Part No. 060318-10, Rev. E
September 2012

OmniSwitch AOS Release 7 Switch Management Guide

Alcatel·Lucent

www.alcatel-lucent.com

This user guide documents AOS Release 7 for the OmniSwitch 10K and OmniSwitch 6900.
The functionality described in this guide is subject to change without notice.

26801 West Agoura Road
Calabasas, CA 91301
(818) 880-3500 FAX (818) 880-3505
support@ind.alcatel.com

US Customer Support—(800) 995-2696
International Customer Support—(818) 878-4507
Internet—service.esd.alcatel-lucent.com

Contents

About This Guide

This *OmniSwitch AOS Release 7 Switch Management Guide* describes basic attributes of your switch and basic switch administration tasks. The software features described in this manual are shipped standard with your switches. These features are used when readying a switch for integration into a live network environment.

Supported Platforms

This information in this guide applies only to the OmniSwitch 10K and OmniSwitch 6900 switches.

Who Should Read this Manual?

The audience for this user guide are network administrators and IT support personnel who need to configure, maintain, and monitor switches and routers in a live network. However, anyone wishing to gain knowledge on how fundamental software features are implemented in the OmniSwitch Series switches will benefit from the material in this configuration guide.

When Should I Read this Manual?

Read this guide as soon as your switch is up and running and you are ready to familiarize yourself with basic software functions. You should have already stepped through the first login procedures and read the brief software overviews in the *Getting Started Guide*.

You should have already set up a switch password and be familiar with the very basics of the switch software. This manual will help you understand the switch's directory structure, the Command Line Interface (CLI), configuration files, basic security features, and basic administrative functions. The features and procedures in this guide will help form a foundation that will allow you to configure more advanced switching features later.

What is in this Manual?

This configuration guide includes information about the following features:

- Basic switch administrative features, such as file editing utilities, procedures for loading new software, and setting up system information (name of switch, date, time).
- Configurations files, including snapshots, off-line configuration, time-activated file download.
- The CLI, including on-line configuration, command-building help, syntax error checking, and line editing.
- Basic security features, such as switch access control and customized user accounts.
- SNMP
- Web-based management (WebView)

What is Not in this Manual?

The configuration procedures in this manual primarily use Command Line Interface (CLI) commands in examples. CLI commands are text-based commands used to manage the switch through serial (console port) connections or via Telnet sessions. This guide does include introductory chapters for alternative methods of managing the switch, such as web-based (WebView) and SNMP management. However the primary focus of this guide is managing the switch through the CLI.

Further information on WebView can be found in the context-sensitive on-line help available with that application.

This guide does not include documentation for the OmniVista network management system. However, OmniVista includes a complete context-sensitive on-line help system.

This guide provides overview material on software features, how-to procedures, and tutorials that will enable you to begin configuring your OmniSwitch. However, it is not intended as a comprehensive reference to all CLI commands available in the OmniSwitch. For such a reference to all CLI commands, consult the *OmniSwitch CLI Reference Guide*.

How is the Information Organized?

Each chapter in this guide includes sections that will satisfy the information requirements of casual readers, rushed readers, serious detail-oriented readers, advanced users, and beginning users.

Quick Information. Most chapters include a *specifications table* that lists RFCs and IEEE specifications supported by the software feature. In addition, this table includes other pertinent information such as minimum and maximum values and sub-feature support. Some chapters include a *defaults table* that lists the default values for important parameters along with the CLI command used to configure the parameter. Many chapters include *Quick Steps* sections, which are procedures covering the basic steps required to get a software feature up and running.

In-Depth Information. All chapters include *overview sections* on software features as well as on selected topics of that software feature. *Topical sections* may often lead into *procedure sections* that describe how to configure the feature just described. Many chapters include *tutorials* or *application examples* that help convey how CLI commands can be used together to set up a particular feature.

Documentation Roadmap

The OmniSwitch user documentation suite was designed to supply you with information at several critical junctures of the configuration process.The following section outlines a roadmap of the manuals that will help you at each stage of the configuration process. Under each stage, we point you to the manual or manuals that will be most helpful to you.

Stage 1: Using the Switch for the First Time

Pertinent Documentation: *OmniSwitch Getting Started Guide*
Release Notes

A hard-copy *OmniSwitch 10K Getting Started Guide* is included with your switch; this guide provides all the information you need to get your switch up and running the first time. It provides information on unpacking the switch, rack mounting the switch, installing NI modules, unlocking access control, setting the switch's IP address, and setting up a password. It also includes succinct overview information on fundamental aspects of the switch, such as hardware LEDs, the software directory structure, CLI conventions, and web-based management.

At this time you should also familiarize yourself with the Release Notes that accompanied your switch. This document includes important information on feature limitations that are not included in other user guides.

Stage 2: Gaining Familiarity with Basic Switch Functions

Pertinent Documentation: *OmniSwitch Hardware Users Guide*
OmniSwitch AOS Release 7 Switch Management Guide

Once you have your switch up and running, you will want to begin investigating basic aspects of its hardware and software. Information about switch hardware is provided in the *OmniSwitch 10K Hardware Guide*. This guide provide specifications, illustrations, and descriptions of all hardware components, such as chassis, power supplies, Chassis Management Modules (CMMs), Network Interface (NI) modules, and cooling fans. It also includes steps for common procedures, such as removing and installing switch components.

This guide is the primary users guide for the basic software features on a single switch. This guide contains information on the switch directory structure, basic file and directory utilities, switch access security, SNMP, and web-based management. It is recommended that you read this guide before connecting your switch to the network.

Stage 3: Integrating the Switch Into a Network

Pertinent Documentation: *OmniSwitch AOS Release 7 Network Configuration Guide*
OmniSwitch AOS Release 7 Advanced Routing Configuration Guide

When you are ready to connect your switch to the network, you will need to learn how the OmniSwitch implements fundamental software features, such as 802.1Q, VLANs, Spanning Tree, and network routing protocols. This guide contains overview information, procedures, and examples on how standard networking technologies are configured on the OmniSwitch.

The guide includes configuration information for networks using advanced routing technologies (OSPF and BGP) and multicast routing protocols (DVMRP and PIM-SM).

Anytime

The *OmniSwitch CLI Reference Guide* contains comprehensive information on all CLI commands supported by the switch. This guide includes syntax, default, usage, example, related CLI command, and CLI-to-MIB variable mapping information for all CLI commands supported by the switch. This guide can be consulted anytime during the configuration process to find detailed and specific information on each CLI command.

Related Documentation

The following are the titles and descriptions of all the related OmniSwitch user manuals:

- *OmniSwitch 10K* and *OmniSwitch 6900 Getting Started Guides*

  Describes the hardware and software procedures for getting an OmniSwitch up and running. Also provides information on fundamental aspects of OmniSwitch software architecture.

- *OmniSwitch 10K* and *OmniSwitch 6900 Hardware Users Guides*

  Complete technical specifications and procedures for all OmniSwitch chassis, power supplies, fans, and Network Interface (NI) modules.

- *OmniSwitch CLI Reference Guide*

  Complete reference to all CLI commands supported on the OmniSwitch. Includes syntax definitions, default values, examples, usage guidelines and CLI-to-MIB variable mappings.

- *OmniSwitch AOS Release 7 Switch Management Guide*

  Includes procedures for readying an individual switch for integration into a network. Topics include the software directory architecture, image rollback protections, authenticated switch access, managing switch files, system configuration, using SNMP, and using web management software (WebView).

- *OmniSwitch AOS Release 7 Network Configuration Guide*

  Includes network configuration procedures and descriptive information on all the major software features and protocols included in the base software package. Chapters cover Layer 2 information (Ethernet and VLAN configuration), Layer 3 information (routing protocols, such as RIP and IPX), security options (authenticated VLANs), Quality of Service (QoS), link aggregation, and server load balancing.

- *OmniSwitch AOS Release 7 Advanced Routing Configuration Guide*

  Includes network configuration procedures and descriptive information on all the software features and protocols included in the advanced routing software package. Chapters cover multicast routing (DVMRP and PIM-SM), Open Shortest Path First (OSPF), and Border Gateway Protocol (BGP).

- *OmniSwitch Transceivers Guide*

  Includes SFP and XFP transceiver specifications and product compatibility information.

- Technical Tips, Field Notices

  Includes information published by Alcatel's Customer Support group.

- *Release Notes*

  Includes critical Open Problem Reports, feature exceptions, and other important information on the features supported in the current release and any limitations to their support.

Technical Support

An Alcatel-Lucent service agreement brings your company the assurance of 7x24 no-excuses technical support. You'll also receive regular software updates to maintain and maximize your Alcatel-Lucent product's features and functionality and on-site hardware replacement through our global network of highly qualified service delivery partners.

With 24-hour access to Alcatel-Lucent's Service and Support web page, you'll be able to view and update any case (open or closed) that you have reported to Alcatel-Lucent's technical support, open a new case or access helpful release notes, technical bulletins, and manuals.

Access additional information on Alcatel-Lucent's Service Programs:

Web: service.esd.alcatel-lucent.com

Phone: 1-800-995-2696

Email: esd.support@alcatel-lucent.com

1 Logging Into the Switch

Logging into the switch may be done locally or remotely. Management tools include: the Command Line Interface (CLI), which may be accessed locally via the console port, or remotely via Telnet; WebView, which requires an HTTP client (browser) on a remote workstation; and SNMP, which requires an SNMP manager (such as Alcatel-Lucent's OmniVista or HP OpenView) on the remote workstation. Secure sessions are available using the Secure Shell interface.

In This Chapter

This chapter describes the basics of logging into the switch to manage the switch through the CLI. It also includes the information about using Telnet, and Secure Shell for logging into the switch as well as information about using the switch to start a Telnet or Secure Shell session on another device. It also includes information about managing sessions and specifying a DNS resolver. For more details about the syntax of referenced commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

Management access is disabled (except through the console port) unless specifically enabled by a network administrator. For more information about management access and methods, use the table here as a guide:

For more information about...	See...
Enabling or "unlocking" management interfaces on the switch	Chapter 7, "Managing Switch Security"
Authenticating users to manage the switch	Chapter 7, "Managing Switch Security"
Creating user accounts directly on the switch	Chapter 6, "Managing Switch User Accounts"
Using the CLI	Chapter 4, "Using the CLI"

For more information about...	See...
Using WebView to manage the switch	Chapter 8, “Using WebView”
Using SNMP to manage the switch	Chapter 9, “Using SNMP”

Login Specifications

Platforms Supported	OmniSwitch 10K, 6900
Login Methods	Telnet, SSH, HTTP, SNMP
Number of concurrent Telnet sessions	**4**
Number of concurrent SSH sessions	**8**
Number of concurrent HTTP (WebView) sessions	**4**
Secure Shell public key authentication	Password DSA/RSA Public Key

Login Defaults

Access to managing the switch is always available for the **admin** user through the console port, even if management access to the console port is disabled.

Parameter Description	Command	Default
Session login attempts allowed before the TCP connection is closed.	**session login-attempt**	3 attempts
Time-out period allowed for session login before the TCP connection is closed.	**session login-timeout**	55 seconds
Inactivity time-out period. The length of time the switch can remain idle during a login session before the switch will close the session.	**session timeout**	4 minutes

Quick Steps for Logging Into the Switch

The following procedure assumes that you have set up the switch as described in your *OS10K Getting Started Guide* and *Hardware Users Guide*. Setup includes:

- Connecting to the switch via the console port.
- Setting up the Ethernet Management Port (EMP).
- Enabling (or "unlocking") management interfaces types through the **aaa authentication** command for the interface you are using. For detailed information about enabling session types, see Chapter 7, "Managing Switch Security."

1 If you are connected to the switch via the console port, your terminal will automatically display the switch login prompt. If you are connected remotely, you must enter the switch IP address in your remote session. The login prompt then displays.

2 At the login prompt, enter the **admin** username. At the password prompt, enter the **switch** password. (Alternately, you may enter any valid username and password.) The switch's default welcome banner will display, followed by the CLI prompt.

```
Welcome to the Alcatel-Lucent OS10K
Software Version 7.1.1.733.R01 Development, April 05, 2010.

Copyright(c), 1994-2010 Alcatel-Lucent. All Rights reserved.

OmniSwitch(TM) is a trademark of Alcatel-Lucent registered
in the United States Patent and Trademark Office.
```

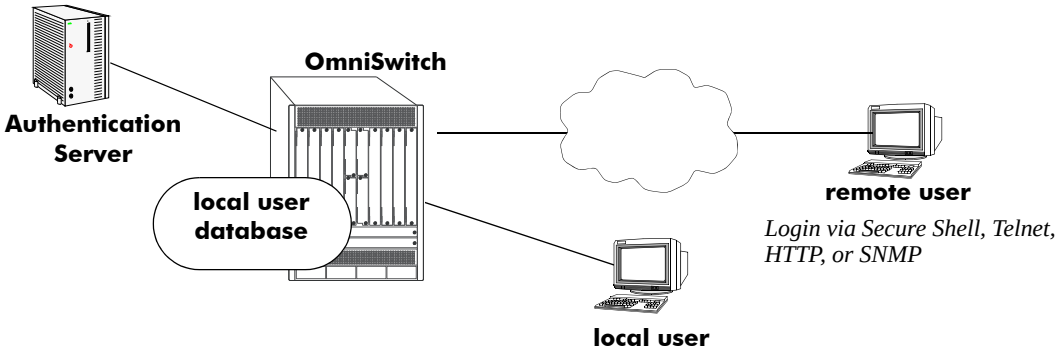
You are now logged into the CLI. For information about changing the welcome banner, see "Modifying the Login Banner" on page 1-14.

