	Host is down.

SP Mount Subcommands

The subcommands in [TABLE H-58](#) manage the SP mount points.

TABLE H-58 SP Mount Subcommands

Subcommand	Description
<code>sp add mount</code>	Creates or resets a mount point.
<code>sp delete mount</code>	Deletes the specified mount point.
<code>sp get mounts</code>	Displays the current mount points on the SP.

SP Add Mount Subcommand

Description: Creates or resets a mount point.

Format

```
sp add mount {-r | --remote} REMOTE_FILE_SYSTEM  
[{-l | --local} LOCAL_MOUNT_POINT] [{-u | --user} USERNAME]  
[{-p | --password} PASSWORD] [{-W | --nowait}]
```

TABLE H-59 lists the arguments for this subcommand.

TABLE H-59 Arguments for Subcommand `sp add mount`

Argument	Description
{-r --remote}	<p>Specifies the remote server and file system to use.</p> <p>If the remote file system is exported over NFS, use the following format to specify it:</p> <pre>-r <server_name_ID>:<path></pre> <p>If the remote file system is exported over CIFS (Windows network share), use the following format to specify it:</p> <pre>-r //<server_name>/<share_name></pre> <p>User name and password options are appropriate only when mounting CIFS file systems. In these examples, <i>Server_Name</i> is the IP address or host name of the remote server.</p> <p>The required format for remote NFS or SMB mounts is:</p> <ul style="list-style-type: none">• NFS: <i>server_name:/server_exported_mountpoint</i>• SMB: <i>//server_name/windows_share_name</i>
{-l --local}	<p>Optional. Specifies the local mount point. The only mount point supported is <i>/mnt</i>.</p>
{-u --user}	<p>Specifies the Microsoft Windows account user name. If Windows domains are in force, you may need to specify the domain, as in the following example:</p> <pre>-u <File_Server_Domain>\<username></pre>
{-p --password}	<p>Specifies the Microsoft Windows account password.</p>
{-W --nowait}	<p>If <code>--nowait</code> is specified, there is no wait for asynchronous commands to complete.</p>

Note – Several error messages might appear when executing an `smb mount` while mounting Microsoft Windows partitions. Check that the mount succeeded after the call by running the subcommand `sp get mounts`.

Return Codes

[TABLE H-60](#) lists the return codes for this subcommand.

TABLE H-60 Return Codes for Subcommand `sp add mount`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoMemory	8	Insufficient memory.
NWSE_Busy	9	Device or resource is busy.
NWSE_RPCConnected	11	RPC client already connected.
NWSE_RPCConnRefused	12	RPC connection refused.
NWSE_NoRouteToHost	13	No route to host (network down).
NWSE_HostDown	14	Host is down.
NWSE_UnknownError	15	Miscellaneous error not captured by other errors.

SP Delete Mount Subcommand

Description: Deletes a mount point.

Format

```
sp delete mount LOCAL MOUNT POINT [-q | --quiet]
```

[TABLE H-61](#) lists the arguments for this subcommand.

TABLE H-61 Arguments for Subcommand `sp delete mount`

Argument	Description
LOCAL MOUNT POINT	Specifies the mount point to remove. If you do not specify the local mount point, <code>/mnt</code> is implicit as the default value.
<code>[-q --quiet]</code>	If the mount point to delete is not found, this argument specifies that no error be returned.

Return Codes

TABLE H-62 lists the return codes for this subcommand.

TABLE H-62 Return Codes for Subcommand `sp delete mount`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoMemory	8	Insufficient memory.
NWSE_Busy	9	Device or resource is busy.
NWSE_RPCConnected	11	RPC client already connected.
NWSE_RPCConnRefused	12	RPC connection refused.
NWSE_NoRouteToHost	13	No route to host (network down).
NWSE_HostDown	14	Host is down.
NWSE_UnknownError	15	Miscellaneous error not captured by other errors.

SP Get Mount Subcommand

Description: Displays the current mount points on the SP.

Format

```
sp get mounts [{-l | --local} MOUNTPOINT] [-H | --noheader]
[{-D | --Delim <DELIMITER>}]
```

TABLE H-63 lists the arguments for this subcommand.

TABLE H-63 Arguments for Subcommand `sp get mount`

Argument	Description
{ -l --local }	Specifies the local mount point. If you do not specify -l, /mnt is implicit as the local mount point.
{ -H --noheader }	Suppresses column headings.
{ -D --Delim }	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

TABLE H-64 lists the return codes for this subcommand.

TABLE H-64 Return Codes for Subcommand `sp get mount`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_NotFound	5	Entity (user, service, file, path or other) not found.
NWSE_NoMemory	8	Insufficient memory.
NWSE_Busy	9	Device or resource is busy.
NWSE_RPCConnected	11	RPC client already connected.
NWSE_RPCConnRefused	12	RPC connection refused.
NWSE_NoRouteToHost	13	No route to host (network down).
NWSE_HostDown	14	Host is down.
NWSE_NotMounted	21	File system is not mounted.

SP SMTP Subcommands

The subcommands in [TABLE H-65](#) manage SMTP communication.

TABLE H-65 SP SMTP Subcommands

Subcommand	Description
<code>sp get smtp server</code>	Retrieves the SMTP server information.
<code>sp set smtp server</code>	Configures the SP SMTP client with the address for the remote SMTP server.
<code>sp get smtp subscribers</code>	Returns detailed information about one or all SMTP subscribers.
<code>sp update smtp subscriber</code>	Updates the information for an existing SMTP subscriber.