For information about changing the login prompt, see Chapter 4, "Using the CLI."

For information about setting up additional user accounts locally on the switch, see Chapter 6, "Managing Switch User Accounts."

Overview of Switch Login Components

Switch access components include access methods (or interfaces) and user accounts stored on the local user database in the switch and/or on external authentication servers. Each access method, except the console port, must be enabled or “unlocked” on the switch before users can access the switch through that interface.



Switch Login Components

Management Interfaces

Logging into the switch may be done locally or remotely. Remote connections may be secure or insecure, depending on the method. Management interfaces are enabled using the **aaa authentication** command. This command also requires specifying the external servers and/or local user database that will be used to authenticate users. The process of authenticating users to manage the switch is called Authenticated Switch Access (ASA). Authenticated Switch Access is described in detail in Chapter 7, “Managing Switch Security.”

An overview of management methods is listed here:

Logging Into the CLI

- **Console port**—A direct connection to the switch through the console port. The console port is always enabled for the default user account, see “Configuring the Console Port” on page 1-6.
- **EMP Port**—The Ethernet Management Port (EMP) allows you to bypass the Network Interface (NI) modules and remotely manage the switch directly through the CMM., see “Setting the EMP Port’s IP Address” on page 1-7
- **Telnet**—Any standard Telnet client may be used for remote login to the switch. This method is not secure. For more information about using Telnet to access the switch, see “Using Telnet” on page 1-8.
- **Secure Shell**—Any standard Secure Shell client may be used for remote login to the switch. See “Using Secure Shell” on page 1-9.

Using the WebView Management Tool

- **HTTP**—The switch has a Web browser management interface for users logging in via HTTP. This management tool is called WebView. For more information about using WebView, see Chapter 8, "Using WebView."

Using SNMP to Manage the Switch

- **SNMP**—Any standard SNMP application may be used for configuring the switch. See Chapter 9, "Using SNMP."

User Accounts

User accounts may be configured and stored directly on the switch, and user accounts may also be configured and stored on an external authentication server or servers.

The accounts include a username and password. In addition, they also specify the user's privileges or end-user profile, depending on the type of user account. In either case, the user is given read-only or read-write access to particular commands.

- **Local User Database**

The **user** command creates accounts directly on the switch. See Chapter 6, "Managing Switch User Accounts,"for information about creating accounts on the switch.

- **External Authentication Servers**

The switch may be set up to communicate with external authentication servers that contain user information. The user information includes usernames and passwords; it may also include privilege information or reference an end-user profile name.

For information about setting up the switch to communicate with external authentication servers, see the *OmniSwitch AOS Release 7 Network Configuration Guide*.

Configuring the Console Port

The console port default settings are listed in the *Hardware Users Guide*. If you wish to modify the default serial connection settings (i.e., baud rate, parity, data bits, stop bits, and mode), use the **modify boot parameters** command as shown:

```
-> modify boot parameters

Boot > boot serialbaudrate 19200
Boot > boot serialparity even
Boot > boot serialwordsize 7
Boot > boot serialstopbits 2
Boot > boot serialmode modemControlOn

Boot > show
Serial (console) baud: 19200
Serial (console) parity: even
Serial (console) wordsize: 7
Serial (console) stopbits: 2
Serial (console) mode: modemControlOn

Boot > commit system
Boot > commit boot
Boot > exit
```

- Output to the terminal may become illegible due to incompatible serial connection settings between the switch and the terminal emulation software.
- If you use the **commit system** command only, changes will not be saved to the switch's non-volatile memory and will be lost if the switch is rebooted.

Setting the EMP Port's IP Address

In order to access the switch through the EMP port the port's default IP and network mask should be changed. There are multiple IP addresses to consider when configuring the EMP port.

- The EMP IP address shared between both CMMs, stored in the **boot.cfg** file.
- The Primary or Secondary's CMM's IP address, stored in NVRAM. (Not required for remote access)

Only the shared EMP IP address stored in the **boot.cfg** file is required for remote access to the switch. However, in some troubleshooting scenarios having an IP address associated to a specific CMM may be helpful. The following should be noted if configuring an IP address stored in NVRAM:

- All the EMP IP addresses and CMM's IP addreses must be in the same subnet.
- Each of the IP addresses must be unique.
- Changes stored in NVRAM will remain with the CMM if the CMM is moved to a different chassis.

Modifying the Shared EMP IP Address

Use the **ip interface** command to modify the shared EMP IP address as shown below.

```
-> ip interface emp address 198.51.100.100 mask 255.255.0.0
```

Changes made using the **ip interface** command are stored in the boot.cfg file.

Modifying the Primary or Secondary CMM's EMP Port IP Address

Must be connected to the associated CMM's console port before attempting to change IP address information using the **modify boot parameters** command as shown below:

```
-> modify boot parameters

Boot > boot empipaddr 198.51.100.2
Boot > boot empmasklength 16
Boot > show

EMP IP Address: 198.51.100.2/16
(additional table output not shown)

Boot > commit system
Boot > commit boot
Boot > exit
```

- If you use the **commit system** command only, changes will not be saved to the switch's non-volatile memory and will be lost if the switch is rebooted.

Using Telnet

Telnet may be used to log into the switch from a remote station. All of the standard Telnet commands are supported by software in the switch. When Telnet is used to log in, the switch acts as a Telnet server. If a Telnet session is initiated from the switch itself during a login session, then the switch acts as a Telnet client.

Logging Into the Switch Via Telnet

Before you can log into the switch using a Telnet interface, the **telnet** option of the **aaa authentication** command must be enabled. Once enabled, any standard Telnet client may be used to log into the switch. To log into the switch, open your Telnet application and enter the switch's IP address (the IP address will typically be the same as the one configured for the EMP). The switch's welcome banner and login prompt is displayed.

Note. A Telnet connection is not secure. Secure Shell is recommended instead of Telnet.

Starting a Telnet Session from the Switch

At any time during a login session on the switch, you can initiate a Telnet session to another switch (or some other device) by using the **telnet** CLI command and the relevant IP address or hostname.

The following shows an example of telnetting to another OmniSwitch:

```
-> telnet 198.51.100.100
Trying 198.51.100.100...
Connected to 198.51.100.100
Escape character is '^]'.
login : admin
password :

Welcome to the Alcatel-Lucent OS10K
Software Version 7.1.1.733.R01 Development, August 05, 2010.

Copyright(c), 1994-2010 Alcatel-Lucent. All Rights reserved.

OmniSwitch(TM) is a trademark of Alcatel-Lucent registered
in the United States Patent and Trademark Office.
```

Using Secure Shell

The Secure Shell feature provides a secure mechanism that allows you to log in to a remote switch, to execute commands on a remote device, and to move files from one device to another. Secure Shell provides secure, encrypted communications even when your transmission is between two untrusted hosts or over an unsecure network. Secure Shell protects against a variety of security risks including the following:

- IP spoofing
- IP source routing
- DNS spoofing
- Interception of clear-text passwords and other data by intermediate hosts
- Manipulation of data by users on intermediate hosts

Secure Shell Components

The OmniSwitch includes both client and server components of the Secure Shell interface and the Secure Shell FTP file transfer protocol. SFTP is a subsystem of the Secure Shell protocol. All Secure Shell FTP data are encrypted through a Secure Shell channel.

Since Secure Shell provides a secure session, the Secure Shell interface and SFTP are recommended instead of the Telnet program or the FTP protocol for communications over TCP/IP for sending file transfers. Both Telnet and FTP are available on the OmniSwitch but they do not support encrypted passwords.

Secure Shell Interface

The Secure Shell interface is invoked when you enter the **ssh** command. After the authentication process between the client and the server is complete, the remote Secure Shell interface runs in the same way as Telnet.

Secure Shell File Transfer Protocol

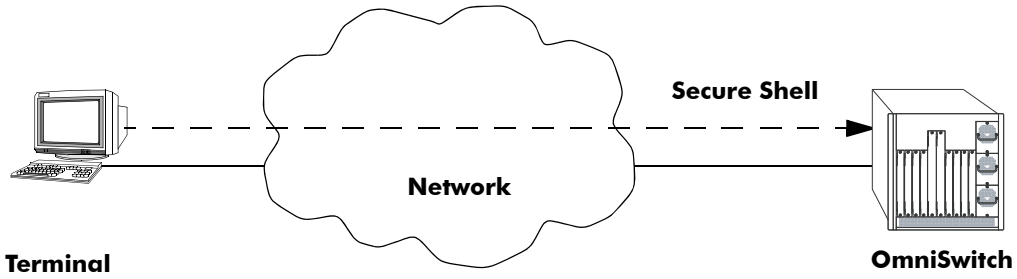
Secure Shell FTP is the standard file transfer protocol used with Secure Shell. Secure Shell FTP is an interactive file transfer program (similar to the industry standard FTP) which performs all file transfer operations over a Secure Shell connection.

You can invoke the Secure Shell FTP session by using the **sftp** command. Once the authentication phase is complete, the Secure Shell FTP subsystem runs. Secure Shell FTP connects and logs into the specified host, then enters an interactive command mode. Refer to "Starting a Secure Shell Session from the OmniSwitch" on page 1-13 for detailed information.

Secure Shell Application Overview

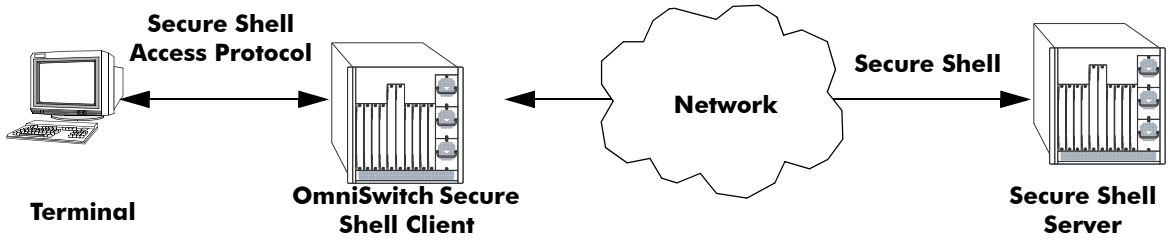
Secure Shell is an access protocol used to establish secured access to your OmniSwitch. The Secure Shell protocol can be used to manage an OmniSwitch directly or it can provide a secure mechanism for managing network servers through the OmniSwitch.

The drawing below illustrates the Secure Shell being used as an access protocol replacing Telnet to manage the OmniSwitch. Here, the user terminal is connected through the network to the switch.



Secure Shell Used as an Access Protocol

The drawing below shows a slightly different application. Here, a terminal connected to a single switch, which acts as a Secure Shell client is an entry point to the network. In this scenario, the client portion of the Secure Shell software is used on the connecting switch and the server portion of Secure Shell is used on the switches or servers being managed.



OmniSwitch as a Secure Shell Client

Secure Shell Authentication

Secure Shell authentication is accomplished in several phases using industry standard algorithms and exchange mechanisms. The authentication phase is identical for Secure Shell and Secure Shell FTP. The following sections describe the process in detail.

Protocol Identification

When the Secure Shell client in the OmniSwitch connects to a Secure Shell server, the server accepts the connection and responds by sending back an identification string. The client will parse the server's identification string and send an identification string of its own. The purpose of the identification strings is to validate that the attempted connection was made to the correct port number. The strings also declare the protocol and software version numbers. This information is needed on both the client and server sides for debugging purposes.

At this point, the protocol identification strings are in human-readable form. Later in the authentication process, the client and the server switch to a packet-based binary protocol, which is machine readable only.

Algorithm and Key Exchange

The OmniSwitch Secure Shell server is identified by one or several host-specific keys. Both the client and server process the key exchange to choose a common algorithm for encryption, signature, and compression. This key exchange is included in the Secure Shell transport layer protocol. It uses a key agreement to produce a shared secret that cannot be determined by either the client or the server alone. The key exchange is combined with a signature and the host key to provide host authentication. Once the exchange is completed, the client and the server turn encryption on using the selected algorithm and key. The following elements are supported:

Host Key Type	DSA/RSA
Cipher Algorithms	AES, Blowfish, Cast, 3DES, Arcfour, Rijndael
Signature Algorithms	MD5, SHA1
Compression Algorithms	None Supported
Key Exchange Algorithms	diffie-hellman-group-exchange-sha1 diffie-hellman-group1-sha1
Key Location	/flash/system
Key File Names	Public - ssh_host_key.pub, ssh_host_dsa_key.pub, ssh_host_rsa_key.pub Private - ssh_host_key, ssh_host_dsa_key, ssh_host_rsa_key

Note. The OmniSwitch contains host keys by default. The keys on the switch are made up of two files contained on **flash**, a private key and a public key. To generate a different key, use the Secure Shell tools available on your Unix or Windows system and copy the files to the OmniSwitch. The new keys will take effect after the OmniSwitch is rebooted.