SP Get SMTP Server Subcommand

Description: Retrieves the SMTP server information, including the *from* address.

Format

```
sp get smtp server [-H | --noheader] [{-D | --Delim <DELIMITER>}]
```

[TABLE H-66](#) lists the arguments for this subcommand.

TABLE H-66 Arguments for Subcommand `sp get smtp server`

Argument	Description
{ -H --noheader }	Suppresses column headings.
{ -D --Delim }	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

TABLE H-67 lists the return codes for this subcommand.

TABLE H-67 Return Codes for Subcommand `sp get smtp server`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.

SP Set SMTP Server Subcommand

Description: Configures the SP SMTP client with the information for the remote SMTP server, including the address and optional port number.

Format

```
sp set smtp server [{-f | --from} FROM FIELD ] IP OR HOSTNAME OF SMTP SERVER
```

TABLE H-68 lists the arguments for this subcommand.

TABLE H-68 Arguments for Subcommand `sp set smtp server`

Argument	Description
<code>{-f --from}</code>	Specifies the <i>from</i> field for the SMTP server.
IP OR HOSTNAME OF SMTP SERVER	Specifies the IP address or the host name of the SMTP server.

The value you supply is prepended onto `@hostname | ip_address`. The default value is `system`.

For example, if you enter `admin` for `sp_22`, email messages are sent from `admin@sp_22`.

If the host name is not set, the IP address is used as shown in this example:
`admin@10.10.30.55`.

Return Codes

[TABLE H-69](#) lists the return codes for this subcommand.

TABLE H-69 Return Codes for Subcommand `sp set smtp server`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.

SP Get SMTP Subscribers Subcommand

Description: Returns detailed information about one or all SMTP subscribers.

Format

```
sp get smtp subscribers [{-n | --name} <NAME>] [-H | noheader]
[{-D | --Delim <DELIMITER>}]
```

[TABLE H-70](#) lists the arguments for this subcommand.

[TABLE H-71](#) lists the default SMTP subscribers.

TABLE H-70 Arguments for Subcommand `sp get smtp subscribers`

Argument	Description
{ -n --namsriver }	Specifies the name of the SMTP subscriber for which to retrieve information. If you do not specify this option, the command returns information for all subscribers.
{ -H --noheader }	Suppresses column headings.
{ -D --Delim }	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

TABLE H-71 Default SMTP Subscribers

Subscriber	Description
SMTP_Info_Short	Short email message, informational severity
SMTP_Info_Long	Long email message, informational severity
SMTP_Warning_Short	Short email message, warning severity
SMTP_Warning_Long	Long email message, warning severity
SMTP_Critical_Short	Short email message, critical severity
SMTP_Critical_Long	Long email message, critical severity

Long email messages contain full event details in the message body, while short email messages contain no message body. Both types have a descriptive subject line.

The short email format is intended to be used with pagers and other wireless access devices with which message-size constraints might prevent reception of the long format message.

Return Codes

[TABLE H-72](#) lists the return codes for this subcommand.

TABLE H-72 Return Codes for Subcommand `sp get smtp subscribers`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NotFound	5	Entity (user, service, file, path or other) was not found.

SP Update SMTP Subscriber Subcommand

Description: Updates the information for an existing SMTP subscriber.

Format

```
sp update smtp subscriber  
{-n | --name} NAME {-r | --recipients} ADDRESS LIST
```

[TABLE H-73](#) lists the arguments for this subcommand.

[TABLE H-74](#) lists the default SMTP subscribers.

Note – All options replace the existing values with the new value. Unspecified options leave existing settings as they are. For example, if you specify only the `-r` option for an existing subscriber, the existing email address list is replaced with the new list specified in the command.

TABLE H-73 Arguments for Subcommand `sp update smtp subscriber`

Argument	Description
<code>{-n --name}</code>	Specifies the name of the SMTP subscriber to update. This argument is repeatable to update multiple SMTP subscribers at one time.
<code>{-r --recipients}</code>	Specifies the address list of recipients for the SMTP subscriber.

TABLE H-74 Default SMTP Subscribers

Subscriber	Description
<code>SMTP_Info_Short</code>	Short email message, informational severity
<code>SMTP_Info_Long</code>	Long email message, informational severity
<code>SMTP_Warning_Short</code>	Short email message, warning severity
<code>SMTP_Warning_Long</code>	Long email message, warning severity
<code>SMTP_Critical_Short</code>	Short email message, critical severity
<code>SMTP_Critical_Long</code>	Long email message, critical severity

Long email messages contain full event details in the message body, while short email messages contain no message body. Both types have a descriptive subject line.

The short email format is intended to be used with pagers and other wireless access devices with which message-size constraints may prevent reception of the long format message.

Return Codes

[TABLE H-75](#) lists the return codes for this command.

TABLE H-75 Return Codes for Subcommand `sp update smtp subscriber`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NotFound	5	Entity (user, service, file, path, etc.) was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.

SP SNMP Subcommands

The subcommands in [TABLE H-76](#) manage SNMP communications.

TABLE H-76 SP SNMP Subcommands

Subcommand	Description
<code>sp add snmp destination</code>	Adds an SNMP destination.
<code>sp delete snmp destination</code>	Deletes the SNMP destination.
<code>sp get snmp destinations</code>	Displays the available SNMP destinations (IP address or host name) to which the SP is configured to send.
<code>sp get snmp proxy community</code>	Returns the community name currently being used by the SP SNMPD to proxy the platform SNMP agent.
<code>sp set snmp proxy community</code>	Sets the proxy entries that specify the OID to be referred, the IP address to which they are referred, and the community string to use while proxying.
<code>sp get snmp community</code>	Returns the community name currently being used by the service processor.
<code>sp set snmp community</code>	Sets the SNMP community string used by the service processor.

SP Add SNMP-Destination Subcommand

Description: Adds a single SNMP destination (either IP address or host name).

Format

```
sp add snmp-destination IP ADDRESS/HOSTNAME
```

[TABLE H-77](#) lists the argument for this subcommand.

TABLE H-77 Argument for Subcommand `sp add snmp-destination`

Argument	Description
<code>IP ADDRESS/HOSTNAME</code>	Specifies the IP address or name of the host for the destination you wish to add. This argument is repeatable to add multiple destinations at one time; however, the number of destinations you can create is limited due to memory constraints.

Return Codes

[TABLE H-78](#) lists the return codes for this subcommand.

TABLE H-78 Return Codes for Subcommand `sp add snmp-destination`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.
<code>NWSE_RPCTimeout</code>	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
<code>NWSE_InvalidArgument</code>	4	One or more arguments were incorrect or invalid.
<code>NWSE_NoPermission</code>	6	Not authorized to perform this operation.
<code>NWSE_NoMemory</code>	8	Insufficient memory.
<code>NWSE_RPCConnRefused</code>	12	RPC connection refused.
<code>NWSE_UnknownError</code>	15	Miscellaneous error not captured by other errors.
<code>NWSE_FileError</code>	18	File open, file missing, or a read or write error occurred.
<code>NWSE_Exist</code>	19	Entity (user, service or other) already exists.

SP Delete SNMP-Destination Subcommand

Description: Deletes a single SNMP destination (either IP address or host name).

Format

```
sp delete snmp-destination { IP_ADDRESS/HOSTNAME | {-a | --all}  
[-q | --quiet]
```

[TABLE H-79](#) lists the arguments for this subcommand.

TABLE H-79 Arguments for Subcommand `sp delete snmp-destination`

Argument	Description
<code>IP ADDRESS/HOSTNAME</code>	Specifies the IP address or host name of the destination to remove. This argument is repeatable to remove multiple destinations at one time.
<code>[-a --all]</code>	Removes all SNMP destinations.
<code>[-q --quiet]</code>	If the SNMP destination to delete is not found, this argument specifies that no error be returned.

Return Codes

[TABLE H-80](#) lists the return codes for this subcommand.

TABLE H-80 Return Codes for Subcommand `sp delete snmp-destination`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.
<code>NWSE_RPCTimeout</code>	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
<code>NWSE_InvalidArgument</code>	4	One or more arguments were incorrect or invalid.
<code>NWSE_NotFound</code>	5	Entity (user, service, file, path or other) not found.
<code>NWSE_NoPermission</code>	6	Not authorized to perform this operation.
<code>NWSE_NoMemory</code>	8	Insufficient memory.

TABLE H-80 Return Codes for Subcommand `sp delete snmp-destination`

Return Code	ID	Description
NWSE_RPCConnRefused	12	RPC connection refused.
NWSE_UnknownError	15	Miscellaneous error not captured by other errors.
NWSE_FileError	18	File open, file missing, or a read or write error occurred.

SP Get SNMP-Destinations Subcommand

Description: Displays the available SNMP destinations (IP address or host name) to which the SP is configured to send. Many networking programs use this information to identify the machine.

Format

```
sp get snmp-destinations
```

Return Codes

[TABLE H-81](#) lists the return codes for this subcommand.

TABLE H-81 Return Codes for Subcommand `sp get snmp-destinations`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.
NWSE_NoMemory	8	Insufficient memory.
NWSE_UnknownError	15	Miscellaneous error not captured by other errors.
NWSE_FileError	18	File open, file missing, or a read or write error occurred.

SP Get SNMP Proxy Community Subcommand

Description: Returns the community name the SP is currently using to proxy the platform SNMP agent.

Format

```
sp get snmp proxy community
```

Return Codes

[TABLE H-82](#) lists the return codes for this subcommand.

TABLE H-82 Return Codes for Subcommand `sp get snmp proxy community`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.

SP Set SNMP Proxy Community Subcommand

Description: The SNMP agent on the SP acts as a proxy for the master SNMP agent running on the platform. These proxy entries specify the OID to be referred, the IP address to which they are referred, and the community string to use while proxying. The community string is the value configured on the platform-side SNMP configuration.

Format

```
sp set snmp proxy community COMMUNITY STRING
```

TABLE H-83 lists the argument for this subcommand.

TABLE H-83 Argument for Subcommand `sp set snmp proxy community`

Argument	Description
COMMUNITY STRING	Specifies the name of the community to configure.

There are no restrictions on the length of the community strings; common names are “private” and “public.” The default name of the community string is *private*.

If you run the subcommand `sp get snmp proxy community` without setting it, the return value is *private*. Otherwise, you can set it to any string.