Authentication Phase

When the client tries to authenticate, the server determines the process used by telling the client which authentication methods can be used. The client has the freedom to attempt several methods listed by the server. The server will disconnect itself from the client if a certain number of failed authentications are attempted or if a time-out period expires. Authentication is performed independent of whether the Secure Shell interface or the SFTP file transfer protocol will be implemented.

Connection Phase

After successful authentication, both the client and the server process the Secure Shell connection protocol.

Using Secure Shell Public Key Authentication (PKA)

Generating and copying Keys

The following procedure is used to set up Secure Shell PKA between an OmniSwitch and a client device. The steps below use a *userid* of "new_ssh_user" on the OmniSwitch as an example:

Note: A comment must be provided when generating the public key (ex. remote_ssh_user@device) and the key must be in the format show below**.**
<ssh-rsa | ssh-dsa> <encrypted key> <remote_ssh_user@device>

Example Key:

```
ssh-rsa AAAAB3NzaC1yc2EAAkjgnivubn9872435nsdg8dfsgfd8dfgfd7Rah1sqeyh6
v3v6Hji4sOXwn+jdhAHJTM2Iq1RjwccObEdYc67VM9+2ZwEipJI5HYl1qbYKTA0em0kwK
HNa+naIkWsTSwNj81HaAkaL21LMhcHnRytBfTeyySLgNHxy6VFX1ipMN3pdtQbJn0cfRI
evyxroMs7S+nMvhtr1lhrRzNaC3iW9OIskS9zNjKUd2Becj5+Bt1JHmlqu3Is9H67kySd
HeF1XTMVWHDo30n9msA1vB7Bqo1w26qzV3S97vbhrApQtYJAn0bIilVIAEasIYIbqrkTQ
/kmDO4uMpCDgZKta7bP+P3CjBrGmK1w98 remote_ssh_user@device
```

1 Use the ssh-keygen utility of the OpenSSH software suite to generate a private and public key pair as show below:

#ssh-keygen -t rsa -C remote_ssh_user@device

2 Save the private key on the client device.

3 Copy the the public key to the switch in the preferred directory. Including the user id as part of the file-name can help identify the different keys:

#scp ~/.ssh/new_ssh_user_rsa.pub admin@192.168.2.1:/flash/system

4 Verify that the *userid* that will use SSH is a valid user name on the OmniSwitch. If the username does not already exist on the switch create the user name with the appropriate privileges.

5 Install the public key on the OmniSwitch for the specified user.

-> **installsshkey new_ssh_user /flash/system/new_ssh_user_rsa.pub**

6 Connect to the OmniSwitch using SSH with PKA.

#ssh -o PreferredAuthentications=publickey new_ssh_user@192.168.2.1 –v

---

Note. By default if PKA fails, the user is prompted for a password. This is the password that was specified when the user name was created on the OmniSwitch.

---

7 (Optional) To enforce Secure Shell PKA on a switch and not prompt for a password use the **ssh enforce-pubkey-auth** command.

Revoking a Key

The following procedure can be used to revoke a key:

->revokesshkey new_ssh_user remote_ssh_user@192.168.10.1

Starting a Secure Shell Session from the OmniSwitch

To start a Secure Shell session, issue the **ssh** command and identify the IP address or hostname for the device you are connecting to.

The following command establishes a Secure Shell interface from the local OmniSwitch to a remote device:

```
-> ssh 198.51.100.50
login as:
```

You must have a login and password that is recognized by the IP address you specify. When you enter your login, the device you are logging in to, will request your password as shown here:

```
-> ssh 198.51.100.50
Password:
Welcome to the Alcatel-Lucent OS10K 7.1.1.1638.R01 Development, August 26, 2010.
Copyright (c) 1994-2010 Alcatel-Lucent.  All Rights Reserved.
OmniSwitch(tm) is a trademark of Alcatel-Lucent, registered in the United States
Patent and Trademark Office.
```

Once the Secure Shell session is established, you can use the remote device specified by the IP address on a secure connection from your OmniSwitch.

Note. The login parameters for Secure Shell session login parameters can be affected by the **session login-attempt** and **session login-timeout** CLI commands.

Modifying the Login Banner

The Login Banner feature allows you to change the banner that displays whenever someone logs into the switch. This feature can be used to display messages about user authorization and security. You can display the same banner for all login sessions or you can implement different banners for different login sessions. You can display a different banner for logins initiated by FTP sessions than for logins initiated by a direct console or a Telnet connection. The default login message looks similar to the following:

```
login : user123
password :

Welcome to the Alcatel-Lucent OS10K
Software Version 7.1.1.733.R01 Development, August 05, 2010.

Copyright(c), 1994-2010 Alcatel-Lucent. All Rights reserved.

OmniSwitch(TM) is a trademark of Alcatel-Lucent registered
in the United States Patent and Trademark Office.
```

Here is an example of a banner that has been changed:

```
login : user123
password :

Welcome to the Alcatel-Lucent OS10K
Software Version 7.1.1.733.R01 Development, August 05, 2009.

Copyright(c), 1994-2010 Alcatel-Lucent. All Rights reserved.

OmniSwitch(TM) is a trademark of Alcatel-Lucent registered
in the United States Patent and Trademark Office.

********** LOGIN ALERT ************************
This switch is a secure device. Unauthorized
use of this switch will go on your permanent record.
```

Two steps are required to change the login banner. These steps are listed here:

- Create a text file that contains the banner you want to display in the switch's /**flash**/**switch** directory.
- Enable the text file by entering the **session banner** CLI command followed by the filename.

To create the text file containing the banner text, you may use the **vi** text editor in the switch or you create the text file using a text editing software package and transfer the file to the switch's /**flash/switch** directory.

If you want the login banner in the text file to apply to FTP switch sessions, execute the following CLI command where the text filename is **firstbanner.txt**.

```
-> session ftp banner/flash/switch/firstbanner.txt
```

If you want the login banner in the text file to apply to CLI switch sessions, execute the following CLI command where the text filename is **secondbanner.txt**.

```
-> session cli banner /flash/switch/secondbanner.txt
```

If you want the login banner in the text file to apply to HTTP switch sessions, execute the following CLI command where the text filename is **thirdbanner.txt**.

```
-> session http banner/flash/switch/thirdbanner.txt
```

The banner files must contain only ASCII characters and should bear the **.txt** extension. The switch will not reproduce graphics or formatting contained in the file.

Modifying the Text Display Before Login

By default, the switch does not display any text before the login prompt for any CLI session.

At initial bootup, the switch creates a **pre_banner.txt** file in the /**flash/switch** directory. The file is empty and may be edited to include text that you want to display before the login prompt.

For example:

```
Please supply your user name and password at the prompts.

login : user123
password :
```

In this example, the pre_banner.txt file has been modified with a text editor to include the **Please supply your user name and password at the prompts** message.

The pre-banner text cannot be configured for FTP sessions.

To remove a text display before the login prompt, delete the pre_banner.txt file (it will be recreated at the next bootup and will be empty), or modify the pre_banner.txt file.

Configuring Login Parameters

You can set the number of times a user may attempt unsuccessfully to log in to the switch's CLI by using the **session login-attempt** command as follows:

```
-> session login-attempt 5
```

In this example, the user may attempt to log in to the CLI five (5) times unsuccessfully. If the user attempts to log in the sixth time, the switch will break the TCP connection.

You may also set the length of time allowed for a successful login by using the **session login-timeout** command as follows:

```
-> session login-timeout 20
```

In this example, the user must complete the login process within 20 seconds. This means that the time between a user entering a login name and the switch processing a valid password must not exceed 20 seconds. If the time-out period exceeds, the switch will break the TCP connection.

Configuring the Inactivity Timer

You can set the amount of time that a user must be inactive before the session times out. To change the setting, enter the **session timeout** command with the type of session and the desired number of minutes.

For example:

```
-> session cli timeout 8

-> session ftp timeout 5

-> session http timeout 10
```

Enabling the DNS Resolver

A Domain Name System (DNS) resolver is an optional internet service that translates host names into IP addresses. Every time you enter a host name when logging into the switch, a DNS service must look up the name on a server and resolve the name to an IP address. You can configure up to three IPv4 domain name servers and three IPv6 domain name servers that will be queried in turn to resolve the host name. If all servers are queried and none can resolve the host name to an IP address, the DNS fails. If the DNS fails, you must either enter an IP or IPv6 address in place of the host name or specify the necessary lookup tables on one of the specified servers.

Note. You do not need to enable the DNS resolver service unless you want to communicate with the switch by using a host name. If you use an IP or IPv6 address rather than a host name, the DNS resolver service is not needed.

You must perform three steps on the switch to enable the DNS resolver service.

1 Set the default domain name for DNS lookups with the **ip domain-name** CLI command.

```
-> ip domain-name mycompany1.com
```

2 Use the **ip domain-lookup** CLI command to enable the DNS resolver service.

```
-> ip domain-lookup
```

You can disable the DNS resolver by using the **no ip domain-lookup** command. For more information, refer to the *OmniSwitch CLI Reference Guide*.

3 Specify the IP addresses of the servers with the **ip name-server** CLI command. These servers will be queried when a host lookup is requested.

```
-> ip name-server 189.202.191.14 189.202.191.15 189.255.19.1
```

Verifying Login Settings

To display information about login sessions, use the following CLI commands:

who	Displays all active login sessions (e.g., console, Telnet, FTP, HTTP, Secure Shell, Secure Shell FTP).
whoami	Displays the current user session.
show session config	Displays session configuration information (e.g., default prompt, banner file name, inactivity timer, login timer, login attempts).
show dns	Displays the current DNS resolver configuration and status.

For more information about these commands, refer to the *OmniSwitch CLI Reference Guide*.

2 Managing System Files

This chapter describes the several methods of transferring software files onto the OmniSwitch and how to register those files for use by the switch. This chapter also describes several basic switch management procedures and discusses the Command Line Interface (CLI) commands used.

- File Management (copy, secure copy, edit, rename, remove, change, and display file attributes)
- Directory Management (create, copy, move, remove, rename, and display directory information)
- System Date and Time (set system clock)

CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

In This Chapter

Configuration procedures described in this chapter include:

For related information about connecting a terminal to the switch, see your *Getting Started Guide*. For information about switch command privileges, see Chapter 7, "Managing Switch Security."

File Management Specifications

The functionality described in this chapter is supported on the OmniSwitch Series switches unless otherwise stated in the following Specifications table or specifically noted within any section of this chapter.

Platforms Supported	OmniSwitch 10K, 6900
Software Licensing	OmniSwitch 10K, 6900.
File Transfer Methods	FTP (v4/v6), SFTP (v4/v6), SCP (v4/v6), TFTP
Client/Server Support	FTP - Client (IPv4 Only) or Server SFTP - Client or Server SCP - Client or Server TFTP - Client
Number of concurrent FTP/ SFTP sessions	4
Configuration Recovery	The **flash**/**certified** directory holds configurations that are certified as the default start-up files for the switch. They will be used in the event of a non-specified reload.
Default Switch Directory - /**flash**	Contains the **certified**, **working, switch, network,** and user-defined directories.
File/Directory Name Metrics	255 character maximum. File and directory names are case sensitive.
File/Directory Name Characters	Any valid ASCII character except '/'.
Sub-Directories	Additional user-defined directories created in the /**flash** directory.
Text Editing	Standard Vi standard editor.
System Clock	Set local date, time and time zone, Universal Time Coordinate (UTC), Daylight Savings (DST or summertime).

Switch Administration Overview

The OmniSwitch has a variety of software features designed for different networking environments and applications. Over the life of the switch, it is very likely that your configuration and feature set will change because the needs of your network are likely to expand. Also, software updates become available from Alcatel-Lucent. If you change your configuration to upgrade your network, you must understand how to install switch files and to manage switch directories.

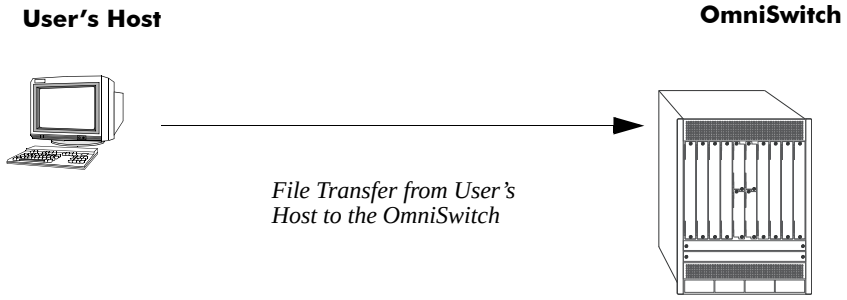
The OmniSwitch Series uses flash memory store files, including executable files (used to operate switch features and applications), configuration files, and log files.

You need to understand the various methods of loading files onto the switch for software upgrades and new features. Once the files are on the switch, the CLI has commands that allow you to load, copy, and delete these files. The CLI also has commands for displaying, creating, and editing ASCII files directly on the switch. You may also want to establish a file directory structure to help organize your files on the switch.

All the files and directories on the switch bear a time stamp. This is useful for switch administration because the time stamp allows you to tell at a glance which files are the most recent. You can set the system clock that controls these time stamps as well as other time based switch functions.

File Transfer

The switch can receive and send files by using industry standard local and remote transfer methods. Each of these methods is defined and explained. Because file transfers can involve logging onto the switch from a remote host, security factors, such as DNS resolver and Authenticated Switch Access requirements should be considered.