Return Codes

TABLE H-84 lists the return codes for this subcommand.

TABLE H-84 Return Codes for Subcommand `sp set snmp proxy community`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request may or may not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.

SP Get SNMP Community Subcommand

Description: Returns the community name that is currently being used by the service processor.

Format

```
sp get snmp community
```

Return Codes

[TABLE H-85](#) lists the return codes for this subcommand.

TABLE H-85 Return Codes for Subcommand `sp get snmp community`

Return Code	ID	Description
NWSE_Success	0	Command completed successfully.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request might or might not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.

SP Set SNMP Community Subcommand

Description: Enables you to set the community name to be used by the service processor itself, as opposed to the proxy community string that is used between the service processor and the platform.

There are no restrictions on the length of the community string. Typical names are “private” and “public.” The factory default name of the community string is “public,” so if you run the command `sp get snmp community` before you set a value, `public` is returned. Set the value to any string without spaces.

Format

```
sp set snmp community COMMUNITY_STRING
```

Return Codes

[TABLE H-86](#) lists the argument for this subcommand.

TABLE H-86 Argument for Subcommand `sp set snmp community`

Argument	Description
COMMUNITY STRING	Specifies the name of the community to configure.

[TABLE H-87](#) lists the return codes for this subcommand.

TABLE H-87 Return Codes for Subcommand `sp set snmp community`

Return Code	ID	Description
NWSE_Success	0	Command completed successfully.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_RPCTimeout	2	Request was issued, but was not serviced by the server. RPC procedure timed out and the request might or might not have been serviced by the server.
NWSE_RPCNotConnected	3	Unable to connect to the RPC server.

SP SSH1 Subcommands

The subcommands in [TABLE H-88](#) manage SSH1 protocol.

TABLE H-88 SP SSH1 Subcommands

Subcommand	Description
<code>sp disable ssh1</code>	Disables protocol version 1 accesses to sshd.
<code>sp enable ssh1</code>	Enables protocol version 1 accesses to sshd.
<code>sp get ssh1</code>	Retrieves the current state of the protocol version 1 activation.

SP Disable SSH1

Description: Disables older less secure protocol version 1 accesses to the secure shell daemon, sshd.

Note – Protocol version 2 remains enabled.

Format

```
sp disable ssh1
```

Successful Output:

```
localhost # sp disable ssh1
```

SP Enable SSH1

Description: Enables protocol version 1 accesses to the secure shell daemon, sshd.

Note – Protocol version 2 remains enabled.

Format

```
sp enable ssh1
```

Successful Output:

```
localhost # sp enable ssh1
ssh protocol version 1 and version 2 accesses allowed
Restarting OpenBSD Secure Shell Server: SSHD
```

SP Get SSH1

Description: Retrieves the current status of the protocol level 1 activation.

Note – The status of SSH1 is set using `SP Disable SSH1` and `SP Enable SSH1` commands.

Format

```
sp get ssh1
```

Successful Output:

```
localhost # sp get ssh1
SSH protocol version 1 and version 2 accesses allowed

or

SSH protocol version 2 accesses allowed. Version 1 NOT allowed!
```

SP SSL Subcommands

The subcommands in [TABLE H-89](#) manage SSL capabilities.

TABLE H-89 SP SSL Subcommands

Subcommand	Description
<code>sp disable ssl-required</code>	Disables forced use of the secure HTTP (HTTPS) protocol.
<code>sp enable ssl-required</code>	Enables forced use of the secure HTTP (HTTPS) protocol.
<code>sp get ssl</code>	Determines whether the Apache Web server is using factory-supplied files or user-supplied files.
<code>sp set ssl</code>	Enables you to use site SSL certificates in the SP environment.

SP Disable SSL-Required Subcommand

Description: Disables automatic redirect to secure HTTP (HTTPS) URLs. With SSL disabled, HTTP requests are serviced directly without redirecting to HTTPS. HTTPS requests continue to be secure.

Format

```
sp disable ssl-required
```

Return Codes

[TABLE H-90](#) lists the return codes for this subcommand.

TABLE H-90 Return Codes for Subcommand `sp disable ssl-required`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.
<code>NWSE_NoPermission</code>	6	Not authorized to perform this operation.

TABLE H-90 Return Codes for Subcommand `sp disable ssl-required` (Continued)

Return Code	ID	Description
NWSE_NoMemory	8	Insufficient memory.
NWSE_FileError	18	File open, file missing, or a read or write error occurred.
NWSE_ServiceNotAvailable	24	Requested service is not available.

SP Enable SSL-Required Subcommand

Description: Enables automatic redirect to secure HTTP (HTTPS) URLs. With SSL enabled, HTTP requests are automatically redirected to equivalent HTTPS requests to maintain site security.

SSL version 0.9.6j is supported.

Format

```
sp enable ssl-required
```

Return Codes

[TABLE H-91](#) lists the return codes for this subcommand.

TABLE H-91 Return Codes for Subcommand `sp enable ssl-required`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoMemory	8	Insufficient memory.
NWSE_FileError	18	File open, file missing, or a read or write error occurred.
NWSE_ServiceNotAvailable	24	Requested service is not available.

SP Get SSL Subcommand

Description: Determines whether automatic redirect to secure HTTP (HTTPS) is required or optional, and whether Apache Web Server is using factory-supplied or user-supplied SSL certificate files.

Format

```
sp get ssl [{-H | noheader}] [{-D | --Delim <DELIMITER>}]
```

[TABLE H-92](#) lists the arguments for this subcommand.

TABLE H-92 Arguments for Subcommand `sp get ssl`

Argument	Description
{ -H --noheader }	Suppresses column headings.
{ -D --Delim }	Delimits columns with the specified delimiter. Headings are also delimited unless suppressed. The delimiter can be any character or string.

Return Codes

[TABLE H-93](#) lists the return codes for this subcommand.

TABLE H-93 Return Codes for Subcommand `sp get ssl`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoMemory	8	Insufficient memory.
NWSE_ServiceNotAvailable	24	Requested service is not available.

SP Set SSL Subcommand

Description: Enables you to use site SSL certificates in the SP environment. This command enables you to replace the Server Certificate in the SP Value-Add image with your own internally generated certificate and to restore the factory settings.

Format

```
sp set ssl [-f]
sp set ssl {-c | --certfile} <FULL PATH OF THE SERVER CERTIFICATE
FILE>
{-k | --keyfile} <FULL PATH OF PRIVATE KEY FILE>
```

[TABLE H-94](#) lists the arguments for this subcommand.

TABLE H-94 Arguments for Subcommand `sp set ssl`

Argument	Description
<code>[-f]</code>	Restores factory settings.
<code>{-c --certfile}</code>	Flags the names of the files to be installed.
<code>{-k --keyfile}</code>	Flags the names of the files to be installed.

Return Codes

[TABLE H-95](#) lists the return codes for this subcommand.

TABLE H-95 Return Codes for Subcommand `sp set ssl`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.
<code>NWSE_NoPermission</code>	6	Not authorized to perform this operation.
<code>NWSE_NoMemory</code>	8	Insufficient memory.
<code>NWSE_FileError</code>	18	File open, file missing, or a read or write error occurred.
<code>NWSE_ServiceNotAvailable</code>	24	Requested service is not available.

SP Update Subcommands

The subcommands in [TABLE H-96](#) manage the SP flash.

TABLE H-96 SP Flash Subcommands

Subcommand	Description
<code>sp update flash all</code>	Sets the update flag to start the full flash update on the next reset of the SP.
<code>sp update flash applications</code>	Copies the file <code>Value-Add</code> to the <code>Value-Add</code> component of the SP flash.
<code>sp update diags</code>	Updates the diagnostics to a newer version.

SP Update Flash All Subcommand

Note – Before using this command you must start the Java Update Server. For instructions on starting Java Update Server, see [“Updating the SP Base Package” on page 32](#).

Description: Updates the entire SP flash image (kernel, base file system, and value-add) as part of a major SP software update.

This subcommand first verifies that the Java `sp update` program is on the specified remote server and that the correct version of the update server is running

Once verified, the subcommand sets the `update` flag to start the full flash update on the next reset of the SP. When the SP boots, it connects with the `sp update` program, downloads and installs the new flash image, and reboots the SP into normal operation mode. It also sets the server IP address and optional server port number in the environment variables.

Run the command `sp -v` from the SP to verify the version of the new flash image.

- If this command fails for the Value-Add image, run the subcommand `sp update flash applications`.
- If this command fails for the Base image, update the flash using the operator panel. For information about the operator panel, refer to the *Sun Fire V20z and Sun Fire V40z Servers—User Guide*.

Note – The subcommand `sp update flash all` does not update `pstore` data.

Format

```
sp update flash all { i | --serverip} <ipaddress xx.xx.xx.xx>
[ { p | --port} <port#> ] [ {-r|--remote} REMOTE_VERSION ]
```

[TABLE H-97](#) lists the arguments for this subcommand.

TABLE H-97 Arguments for Subcommand `sp update flash all`

Argument	Description
{-i --serverip}	The IP address of the remote server on which the update server (Java application) is running. The update server also contains the flash images.
{-p --port}	Optional. The port number on the remote server on which the Java <code>sp update</code> program is listening for SP flash update requests. If the port number is not provided, the command tries to connect to the default port. The default port number is 52708.
{-r --remote}	Identifies the version to be used for the update. Specify a version (for example, v1.2.3.4) or select LATEST to use the latest version available on the update server.

Return Codes

[TABLE H-98](#) lists the return codes for this subcommand.

TABLE H-98 Return Codes for Subcommand `sp update flash all`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_NotFound	5	Requested version was not found.
NWSE_NoPermission	6	Not authorized to perform this operation.
NWSE_NoMemory	8	Insufficient memory.
NWSE_UnknownError	15	Miscellaneous error not captured by other errors.
NWSE_ServiceNotAvailable	24	Requested service is not available.

Downgrades

If you want to downgrade service processor versions, you must use the procedure below to avoid problems when the service processor attempts to boot.

1. Execute `sp reset to default-settings --all`.
2. After the SP reboots:
 - a. Create a new manager user by logging in as the user setup.
 - b. Execute `sp update flash all`.
3. After the SP reboots, execute `sp reset to default-settings --all`.
4. After the SP reboots, create a new manager user and update your configuration as needed. If you are performing this procedure on multiple machines, use the `sp autoconfigure` command from a preconfigured service processor.