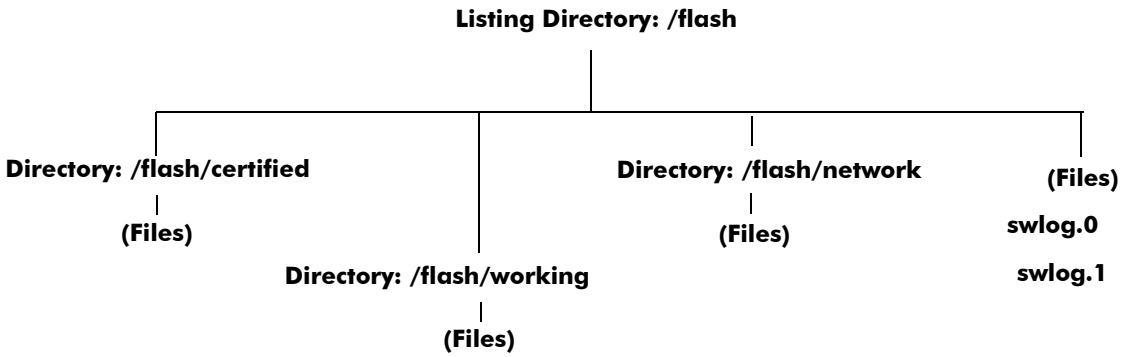


File Transfer to OmniSwitch

The OmniSwitch has a directory structure that allows you to install new software while maintaining a backup copy of your old configuration.

Switch Directories

You can create your own directories in the switch ***flash*** directory. This allows you to organize your configuration and text files on the switch. You can also use the **vi** command to create files. This chapter tells you how to make, copy, move, and delete both files and directories.



Switch Flash Directory

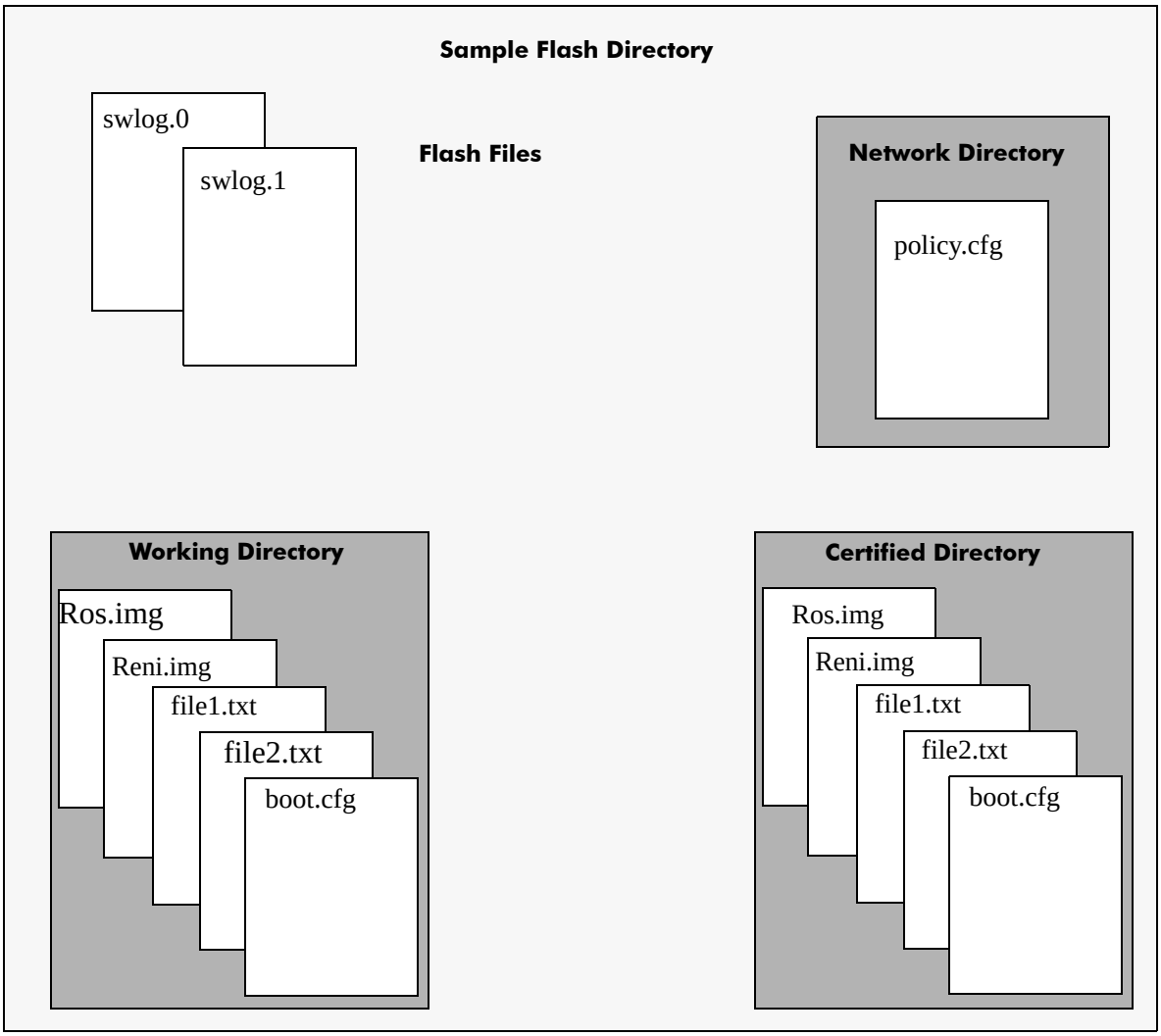
File and Directory Management

A number of CLI commands allow you to manage files on your switch by grouping them into sub-directories within the switch's flash directory. For documentation purposes, we have categorized the commands into the following three groups.

- **Directory** commands allow you to create, copy, move, remove, rename, and display directories.
- **File** commands allow you copy, secure copy, edit, rename, remove, change, and display file attributes.
- **Utility** commands display memory and system diagnostic information.

The following illustration represents a *sample* flash directory. The sample directories hold various files. This sample flash directory is used in the explanations of the directory, file and utility CLI commands described in the following section.

The switch may show files and directories different from the ones shown in this example.



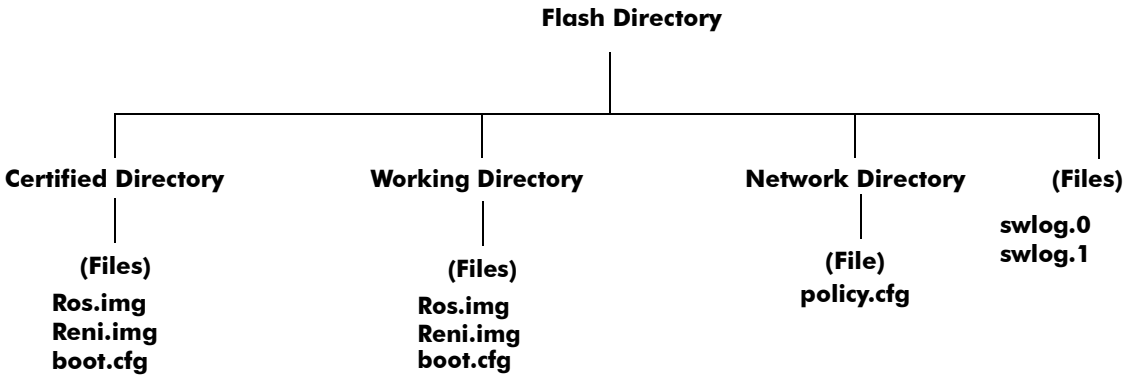
To list all the files and directories in your current directory, use the **ls** command. Here is a sample display of the flash directory.

```
-> ls -l
-rw-r--r--    1 root     0              342 Aug 30 18:28 boot.cfg.1.err
drwxrwxrwx    2 root     0             1024 Aug 30 18:28 certified
drwx------    2 root     0          1638400 Aug 30 18:28 lost+found
d---------    2 root     0             1024 Aug 30 18:28 network
drwxr-xr-x    3 root     0             1024 Aug 30 18:28 switch
-rw-r--r--    1 root     0            51569 Aug 30 22:52 swlog
drwxr-xr-x    2 root     0             1024 Aug 30 18:28 system
drwxrwxrwx    2 root     0             1024 Aug 30 18:28 dir1
```

Directory Commands

The directory commands are applied to the switch file system and to files contained within the file system. When you first enter the flash directory, your login is located at the top of the directory tree. You may navigate within this directory by using the **pwd** and **cd** commands (discussed below). The location of your login within the directory structure is called your *current directory*. You need to observe your login location because when you issue a command, that command applies only to directories and files in your current directory unless another path is specified.

The following drawing is a logical representation of the OmniSwitch file directory shown in the illustration on page 2-5.



Sample Switch Directory Tree

Determining Your Location in the File Structure

Use the **pwd** command to display the path to your current directory. When you first log into the switch, your current directory is the ***flash*** directory. If you enter the **pwd** command, the following will be displayed:

```
-> pwd
/flash

->
```

The display shows the name of the current directory and its path. If your current directory is the ***certified*** directory and you enter the **pwd** command, the following will be displayed:

```
-> pwd
/flash/certified

->
```

The display shows the path to your current directory.

Changing Directories

Use the **cd** command to navigate within the file directory structure. The **cd** command allows you to move "up" or "down" the directory tree. To go down, you must specify a directory located in your current directory. For example:

```
->pwd
/flash
->cd certified
->pwd
/flash/certified
```

To move "up" the directory tree, use the **cd** command. Enter **cd ..** without specifying a directory name and your current directory will move up one directory level. If you enter **cd** without the dots, your current directory will move to the top of the tree. The following example shows the **cd** command used where the current directory is /**flash**/**certified**.

```
->pwd
/flash/certified

-> cd
->
```

To verify that your current directory has moved up the directory tree, use the **pwd** command to display your location. The display shows you have moved up one level from the /**flash**/**certified** directory and that your current directory is /**flash**.

```
-> pwd
/flash
```

If you use the **cd** command while you are at the top of the directory tree, the **cd** command will have no effect on the location of your login. In other words, if you use **cd** while your current directory is /**flash**, your current directory will remain /**flash** after you execute the **cd** command.

Making a New Directory

To make a new directory use the **mkdir** command. You may specify a path for the new directory. Otherwise, the new directory will be created in your current directory. The syntax for this command requires a slash (/) and no space between the path and the new directory name. Also, a slash (/) is required at the beginning of your path specification.

The following command makes a new directory in the ***dir1*** directory on an OmniSwitch:

```
-> mkdir /flash/dir1/newdir1
```

Copying an Existing Directory

The **cp** command copies directories, as well as any associated subdirectories and files. Before using this command, you should make sure you have enough memory space in your target directory to hold the new material you are copying.

In this example, a copy of the ***dir1*** directory and all its contents will be created in the /***flash*** directory.

```
->cp -r /flash/dir1 /flash/dir2
```

Removing a Directory and its Contents

The **rmdir** command removes the specified directory and all its contents. The following command would remove the ***dir1*** directory.

```
->rmdir /flash/dir1

or

->rm -rf /flash/dir1
```

File Commands

The file commands apply to files located in the /**flash** file directory and its sub-directories.

Creating or Modifying Files

The switch has an editor for creating or modifying files. The editor is invoked by entering the **vi** command and the name of the new file or existing file that you want to modify. For example:

```
-> vi /flash/my_file
```

This command puts the switch in editor mode for **my_file**. If my_file does not already exist, the switch will create the file in the flash directory. In the editing mode, the switch uses command keystrokes similar to any vi UNIX text editor. For example, to quit the edit session and save changes to the file, type **ZZ**.

Copy an Existing File

Use the **cp** command to copy an existing file. You can specify the path and filename for the original file being copied as well as the path and filename for the new copy being created. If no path is specified, the command assumes the current directory.

For example:

```
->cp /flash/dir1/sourcefile.img /flash/certified

->cp sourcefile.img /flash/certified

->cp /flash/dir1/sourcefile.img newfile.img
```

Secure Copy an Existing File

Use the **scp** command to copy an existing file in a secure manner. You can specify the path and filename for the original file being copied as well as the path and filename for a new copy being created. If no path is specified, the command assumes the current directory. The following syntax copies all of the image files in the **working** directory from a remote switch to the local **working** directory:

```
-> scp admin@198.51.100.1:/flash/working/*.img /flash/working
admin's password for keyboard-interactive method:
```

This second example helps copy all the image files from the user's current **working** directory to the remote switch's **working** directory. A copy of all the image files will appear in the /**flash**/**working** directory of the remote switch, once the following command is executed.

```
-> scp /flash/working/*.img admin@198.51.100.1:/flash/working
admin's password for keyboard-interactive method:
```

Move an Existing File or Directory

The **mv** command is used to move an existing file or directory to another location. You can specify the path and name for the file or directory being moved. If no path is specified, the command assumes the current path. You can also specify a path and a new name for the file or directory being moved. If no name is specified, the existing name will be used.

```
-> mv /flash/testfiles/testfile2 /flash/working/testfile2
-> mv testfile2 /flash/working/newtestfile2
```

Change File Attribute and Permissions

The **chmod** command can be used to change read-write privileges for the specified file. The following syntax sets the privilege for the **config1.txt** file to read-write. In this example, the user's current directory is the /**flash** file directory. For example:

To set the permission for the **config1.txt** file to read-only, use the following syntax.

```
-> chmod -w /flash/config1.txt
```

To set the permission for the **config1.txt** file to read/write, use the following syntax.

```
-> chmod +w /flash/config1.txt
```

Delete an Existing File

The delete command deletes an existing file. If you use the **rm** command from the directory containing the file, you do not need to specify a path. If you are in another directory, you must specify the path and name for the file being deleted. For example:

```
-> rm /flash/config.txt
```

Managing Files on Redundant CMMs

On an OmniSwitch you can copy a file from a secondary management module to a primary management module or from a primary management module to a secondary management module with the **rcp** command. To use this command enter **rcp** followed the secondary management module of the switch, the path and file name of the source file on the secondary management module of the switch, and the destination file name on the primary management module of the switch.