If the `sp update flash all` command fails for the Value-Add image, use the `sp update flash applications` command. If this command fails for the Base image, update the flash using the operator panel. See [“Operator Panel” on page 8](#) and the *Sun Fire V20z and Sun Fire V40z Servers—User Guide* for information about using the operator panel.

SP Update Flash Applications Subcommand

Description: The SP file system is divided into two components: Base and Value-Add. The Base component includes the repository and the Value-Add component includes the application software.

This command copies the file `Value-Add` to the Value-Add component of the SP flash. The new Value-Add image takes effect after you reset the SP.

If the subcommand `sp update flash applications` fails and the Value-Add image is corrupted, you can use the similar command that is available in the SP Base image.

Format

```
sp update flash applications [{-f|--filename} FILE]
[{-h|--help}] [{-i|--ipaddress} REMOTE_ADDRESS]
[{-p|--port} REMOTE_PORT] [{-r|--remote} REMOTE_VERSION]
```

TABLE H-99 lists the arguments for this subcommand.

TABLE H-99 Arguments for Subcommand `sp update flash applications`

Argument	Description
<code>{-f --filename}</code>	Specifies the full path of the file.
<code>{-i --ipaddress}</code>	The IP address of the server on which the update server (Java application) is running.
<code>{-p --port}</code>	Optional. The port number on the remote server on which the Java <code>sp update</code> program is listening for SP flash update requests. If the port number is not provided, the command tries to connect to the default port. The default port number is 52708.
<code>{-r --remote}</code>	Identifies the version to be used for the update. Specify a version (for example, v1.2.3.4) or select LATEST to use the latest version available on the update server.

Return Codes

TABLE H-100 lists the return codes for this subcommand.

TABLE H-100 Return Codes for Subcommand `sp update flash applications`

Return Code	ID	Description
<code>NWSE_Success</code>	0	Command successfully completed.
<code>NWSE_InvalidUsage</code>	1	Invalid usage: bad parameter usage, conflicting options specified.
<code>NWSE_NotFound</code>	5	Entity (user, service, file, path or other) was not found.
<code>NWSE_NoPermission</code>	6	Not authorized to perform this operation.
<code>NWSE_NoMemory</code>	8	Insufficient memory.
<code>NWSE_FileError</code>	18	File open, file missing, or a read or write error occurred.
<code>NWSE_ServiceNotAvailable</code>	24	Requested service is not available.
<code>NWSE_DeviceError</code>	25	Unable to read or write to the device.

SP Update Diags Subcommand

Description: Updates the current version of diagnostics, if an update is available.

While the SP functions normally without access to an external file system, a file system is required to enable several features, including diagnostics. The SP software uses a default version of diagnostics. However, if a new version is released and stored on the Network Share Volume, you must explicitly point to that new version to use it.

Format

```
sp update diags {-p | --path} <PATH_TO_DIAGS_FOLDER>
```

[TABLE H-101](#) lists the argument for this subcommand.

TABLE H-101 Argument for Subcommand `sp update diags`

Argument	Description
{-p --path}	Points to the location of the new diagnostics.

Return Codes

[TABLE H-102](#) lists the return codes for this subcommand.

TABLE H-102 Return Codes for Subcommand `sp update diags`

Return Code	ID	Description
NWSE_Success	0	Command successfully completed.
NWSE_InvalidUsage	1	Invalid usage: bad parameter usage, conflicting options specified.
NWSE_InvalidArgument	4	One or more arguments were incorrect or invalid.
NWSE_UnknownError	15	Miscellaneous error not captured by other errors.

Index

A

access commands

- add public key, 131
 - add trust, 127
 - add user, 135
 - delete public keys, 133
 - delete trust, 129
 - delete user, 136
 - directory services subcommands, 122
 - disable service, 122
 - enable service, 123
 - get group, 114, 115, 116, 117
 - get groups, 118
 - get map, 119
 - get public key users, 133
 - get services, 125
 - get trusts, 130
 - get users, 137
 - groups subcommands, 114, 117
 - map, 118, 120
 - public key, 131
 - subcommand groups summary, 113
 - trust, 127
 - unmap, 121
 - update password, 138
 - update user, 139
 - user, 135
- ACPI, 2
- acronym list, 2 to 4
- Active Directory Service (ADS), 43
 - server requirements for, 46
- admin user type, 10
- ADS (Active Directory Service), 2, 43

server requirements for, 46

Agent X for SNMP, 83

alerts, 67

ARP, 2

authentication, 16

troubleshooting, 76

autoconfiguring SP, 36

B

back panel connectors, 5

on Sun Fire V20z, 5

baseboard management controller, 2

baseboard management controller (BMC), 58

BIOS

changing settings for console redirection, 103

POST codes table, 223

setting up, 41

BMC, 2

boot block codes for flash ROM, 228

buttons, on operator panel, 9

C

checkmark, on operator panel display, 9

command-line interface, 6

commands

command type overview table, 110

return codes summary table, 111

sshz, using protocol, 109

UNIX, xiv

See also command names

community name for SNMP, 82

- configuration
 - date and time, 47
 - directory services, 43
 - SMTP event notification, 42
 - SSL certificate, 48
- configuring, service processor, 12 to 23
- connection problems, troubleshooting, 76
- connectors, 5
 - on Sun Fire V20z, 5
 - on Sun Fire V40z, 5
- console redirection over serial
 - configuring BIOS settings for, 103
 - getty, using, 101
 - grub, using, 99
 - LILO, using, 101
 - overview, 98
 - securetty, using, 102
- console, Systems Management, 39
- CRU, 2

D

- daisy-chain server configuration, 24
- date and time settings, 47
- default gateway, 14
- diags commands
 - cancel tests, 142
 - get modules, 143
 - get state, 144
 - get tests, 145
 - run tests, 146
 - start, 148
 - subcommands overview, 141
 - terminate, 150
- DIMMs, memory sensors for, 63
- directory services
 - configuring, 43
 - mapping groups, 45
- documentation
 - providing feedback on, xvi
 - related, xv
- downgrades, SP, 261
- DPC, 2
- drivers
 - IPMI Linux kernel device driver, 73
 - platform, 27

E

- email configuration, 42
- enabling
 - IPMI access, 18
 - IPMI LAN access, 21
- escape sequences, remote console, 108
- event filtering, 60
- event logging disabled sensor, 65
- events
 - icons, 55
 - system, 53

F

- fan sensors, 63
- flash ROM boot block codes, 228
- FRU, 2

G

- getty, using for console redirection, 101
- graceful shutdown, 66
- group mapping, 45
- grub, 2
- grub, using for console redirection, 99

H

- HDT, 2
- host-key pairs, 93
 - generating on Microsoft Windows, 95 to 96
 - scripting, 91

I

- ICMB, 2
- icons, 55
- initial manager account, creating, 16
- integrating servers, 23
- integration of SNMP protocol, 78
- Intelligent Platform Management Interface. *See* IPMI interface; IPMItool
- interconnecting servers, diagram, 24
- interfaces supported, for server management, 6
- inventory commands
 - compare versions, 152
 - get all, 158
 - get hardware, 153
 - get software, 156, 157

- subcommand summary table, 151
- IP address
 - DHCP setting, 13 to 14
 - static, 14 to 16
- IPMB, 2
- IPMI, 1, 3
- IPMI access
 - enabling, 18
 - in-band enabling on Linux server, 18
 - in-band enabling on Solaris x86 server, 20
 - upgrading kernel and, 22 to 23
- ipmi commands
 - disable channel, 160
 - disable pef, 162
 - enable channel, 161
 - enable pef, 162
 - get channels, 163
 - get global enables, 164
 - get sel command, 164
 - reset, 168
 - set global enable, 167
 - subcommand summary table, 159
- IPMI interface
 - baseboard management controller, 58
 - compliance standards, 60
 - IPMItool, 67
 - LAN channel access, 60
 - LAN interface for BMC, 74
 - lights out management, 67
 - Linux kernel device driver, 73
 - manageability features, 59
 - overview, 57
 - system event log, 61, 75
 - troubleshooting, 76
- IPMI kernel, upgrading, 22 to 23
- IPMI LAN access
 - enabling, 21
 - in-band enabling on Linux server, 21
 - in-band enabling on Solaris x86 server, 21
 - out-of-band enabling on Linux server, 22
- IPMItool
 - command expressions and parameters, 69 to 73
 - command options, 68
 - command syntax, 68
 - download sources, 67

K

- KCS, 3
- keytab file, 46
- KVM, 3

L

- LAN, 3
 - channels, 60
 - diagram, 24
- language support, 27
- lights out management, IPMI, 67
- LILO, 3
 - using for console redirection, 101
- Linux kernel
 - device driver for, 73
 - upgrading, 22 to 23
- Linux software, on service processor, 4
- logging in, with setup account, 16
- LOM, 3
 - See also* lights out management

M

- MAC addresses, determining, 39
- management information base (MIB) for SNMP, 79
- management-controller sensors, 64
- manager user type, 10
- mapping directory services groups, 45
- memory sensors, for DIMMs, 63
- MIB, 3
 - browser, 83
 - tree diagram, 79
- monitor user type, 10
- monitoring server, 59

N

- network settings, assigning using DHCP, 13 to 14
- network share volume
 - extracted content, 105
 - structure, 105
- NIS, 43
- NNM, 3
- NPS, 3
- null user, 60

O

- octets, on operator panel display, 9
- operating system states, 41
- operator panel buttons
 - functions defined, 9
 - illustration, 8
- operator panel display, 9
- overview of book, xiii

P

- passwords, 16
 - files, 10
 - rules for, 16
 - troubleshooting problems with, 76
- PEF, 3
- PET, 3
- platform commands
 - console, 172
 - console subcommands summary, 172
 - get console, 175
 - get hostname, 190, 191
 - get os state, 180
 - get power state, 187
 - get product-id, 192
 - os state subcommands summary, 179
 - power state subcommand summary, 187
 - set console, 177
 - set power state, 188
 - subcommand summary table, 171
- platform drivers, 27
 - graceful shutdown and, 66
- platform event trap alerts, 60, 67
- platform MAC address, 39
- platform operations, 41
- POST codes table, 223
- power, 41
- power state, 27
- power-supply sensors, 64
- propagating SP settings, 36
- PRS, 3
- public keys
 - managing, 131 to 132
 - for scripting, 93
- PXE, 3

R

- recovery, 59
- related documentation, xv
- remote console escape sequences, 108
- restart, 41
- return codes summary table, 111
- RMCP, 3
- RPM, 3
- RTC, 3
- rules, for user names and passwords, 16