For example, to copy the **boot.cfg** file to the **/flash** directory on primary management module in a switch and name it **boot.cfg.bak** enter:

```
-> rcp cmm-b: /flash/boot.cfg boot.cfg.bak
```

To delete a file on a secondary management module of the non-primary switch, use the **rrm** command. To use this command, enter **rrm** followed by the path and file name of the file on the secondary management module of the non-primary switch to be deleted.

For example, to delete the **boot.cfg** file in the **/flash** directory on a secondary management module of the non-primary switch, enter:

```
-> rrm 2/flash/boot.cfg
```

To list the directory contents of a secondary management module of the non-primary switch, use the **rls** command by entering **rls**, followed by the path name of the directory you want to display. (As an option, you can also specify a specific file name to be displayed.)

For example, to display the contents of the **/flash** directory on a secondary management module non-primary switch, enter:

```
-> rls 2/flash
```

A screen similar to the following will be displayed:

```
drw         1024  Sep 13 16:46   certified/
drw         1024  Sep 13 16:45   working/
-rw        64000  Sep 13 16:46   swlog.0
-rw        64000  Sep  8 21:24   swlog.1
drw         1024  Sep 13 16:45   switch/
drw         1024  Sep 10 17:34   network/
-rw          256  Sep 13 16:41   random-seed
drw         1024  Jun 22  1986   tk.dir/
```

Utility Commands

The utility commands include **freespace**, **fsck**, and **newfs**. These commands are used to check and verify flash.

Displaying Free Memory Space

The **freespace** command displays the amount of free memory space available for use in the switch's file system. You may issue this command from any location in the switch's directory tree.

```
-> freespace
/flash 16480256 bytes free
```

Performing a File System Check

The **fsck** command performs a file system check and can repair any errors found. It displays diagnostic information in the event of file corruption.

There are two options available with the **fsck** command: **no-repair** and **repair**. Specifying the **no-repair** option performs only the file system check whereas specifying the **repair** option performs the file system check and also repairs any errors found on the file system.

If you want to repair any errors found automatically while performing the file system check, you must specify the flash directory as follows:

```
-> fsck /uflash repair
```

The screen displays the following output:

```
/uflash/  - disk check in progress ...
/uflash/  - Volume is OK
Change volume Id from 0x0 to 0xef2e3c

          total # of clusters:  29,758
           # of free clusters:  18,886
            # of bad clusters:  0
              total free space:  77,357,056
    max contiguous free space:  55,451,648 bytes
                  # of files:  59
                # of folders:  5
         total bytes in files:  44,357,695
             # of lost chains:  0
   total bytes in lost chains:  0
```

While performing the repair operation, the switch will display the errors found and specify those errors that have been repaired. If there are no errors found, then just the file system information is displayed.

Deleting the Entire File System

The **newfs** command deletes the file system and all the files and directories contained in it. This command is used when you want to reload all files in the file system.

Caution. This command will delete all of the switch's system files. All configurations programmed into the switch will be lost. Do not use this command unless you are prepared to reload *all* files.

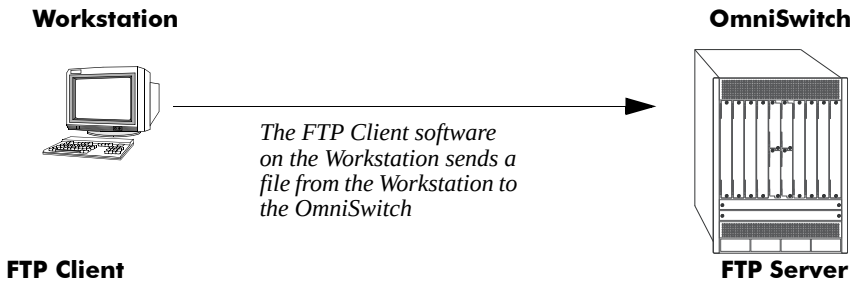
Loading Software onto the Switch

There are multiple methods for loading software to and from your switch. The method you use depends on your workstation software, your hardware configuration, and the location and condition of your switch. These methods are discussed here.

- **FTP/SFTP/SCP Server**—You can use the switch as a FTP/SFTP server. If you have client software on your workstation, you can transfer a file to the switch. This is normally done to load or upgrade the switch's software or configurations.
- **TFTP Client**—You can use the TFTP client functionality on an OmniSwitch to transfer software to/ from a TFTP server.
- **FTP/SFTP/SCP Client**—You can use the switch as an FTP/SFTP client by connecting a terminal to the switch's console port and using standard FTP commands. This feature is useful in cases where you do not have access to a workstation with an FTP client. .

Using the Switch as a Server

The switch can act as a server for receiving files transferred from your workstation. You can transfer software files to the switch by using standard client software located on a host workstation. This is normally done to load or upgrade the switch software.



OmniSwitch as a Server

The following describes how to transfer files where the switch is acting as an FTP server.

1 Log into the switch. Use your workstation's FTP client software just as you would with any FTP application. To log in to the switch, start your FTP client. Where the FTP client asks for "Name", enter the IP address of your switch. Where the FTP client asks for "User ID", enter the username of your login account on the switch. Where the FTP client asks for "Password", enter your switch password.

2 Specify the transfer mode. If you are transferring a switch image file, you must specify the binary transfer mode on your FTP client. If you are transferring a configuration file, you must specify the ASCII transfer mode.

3 Transfer the file. Use the FTP "put" command or click the client's download button to send the file to the switch.

Using the Switch as an FTP Client

Using the switch as an FTP client is useful in cases where you do not have access to a workstation with an FTP client. You can establish an FTP session locally by connecting a terminal to the switch console port. You can also establish an FTP session to a remote switch by using a Telnet session. Once you are logged into the switch as an FTP client, you can use standard FTP commands.

Use the switch **ftp** command to start its FTP client.

1 Establish a connection to the switch as explained in your appropriate *Getting Started Guide.*

2 Log on to the switch and enter the **ftp** command to start the FTP client. Next, enter a valid host name or IP address.

```
-> ftp 198.51.100.101
Connecting to [198.51.100.101]...connected
220 cosmo FTP server (UNIX(r) System V Release 4.1) ready
Name :
```

Note. You can only use a host name instead of an IP address if the DNS resolver has been configured and enabled. If not, you must specify an IP address.

---

3 Set the client to binary mode with the **bin** command. Enter a valid user name and password for the host you specified with the **ftp** command. A screen similar to the following is displayed:

```
Name: Jsmith
331 Password required for Jsmith
Password: *****
230 User Jsmith logged in.
```

4 After logging in, you will receive the **ftp->** prompt. You may enter a question mark (?) to view available FTP commands as shown below.

```
ftp->?

Supported commands:
 ascii       binary      bye         cd          delete
 dir         get         help        hash        ls
 put         pwd         quit        remotehelp  user
 lpwd        mput        mget        prompt      !ls
 lcd         user
```

Using Secure Shell FTP

1 Log on to the OmniSwitch and issue the **sftp** CLI command. The command syntax requires you to identify the IP address for the device you are connecting to. The following command establishes a Secure Shell FTP interface from the local OmniSwitch to IP address 198.51.100.125.

```
-> sftp 198.51.100.125
login as:
```

2 You must have a login and password that is recognized by the IP address you specify. When you enter your login, the device you are logging in to, will request your password as shown here.

```
-> sftp 198.51.100.125
login as: rrlogin2
rrlogin2's password for keyboard-interactive method:
```

3 After logging in, you will receive the **sftp>** prompt. You may enter a question mark (?) to view available Secure Shell FTP commands and their definitions

Closing a Secure Shell FTP Session

To terminate the Secure Shell FTP session, issue the **exit** command. The following will display:

```
-> exit
Connection to 10.222.30.125 closed.
```

Using TFTP to Transfer Files

Trivial File Transfer Protocol (TFTP), a client-server protocol, can be used to transfer files between the TFTP server and client. TFTP client functionality on the OmniSwitch is used to download files from or upload files to the TFTP server within a LAN using the **tftp** command.

The following is an example of how to start a TFTP session to download a file from a TFTP server:

```
-> tftp -g -l local_file -r remote_file 198.51.100.50
```

When you enter the above command the following actions are performed:

- Establishes a TFTP session with the TFTP server 198.51.100.50.
- Downloads the ‘remote_file’ file and saves it to file named ‘local_file’.

You can specify a path for the specified file and if the file name is specified without a path then the current path (**/flash**) is used by default. If a local filename is not specified, then the remote filename is used by default. A TFTP server does not prompt for a user to login and only one active TFTP session is allowed at any point of time.

Note. When downloading a file to the switch, the file size must not exceed the available flash space.

Installing Software Licenses

Some features require a software license and are restricted only to a licensed user. Purchasing a license part number along with an authorization code from Alcatel-Lucent is required. The authorization code is then used to generate a license file.

To generate a license file, install the file on the switch, and active features, do the following:

1 Log on to https://service.esd.alcatel-lucent.com/portal/page/portal/EService/LicenseGeneration and provide the serial number and MAC address of the switch along with the authorization code. Use the serial number and CMM MAC address from the **show chassis** command.

A license file, for example *swlicense.txt*, is generated. A license file can have any name.

2 Save the *swlicense.txt* file in the **/flash** directory of the primary CMM.

3 To install the license onto the switch, use the **license** command with the file name and reboot the switch. For example:

```
-> license apply file /flash/swlicense.txt
```

4 To verify the installation after reboot, use the **show license-info** command.

Note. For multiple entries of serial numbers, MAC addresses, and authorization codes, use a CSV formatted file and upload the file on to the website. A single license file is generated for all the switches.

Once the license is applied it is written to the EEPROM and the license file is no longer needed.

Licensed Feature Matrix

OmniSwitch 6900			OmniSwitch 10K		
Advanced		**DataCenter**	**Advanced**	**DataCenter**	**U16L**
OSPF v2/v3	VRRP	DCB	SPB	DCB	OS10K-XNI-U16
BGP	VRRP v3	EVB		EVB	
MP-BGP	ECMP for OSPF	ETS		ETS	
Policy Based Routing	RIPng				
DVMRP	PIM-SM				
PIM-SM IPv6	IPSec				
VRF	SPB				

Setting the System Clock

The switch clock displays time by using a 24-hour clock format. It can also be set for use in any time zone. Daylight Savings Time (DST) is supported for a number of standard time zones. DST parameters can be programmed to support non-standard time zones and time off-set applications.

All switch files and directories listed in the flash directory bear a time stamp. This feature is useful for file management purposes.

Setting Date and Time

You can set the local date, time zone, and time for your switch or you can also set the switch to run on Universal Time Coordinate (UTC or GMT).

Date

To display the current system date for your switch, use the **system date** command. If you do not specify a new date in the command line, the switch will display the current system date.

To modify the switch's current system date, enter the new date with the command syntax. The following command will set the switch's system date to June 23, 2002.

```
-> system date 06/23/2002
```

When you specify the date you must use the *mm/dd/yyyy* syntax where *mm* is the month, *dd* is the day and *yyyy* is the year.

Time Zone

To determine the current time zone or to specify a new time zone for your switch, use the **system timezone** command. This specifies the time zone for the switch and sets the system clock to run on UTC time (or Greenwich Mean Time). The following is displayed for the Pacific standard time zone:

```
-> system timezone
PST: (Coordinated Universal Time) UTC-8 hours
```

To set a new time zone for the system clock, use the **system timezone** command along with the appropriate time zone abbreviation. Refer to the table in "Daylight Savings Time Configuration" on page 2-19 for time zone abbreviations. The following command sets the system clock to run on Pacific Standard Time:

```
-> system timezone pst
```

Time

To display the current local time for your switch, use the **system time** command. If you do not specify a new time in the command line, the current system time is displayed as shown:

```
-> system time
17:08:51
```

To modify the switch's current system time, enter the **system time** command. When you specify the time you must use the *hh:mm:ss* syntax where *hh* is the hour based on a 24 hour clock. The *mm* syntax represents minutes and *ss* represents seconds. You must use two digits to specify the minutes and two digits to specify the seconds. The following command will set the switch's system time to 10:45:00 a.m:

```
-> system time 10:45:00
```

The following command will set the switch's system time to 3:14:00 p.m:

```
-> system time 15:41:00
```

Daylight Savings Time Configuration

The switch automatically adjusts for Daylight Savings Time (DST) depending on the timezone selected. If the configured timezone supports DST it is automatically enabled and cannot be disabled. If the configured timezone does not support DST it is automatically disabled and cannot be enabled. Refer to the table on page 2-19 to determine daylight savings time settings.

The following table shows a list of supported time zone abbreviations and DST parameters.

Time Zone and DST Information Table

Abbreviation	Name	Hours from UTC	DST Start	DST End	DST Change
nzst	New Zealand	+12:00	1st Sunday in Oct. at 2:00 a.m.	3rd Sunday in Mar. at 3:00 a.m.	1:00
zp11	No standard name	+11:00	No default	No default	No default
aest	Australia East	+10:00	Last Sunday in Oct. at 2:00 a.m.	Last Sunday in Mar. at 3:00 a.m.	1:00
gst	Guam	+10:00	No default	No default	No default
acst	Australia Central Time	+09:30	Last Sunday in Oct. at 2:00 a.m.	Last Sunday in Mar. at 3:00 a.m.	1:00
jst	Japan	+09:00	No default	No default	No default
kst	Korea	+09:00	No default	No default	No default
awst	Australia West	+08:00	No default	No default	No default
zp8	China; Manila, Philippines	+08:00	No default	No default	No default
zp7	Bangkok	+07:00	No default	No default	No default
zp6	No standard name	+06:00	No default	No default	No default
zp5	No standard name	+05:00	No default	No default	No default
zp4	No standard name	+04:00	No default	No default	No default
msk	Moscow	+03:00	Last Sunday in Mar. at 2:00 a.m.	Last Sunday in Oct. at 3:00 a.m.	1:00
eet	Eastern Europe	+02:00	Last Sunday in Mar. at 2:00 a.m.	Last Sunday in Oct. at 3:00 a.m.	1:00
cet	Central Europe	+01:00	Last Sunday in Mar. at 2:00 a.m.	Last Sunday in Oct. at 3:00 a.m.	1:00
met	Middle Europe	+01:00	Last Sunday in Mar. at 2:00 a.m.	Last Sunday in Oct. at 3:00 a.m.	1:00
bst	British Standard Time	+00:00	Last Sunday in Mar. at 1:00 a.m.	Last Sunday in Oct. at 3:00 a.m.	1:00
wet	Western Europe	+00:00	Last Sunday in Mar. at 1:00 a.m.	Last Sunday in Oct. at 3:00 a.m.	1:00
gmt	Greenwich Mean Time	+00:00	No default	No default	No default
wat	West Africa	-01:00	No default	No default	No default
zm2	No standard name	-02:00	No default	No default	No default
zm3	No standard name	-03:00	No default	No default	No default

Time Zone and DST Information Table (continued)

Abbreviation	Name	Hours from UTC	DST Start	DST End	DST Change
nst	Newfoundland	-03:30	1st Sunday in Apr. at 2:00 a.m.	Last Sunday in Oct. at 2:00 a.m.	1:00
ast	Atlantic Standard Time	-04:00	2nd Sunday in Mar. at 2:00 a.m.	1st Sunday in Nov. at 2:00 a.m.	1:00
est	Eastern Standard Time	-05:00	2nd Sunday in Mar. at 2:00 a.m.	1st Sunday in Nov. at 2:00 a.m.	1:00
cst	Central Standard Time	-06:00	2nd Sunday in Mar. at 2:00 a.m.	1st Sunday in Nov. at 2:00 a.m.	1:00
mst	Mountain Standard Time	-07:00	2nd Sunday in Mar. at 2:00 a.m.	1st Sunday in Nov. at 2:00 a.m.	1:00
pst	Pacific Standard Time	-08:00	2nd Sunday in Mar. at 2:00 a.m.	1st Sunday in Nov. at 2:00 a.m.	1:00
astcam	Atlantic Standard Time Central America	-04:00	1st Sunday in Apr. at 2:00 a.m.	Last Sunday in Oct. at 2:00 a.m.	1:00
estcam	Eastern Standard Time Central America	-05:00	1st Sunday in Apr. at 2:00 a.m.	Last Sunday in Oct. at 2:00 a.m.	1:00
cstcam	Central Standard Time Central America	-06:00	1st Sunday in Apr. at 2:00 a.m.	Last Sunday in Oct. at 2:00 a.m.	1:00
mstcam	Mountain Standard Time Central America	-07:00	1st Sunday in Apr. at 2:00 a.m.	Last Sunday in Oct. at 2:00 a.m.	1:00
pstcam	Pacific Standard Time Central America	-08:00	1st Sunday in Apr. at 2:00 a.m.	Last Sunday in Oct. at 2:00 a.m.	1:00
akst	Alaska	-09:00	1st Sunday in Apr. at 2:00 a.m.	Last Sunday in Oct. at 2:00 a.m.	1:00
hst	Hawaii	-10:00	No default	No default	No default
zm11	No standard name	-11:00	No default	No default	No default

3 Managing CMM Directory Content

The CMM (Chassis Management Module) software runs the OmniSwitch Series switches. Each OmniSwitch chassis can run with two CMMs to provide redundancy; one CMM is designated as the primary CMM, and the other is designated as the secondary CMM. The directory structure of the CMM software is designed to prevent corrupting or losing switch files. It also allows you to retrieve a previous version of the switch software.

In This Chapter

This chapter describes the basic functions of CMM software directory management and how to implement them by using the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

This chapter contains the following information:

- The interaction between the running configuration, the working directory, and the certified directory is described in "CMM Files" on page 3-3.
- A description of how to restore older versions of files and prevent switch downtime is described in "Software Rollback Feature" on page 3-4.
- The CLI commands available for use and the correct way to implement them are listed in "Managing Switch Configurations - Single CMM" on page 3-11.
- Managing, upgrading and restoring files using a USB flash drive described in "Using the USB Flash Drive" on page 3-21.
- Upgrading switch code using ISSU described in "In-Service Software Upgrade" on page 3-23.

CMM Specifications

Platforms Supported	OmniSwitch 10K, 6900
Size of Flash Memory	2 GB
Maximum Length of File Names	255 Characters
Maximum Length of Directory Names	255 Characters
Default Boot Directory	Certified

USB Flash Drive Specifications

Platforms Supported	OmniSwitch 10K, 6900
USB Flash Drive Support	Alcatel-Lucent Certified USB Flash Drive
Automatic Software Upgrade	Supported
Disaster Recovery	Supported OS10K - **Rrescue.img** file required OS6900 - **Trescue.img** file required

Note. The format of the Alcatel-Lucent certfied USB Flash Drive must be FAT32. To avoid file corruption issues the USB Drive should be stopped before removing from a PC. Directory names are case sensitive and must be lower case.

Note. Many of the examples below use the ***working*** directory as the RUNNING DIRECTORY. However, any user-defined directory can be configured as the RUNNING DIRECTORY.

CMM Files

The management of a switch is controlled by the following types of files:

- Image files, which are proprietary code developed by Alcatel-Lucent. These files are not configurable by the user, but may be upgraded from one release to the next. These files are also known as archive files as they are really the repository of several smaller files grouped together under a common heading.
- A configuration file, named **boot.cfg**, which is an ASCII-based text file, sets and controls the configurable functions inherent in the image files provided with the switch. This file can be modified by the user. When the switch boots, it looks for the file called **boot.cfg**. It uses this file to set various switch parameters defined by the image files.

Modifications to the switch parameters affect or change the configuration file. The image files are static for the purposes of running the switch (though they can be updated and revised with future releases or enhancements). Image and configuration files are stored in the Flash memory (which is equivalent to a hard drive memory) in specified directories. When the switch is running, it loads the image and configuration files from the Flash memory into the RAM. When changes are made to the configuration file, the changes are first stored in the RAM. The procedures for saving these changes via the CLI are detailed in the sections to follow.

Available Files

This table lists the image and configuration files for the OmniSwitch. Most of the files listed here are part of the base switch configuration. Files that support an optional switch feature are noted in the table.

OmniSwitch 10K

File Name	Base or Optional Software	Description
Reni.img	Base Software	NI image for all Ethernet-type NIs
Ros.img	Base Software	CMM Operating System
Rrescue.img	Optional Software	Disaster Recovery file (used on USB flash drive)
boot.cfg		Switch Configuration File

OmniSwitch 6900

File Name	Base or Optional Software	Description
Tos.img	Base Software	CMM and NI Operating System
Trescue.img	Optional Software	Disaster Recovery file (used on USB flash drive)
boot.cfg		Switch Configuration File
boot.md5		Automatically created checksum file used to compare image files between the current running and certified directories.

CMM Software Directory Structure

The directory structure that stores the image and configuration files is divided into multiple parts:

- The ***certified*** directory contains files that have been certified by an authorized user as the default files for the switch. Should the switch reboot, it would reload the files in the ***certified*** directory to reactivate its functionality. Configuration changes CAN NOT be saved directly to the ***certified*** directory.
- The ***working*** *directory* contains files that may or may not be altered from the ***certified*** directory. The ***working*** directory is a holding place for new files. Files in the ***working*** directory must be tested before committing them to the ***certified*** directory. You can save configuration changes to the ***working*** directory.
- User-defined directories are any other directories created by the user. These directories are similar to the ***working*** directory in that they can contain image and configuration files. These directories can have any name and can be used to store additional switch configurations. Configuration changes CAN be saved directly to any user-defined directory.
- The RUNNING DIRECTORY is the directory that configuration changes will be saved to. Typically the RUNNING DIRECTORY is the directory that the switch booted from, however, any directory can be configured to be the RUNNING DIRECTORY.
- The RUNNING CONFIGURATION is the current operating configuration of the switch obtained from the directory the switch booted from in addition to any additional configuration changes made by the user. The RUNNING CONFIGURATION resides in the switch's RAM.

Where is the Switch Running From?

When a switch boots the RUNNING CONFIGURATION will come from either the ***certified, working,*** or a *user-defined* directory. A switch can be rebooted to run from any directory using the **reload from** command.

At the time of a normal boot (cold start or by using the **reload** command) the switch will do the following:

- 7.1.1 - Reboot from CERTIFIED directory.
- 7.2.1 - Reboot from CERTIFIED directory if contents (images and boot.cfg) are different from the RUNNING DIRECTORY. If contents are the same the switch will reboot from the RUNNING DIRECTORY.

If the RUNNING DIRECTORY is the ***certified*** directory, you will not be able to save any changes made to the RUNNING CONFIGURATION. If the switch reboots, any configuration changes will be lost. In order to save configuration changes the RUNNING DIRECTORY cannot be the **certified** directory.

You can determine where the switch is running from by using the **show running-directory** command described in "Show Currently Used Configuration" on page 3-16.

Software Rollback Feature

The directory structure inherent in the CMM software allows for a switch to return to a previous, more reliable version of image or configuration files.

Initially, when normally booting the switch, the software is loaded from the ***certified*** directory. This is the repository for the most reliable software. When the switch is booted, the ***certified*** directory is loaded into the RUNNING CONFIGURATION.

Changes made to the RUNNING CONFIGURATION will immediately alter switch functionality. However, these changes are not saved unless explicitly done so by the user using the **write memory** command. If the switch reboots before the RUNNING CONFIGURATION is saved, then the ***certified*** directory is reloaded to the RUNNING CONFIGURATION and configuration changes are lost.

New image or configuration files should always placed in the ***working*** or *or a user-defined* directory first. The switch can then be rebooted from that directory and be tested for a time to decide whether they are reliable. Once the contents of that directory are established as good files, then these files can be saved to the ***certified*** directory and used as the most reliable software to which the switch can be rolled back in an emergency situation.

Should the configuration or images files prove to be less reliable than their older counterparts in the ***certified*** directory, then the switch can be rebooted from the ***certified*** directory, and "rolled back" to an earlier version.

Software Rollback Configuration Scenarios

The examples below illustrate a few likely scenarios and explain how the RUNNING CONFIGURATION, *user-defined*, ***working***, and ***certified*** directories interoperate to facilitate the software rollback on a single switch.

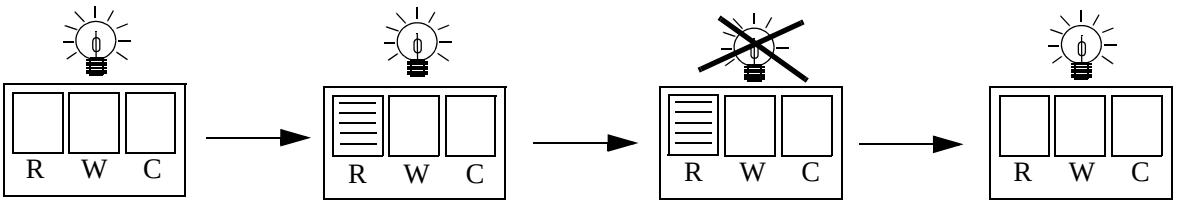
In the examples below, **R** represents the RUNNING CONFIGURATION, **W** represents the ***working*** directory, and **C** represents the ***certified*** directory.

Scenario 1: Running Configuration Lost After Reboot

Switch X is new from the factory and performs a cold reboot booting from the ***certified*** directory. Through the course of several days, changes are made to the RUNNING CONFIGURATION but not saved to a directory.

Power to the switch is interrupted, the switch reboots from the ***certified*** directory and all the changes in the RUNNING CONFIGURATION are lost since they weren't saved.