S

- scripting, 6
- scripts, using
 - command output, 97
 - generating host-key pairs, 91, 93
 - guidelines, 96
 - multiple system configuration, 91
 - overview, 89
 - public keys, adding, 93
 - remote scripting with SSH, 91
 - shell scripts overview, 89
 - SSH access using public keys, 96
 - SSH access using trusted hosts, 95
 - tips for best results, 98
 - trusted-host relationship, 92
- SDR, 3
- SDRR, 3
- Secure Shell (SSH)
 - access using public keys, 96
 - access using trusted hosts, 95
 - using for remote scripting, 91
- securetty, using for console redirection, 102
- SEL, 3
- sensor commands
 - get, 194
 - set, 197
 - subcommand summary table, 193
- Sensor Data Record Repository (SDRR), 59, 62
- sensors, 62 to 65
- serial over LAN feature
 - disabling, 107
 - enabling, 106
 - launching and terminating sessions, 107
- serial-over-LAN, 60
- server interconnections, 24

- server management
 - interfaces for, 6
 - overview, 4, 7
- service processor, 1
 - assigning network settings with DHCP, 13 to 14
 - assigning static network settings, 14 to 16
 - autoconfiguration, 36
 - commands. *See* sp commands
 - configuring, 12 to 23
 - features requiring drivers installed, 27
 - Linux software on, 4
 - MAC address, 39
 - securing with accounts, 16
 - SNMP agent, 81
 - updating SP software, 28
- service user type, 10
- setup account, logging in, 16
- shell scripts, using, 89
- simple network management protocol. *See* SNMP interface
- site integration, 23
- SM console features, 39
- SMS channels, 60
- SMTP, 3
- SMTP event notification, 42
- SNMP, 1, 3, 6
- SNMP interface
 - agent on the SP, 81
 - Agent X, 83
 - architecture diagram, 80
 - community name, setting, 82
 - configuring, 80
 - integration overview, 78
 - logging options, setting, 84
 - management information base (MIB), 79
 - MIB details, 86
 - overview, 77
 - prerequisites, 80
 - proxy agent, 82
 - server event trap destinations, 85
 - server event traps, 84
 - SP events table, 87
 - third-party MIB browser, 83
 - troubleshooting, 88
- SOL, 3
- SP, 3
 - downgrading, 261
- sp commands
 - add mount, 237
 - add snmp-destination, 247
 - create test events, 220
 - date subcommands summary, 200
 - delete event, 205
 - delete mount, 238
 - delete snmp-destination, 248
 - disable dns, 202
 - disable ssh1, 253
 - disable ssl-required, 255
 - dns subcommands summary, 202
 - enable dns, 203
 - enable ssh1, 254
 - enable ssl-required, 256
 - get date, 200
 - get dns, 204
 - get events, 206
 - get hostname, 208
 - get ip, 211
 - get jnet, 213
 - get locatelight, 216
 - get logfile, 217
 - get mount, 239
 - get port80, 222
 - get smtp server, 241
 - get smtp subscribers, 243
 - get snmp proxy community, 250
 - get snmp-destinations, 249
 - get ssh1, 254
 - get ssl, 257
 - get status, 231
 - get tdulog, 231
 - hostname subcommand summary table, 208
 - ip subcommands summary, 210
 - jnet address subcommand summary, 213
 - load settings, 229
 - locatelight subcommand summary, 215
 - logfile subcommand summary, 217
 - miscellaneous subcommand summary, 220
 - mount subcommands summary, 236
 - reboot, 233
 - reset, 234
 - set date, 201
 - set hostname, 209
 - set ip, 212
 - set jnet, 214
 - set locatelight, 216
 - set logfile, 218

- set smtp server, 242
- set snmp proxy community, 250
- set ssl, 257
- smtp subcommands summary, 241
- snmp subcommands summary, 246
- sp events subcommand summary, 205
- ssh1 subcommands summary, 253
- ssl subcommands summary, 255
- subcommand group summary table, 199
- update diags, 262
- update flash all, 259
- update flash applications, 261
- update smtp subscriber, 244
- update subcommands summary, 259

SSH (Secure Shell), 6

ssh command, 60

ssh command protocol, 109

SSL, 3

SSL Certificate, 48

SSU, 3

states, 41

subnet mask, 14

summary, of command types, 110

SunMC, 3

support, contacting, xvi

system event log, 59, 75

- IPMI, 61

system events, 53

system-event sensor, 64

system-firmware progress sensor, 65

Systems Management console features, 39

systems management tasks, 11 to 12

T

tasks. *See* systems management tasks

technical support, contacting, xvi

temperature sensors, 63

time and date settings, 47

translations, of documentation, xv

traps, server events with SNMP, 84

troubleshooting

- IPMI, 76
- SNMP, 88

Troubleshooting Dump Utility (TDU), 231

trusted-host relationship, scripting, 92

type icons, 55

typographic conventions, xiv

U

UART, 3

UDP, 3

UMB, 3

UNIX commands, xiv

updating service processor software, 28

user groups, defined, 10

user names, 16

- rules for, 16

user types, defined, 10

UTC, 4

V

vital product data (VPD), 59

voltage sensors, 63

VPD, 4

W

WAN, 4

Watchdog 2 sensor, 65

watchdog timer, 59, 66

X

X, on operator panel display, 9