This is illustrated in the diagram below:



1. Switch boots from ***certified*** directory.
2. Changes are made to the running configuration but not saved.
3. Power is interrupted and the switch reboots.
4. Switch reboots from *certified* and all running configuration changes are lost since they weren't saved or certified.

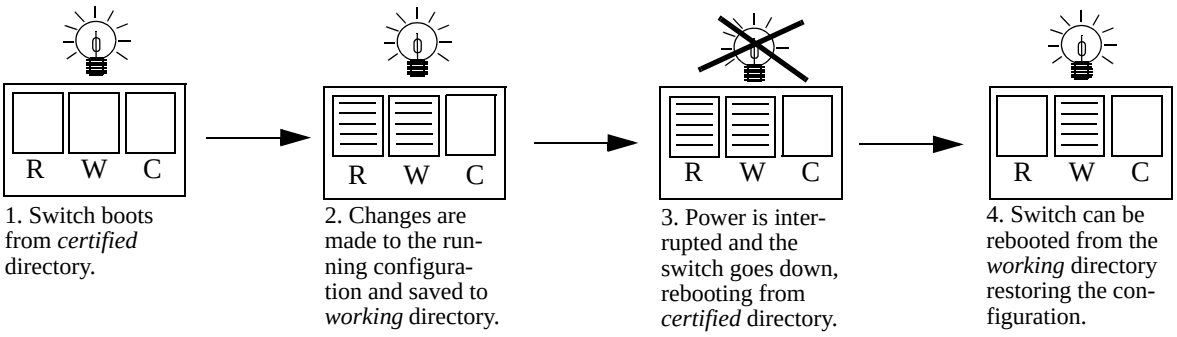
Running Configuration is Overwritten by the Certified Directory on Reboot

Scenario 2: Running Configuration Saved to the Working Directory

The network administrator recreates Switch X's RUNNING CONFIGURATION and immediately saves the running configuration to the ***working*** directory.

In another mishap, the power to the switch is again interrupted. The switch reboots rolls back to the ***certified*** directory. However, since the configuration file was saved to the ***working*** directory, that configuration can be retrieved.

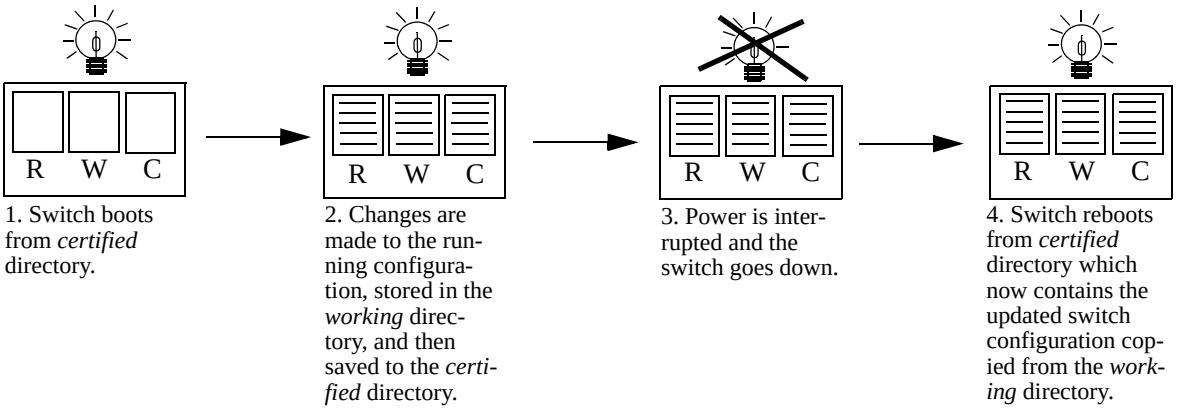
This is illustrated in the diagram below:



Running Configuration Saved to Working Directory

Scenario 3: Saving the Working to the Certified Directory

After running the modified configuration settings and checking that there are no problems, the network administrator decides that the modified configuration settings stored in the ***working*** directory are completely reliable. The administrator then decides to save the contents of the ***working*** directory to the ***certified*** directory. Once the ***working*** directory is saved to the ***certified*** directory, the modified configuration is included in a normal reboot.



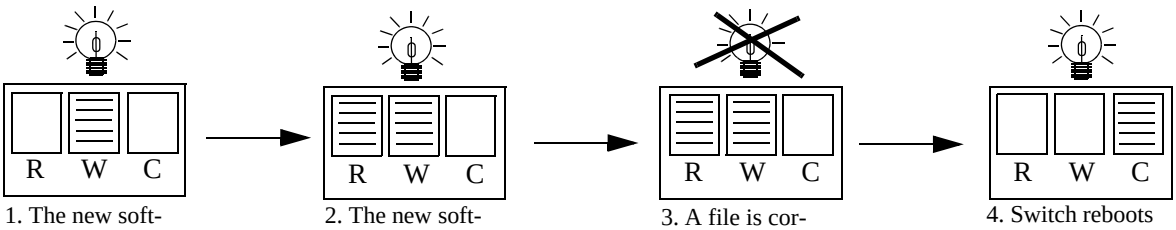
Running Configuration is Saved to Working Directory, then to the Certified Directory

Scenario 4: Rollback to Previous Version of Switch Software

Later that year, a software upgrade is performed. The network administrator loads the new software via FTP to the ***working*** directory and reboots the switch from that directory. Since the switch is specifically booted from the ***working*** directory, the switch is running from the ***working*** directory.

After the reboot loads the new software from the ***working*** directory, it is discovered that an image file was corrupted during the FTP transfer. Rather than having a disabled switch, the network administrator can reboot the switch from the ***certified*** directory (which has the previous, more reliable version of the software) and wait for a new version. In the meantime, the administrator's switch is still functioning.

This is illustrated below:



1. The new software is installed in the *working* directory.
2. The new software is loaded via a reboot from the *working* directory.
3. A file is corrupted and does not boot correctly.
4. Switch reboots from *certified* directory rolling back to the previous software version.

Switch Rolls Back to Previous Software Version

Redundancy

CMM software redundancy is one of the switch's most important fail over features. For CMM software redundancy, two fully-operational CMM modules must be installed at all times. In addition, the CMM software must be synchronized. (Refer to "Synchronizing the Primary and Secondary CMMs" on page 3-18 for more information.)

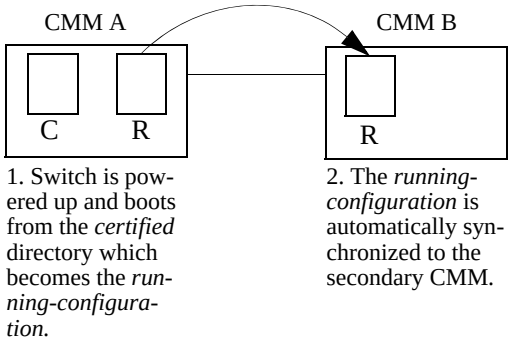
When two CMMs are running one CMM has the primary role and the other has the secondary role at any given time. The primary CMM manages the current switch operations while the secondary CMM provides backup (also referred to as "fail over").

Redundancy Scenarios

The following scenarios demonstrate how the CMM software is propagated to the redundant CMM In the examples below, **R** represents the RUNNING-CONFIGURATION directory and **C** represents the ***certified*** directory.

Scenario 1: Booting the Switch

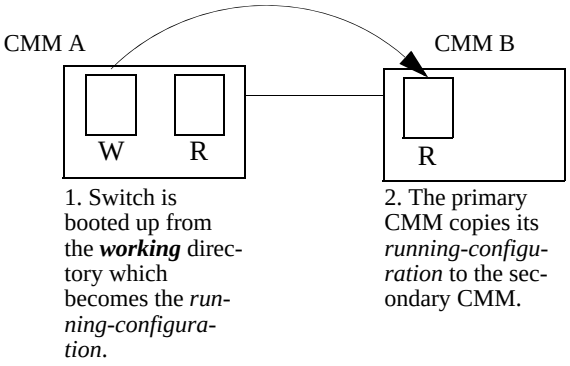
The following diagram illustrates what occurs when a switch powers up.



Powering Up a Switch

Scenario 2: Rebooting from the Working Directory

After changes to the *configuration* and *image* files are saved to the ***working*** directory, sometimes it is necessary to boot from the ***working*** directory to check the validity of the new files. The following diagram illustrates the synchronization process of a ***working*** directory reboot.



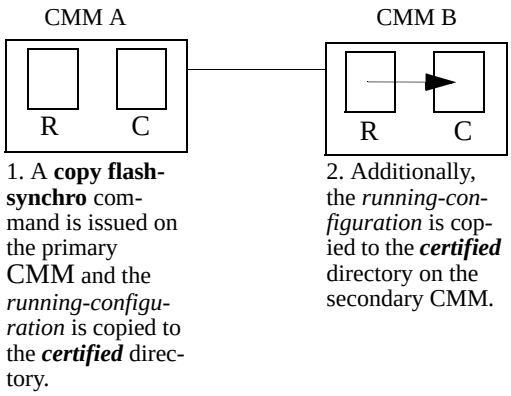
Booting from the Working Directory

Note. It is important to certify the *RUNNING-DIRECTORY* and synchronize the CMMS as soon as the validity of the software is established. Switches booted from the *RUNNING-DIRECTORY* are at risk of mismanaging data traffic due to incompatibilities in different versions of switch software. Certifying the *RUNNING-DIRECTORY* is described in "Copying the RUNNING DIRECTORY to the Certified Directory" on page 3-15, while synchronizing the switch is described in "Synchronizing the Primary and Secondary CMMs" on page 3-18.

Scenario 3: Synchronizing CMMs

When changes have been saved to the primary CMM ***certified*** directory, these changes need to be propagated to the secondary CMM using the **copy flash-synchro** command.

The following diagram illustrates the process that occurs when synchronizing CMMs.



Synchronizing CMMs

The **copy flash-synchro** command (described in "Synchronizing the Primary and Secondary CMMs" on page 3-18) can be issued on its own, or in conjunction with the **copy running certified** command (described in "Synchronizing the Primary and Secondary CMMs" on page 3-18).

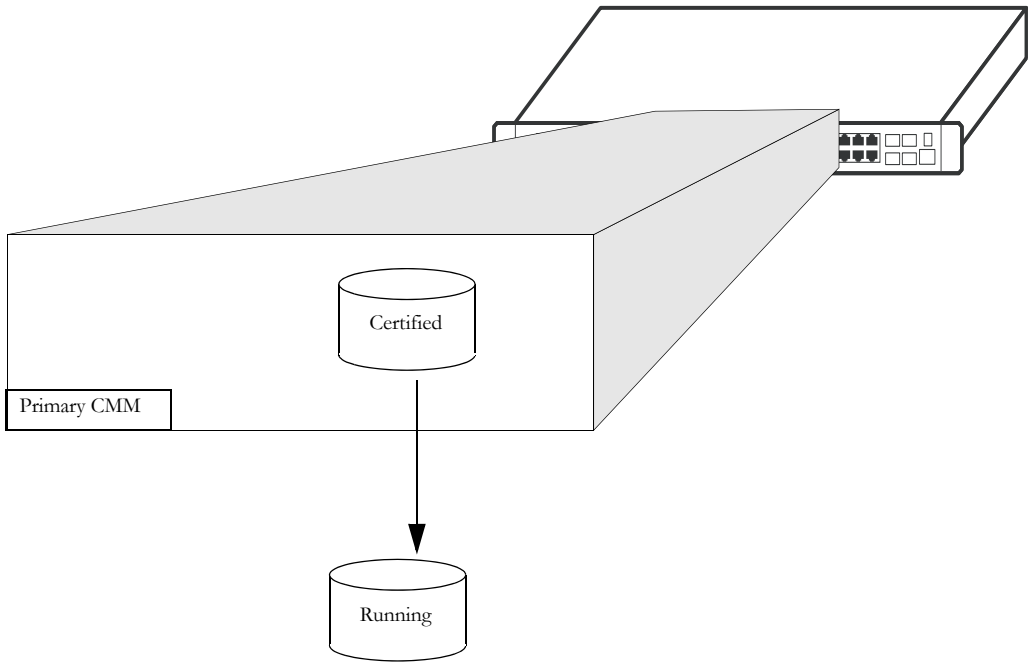
Note. It is important to certify the CMMs as soon as the validity of the software is established. Unsynchronized CMMs are at risk of mismanaging data traffic due to incompatibilities in different versions of switch software.

Managing Switch Configurations - Single CMM

The following sections define commands that allow the user to manipulate the files in the directory structure of a single CMM.

Rebooting the Switch

When booting the switch, the software in the ***certified*** directory is loaded into the RAM memory of the switch and used as a running configuration, as shown:



The ***certified*** directory software should be the best, most reliable versions of both the image files and the **boot.cfg** file (configuration file). The switch will run from the ***certified*** directory after a cold boot or if the **reload** command is issued with no additional parameters.

To reboot the switch from the ***certified*** directory, enter the **reload all** command at the prompt:

```
-> reload all
```

This command loads the image and configuration files in the ***certified*** directory into the RAM memory.

Note. When the switch reboots it will boot from the ***certified*** directory. Any information in the RUNNING CONFIGURATION that has not been saved will be lost.

Scheduling a Reboot

It is possible to cause a reboot of the CMM at a future time by setting time parameters in conjunction with the **reload** command, using the **in** or **at** keywords.

To schedule a reboot of the primary CMM in 3 hours and 3 minutes, you would enter:

```
-> reload all in 3:03
```

To schedule a reboot for June 30 at 8:00pm, you would enter:

```
-> reload all at 20:00 june 30
```

Note. Scheduled reboot times should be entered in military format (i.e., a twenty-four hour clock).

Cancelling a Scheduled Reboot

To cancel a scheduled reboot, use the **cancel** keyword. For example, to cancel the reboot set above, enter the following:

```
-> reload all cancel
```

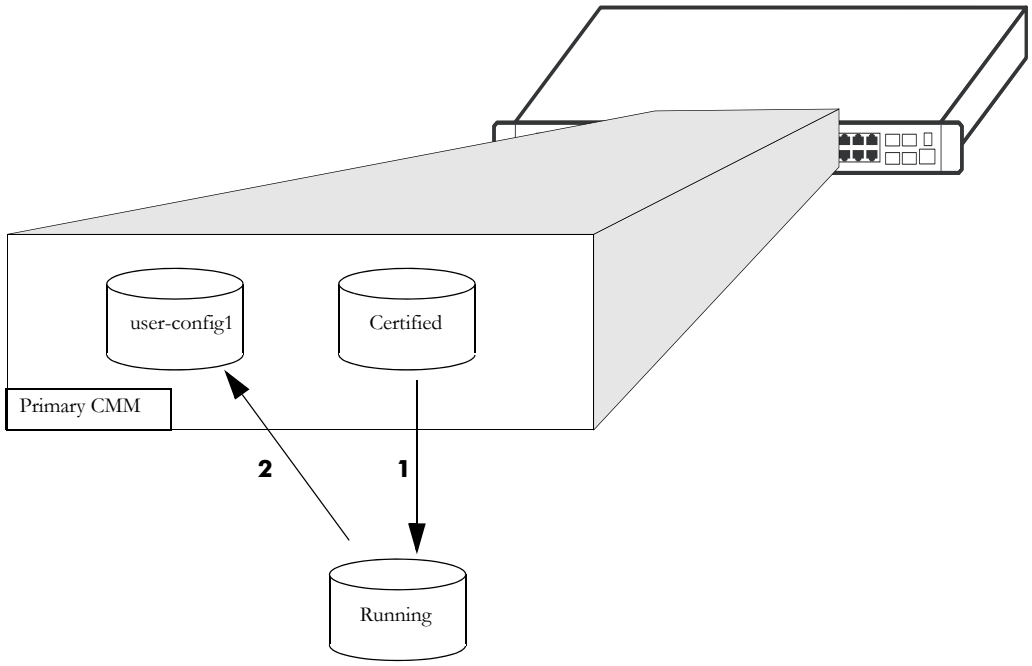
Checking the Status of a Scheduled Reboot

You can check the status of a reboot set for a later time by entering the following command:

```
-> show reload
```

Saving the Running Configuration

Once the switch has booted and is running, a user can modify various parameters of switch functionality. These changes are stored temporarily in the RUNNING CONFIGURATION in the RAM of the switch. In order to save these changes, the RUNNING CONFIGURATION must be saved.



In this diagram:

1 The switch boots from the ***certified*** directory, and the software is loaded to the RAM to create a RUNNING CONFIGURATION. The ***certified*** directory is the RUNNING DIRECTORY.

2 Changes are made to the RUNNING CONFIGURATION and need to be saved.

3 Since configuration changes cannot be saved directly to the ***certified*** directory, the RUNNING DIRECTORY needs to be changed to a different directory before saving the changes.

To change the running directory to a directory other than the ***certified*** use the **modify running-directory** command as shown and then save the configuration with the **write memory** command:

```
-> modify running-directory user-config1

-> write memory
```

Rebooting from a Directory

Besides a regular boot of the switch (from the ***certified*** directory), you can also force the switch to boot from a different directory. This is useful for checking whether a new configuration or image file will boot the switch correctly, before committing it to the ***certified*** directory.

The following steps explain the case of a switch being rebooted from the ***working*** directory, however any user-defined directory can be specified:

1 The ***certified*** directory is used to initially boot the switch.

2 Changes are made to the configuration file and are saved to the configuration file in the ***working*** directory by using the **write memory** command.

3 The switch is rebooted from the ***working*** directory by using the **reload from** command.

To reboot the switch from the ***working*** directory, enter the following command at the prompt, along with a timeout period (in minutes), as shown:

```
-> reload from working rollback-timeout 5
```

At the end of the timeout period, the switch will reboot again normally, as if a **reload** command had been issued.

Rebooting the Switch from a directory with No Rollback Timeout

It is possible to reboot from a directory without setting a rollback timeout, in the following manner:

```
-> reload from working no rollback-timeout
```

Scheduling a Directory Reboot

It is possible to cause a directory reboot of the CMM at a future time by setting time parameters in conjunction with the **reload from** command, using the **in** or **at** keywords. You will still need to specify a rollback time-out time, or that there is no rollback.

To schedule a ***working*** directory reboot of the CMM in 3 hours and 3 minutes with no rollback time-out, you would enter:

```
-> reload from working no rollback-timeout in 3:03
```

To schedule a ***working*** directory reboot of the CMM at 8:00pm with a rollback time-out of 10 minutes, you would enter:

```
-> reload from working rollback-timeout 10 at 20:00
```

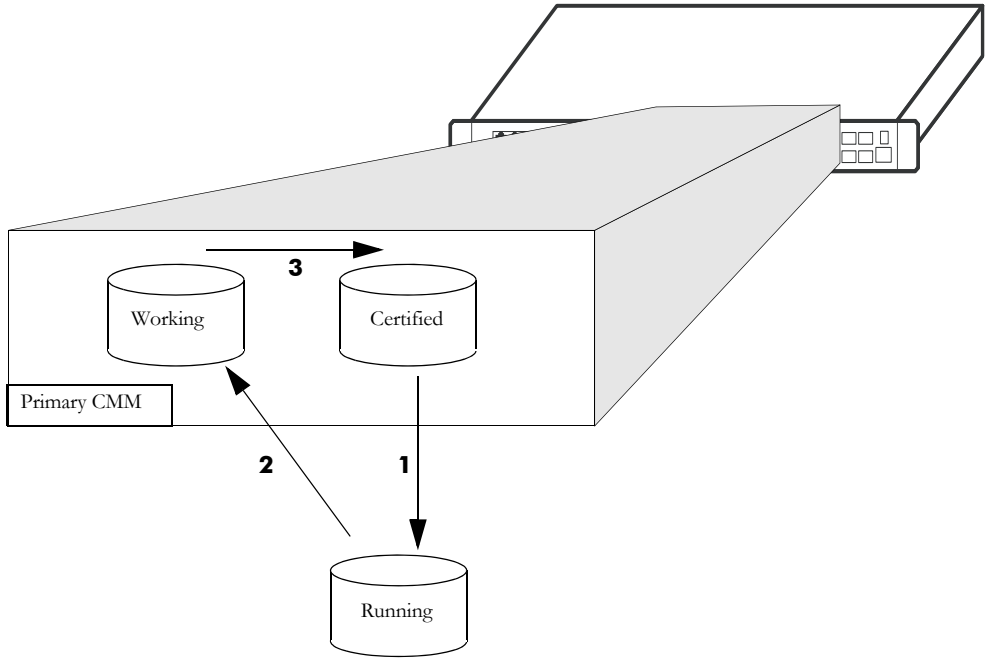
Cancelling a Rollback Timeout

To cancel a rollback time-out, enter the **reload cancel** command as shown:

```
-> reload cancel
```

Copying the RUNNING DIRECTORY to the Certified Directory

When the RUNNING CONFIGURATION is saved to the RUNNING DIRECTORY, the switch's RUNNING DIRECTORY and ***certified*** directories are now different. This difference, if the CMM reboots, causes the switch to boot and run from the ***certified*** directory. When the switch is booted and run from the ***certified*** directory, changes made to switch functionality cannot be saved. The **boot.cfg** file saved in the RUNNING DIRECTORY needs to be saved to the ***certified*** directory, as shown:



In this diagram, the ***working*** directory is the RUNNING DIRECTORY:

1 The switch boots from the ***certified*** directory and changes are made to the RUNNING CONFIGURATION.

2 The RUNNING DIRECTORY is changed from ***certified*** to a different directory such as ***working***.

```
-> modify running-directory working
```

3 The changes are saved to the ***working*** directory in the **boot.cfg** file.

```
-> write memory
```

4 The contents of the ***working*** directory are saved to the ***certified*** directory.

```
-> copy running certified
```

Show Currently Used Configuration

Depending on how a a switch is booted different directories can become the RUNNING DIRECTORY. See "Where is the Switch Running From?" on page 3-4. for additional information.

To check the directory from where the switch is currently running, enter the following command:

```
-> show running-directory

CONFIGURATION STATUS
   Running CMM              : PRIMARY,
   CMM Mode                 : DUAL CMMs,
   Current CMM Slot         : A,
   Running configuration    : WORKING,
   Certify/Restore Status   : CERTIFY NEEDED
SYNCHRONIZATION STATUS
   Running Configuration    : NOT AVAILABLE,
```

The command returns the directory the switch is currently running from and which CMM is currently controlling the switch (primary or secondary). It also displays wheter the switch is synchronized.

Show Switch Files

The files currently installed on a switch can be viewed using the **show microcode** command. This command displays the files currently in the specified directory.

To display files on a switch, enter the **show microcode** command with a directory, as shown:

```
-> show microcode certified
   Package           Release       Size     Description
-----------------+--------------+--------+---------------------------------
Ros.img            7.1.1.311.R01    2486643 Alcatel-Lucent OS
Reni.img           7.1.1.311.R01     941331 Alcatel-Lucent NI
```

If no directory is specified, the files that have been loaded into the running configuration are shown.

Managing CMM Redundancy

The following section describe circumstances that the user should be aware of when managing the CMM directory structure on a switch with redundant CMMs. It also includes descriptions of the CLI commands designed to synchronize software between the primary and secondary CMMs.

Rebooting the Secondary CMM

You can specify a reboot of the secondary CMM by using the **secondary** keyword in conjunction with the **reload** command. For example, to reboot the secondary CMM, enter the **reload** command as shown:

```
-> reload secondary
```

In this case, the primary CMM continues to run, while the secondary CMM reboots.

Scheduling a Reboot

It is possible to cause a reboot of the secondary CMM at a future time by setting time parameters in conjunction with the **reload** command.

For example, to schedule a reboot of the secondary CMM in 8 hours and 15 minutes on the same day, enter the following at the prompt:

```
-> reload secondary in 08:15
```

Cancelling a Scheduled Reboot

To cancel a scheduled reboot, use the **cancel** keyword. For example, to cancel the secondary reboot set above, enter the following:

```
-> reload secondary cancel
```

Secondary CMM Fail Over

If the Primary CMM fails the switch will “fail over” to the secondary CMM. “Fail over” means the secondary CMM takes the place of the primary CMM. This prevents the switch from ceasing functionality during the boot process.

Synchronizing the primary and secondary CMMs is done using the **copy flash-synchro** command described in “Synchronizing the Primary and Secondary CMMs” on page 3-18.

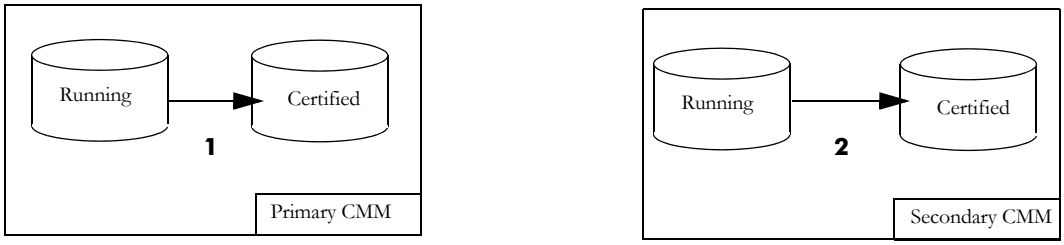
Synchronizing the Primary and Secondary CMMs

If you have a secondary CMM in your switch, it will be necessary to synchronize the software between the primary and secondary CMMs. If the primary CMM goes down then the switch fails over to the secondary CMM. If the software in the secondary CMM is not synchronized with the software in the primary CMM, the switch will not function as configured by the administrator.

At the same time that you copy the RUNNING DIRECTORY to the ***certified*** directory, you can synchronize the secondary CMM with the primary CMM. To copy the RUNNING DIRECTORY to the ***certified*** directory of the primary CMM and at the same time synchronize the software of the primary and secondary CMM, use the following command:

```
-> copy running certified flash-synchro
```

The synchronization process is shown in the diagram below:



In the above diagram:

1 The primary CMM copies its RUNNING-CONFIGURATION to the ***certified*** directory.

2 Since the RUNNING-CONIFIGURATION is always synchronzied between redundant CMMs, the secondary CMM copies its RUNNING-CONIFIGURATION to the ***certified*** directory.

To just synchronize the secondary CMM to the primary CMM, enter the following command at the prompt:

```
-> copy flash-synchro
```

The **copy flash-synchro** command is described in detail in the *OmniSwitch CLI Reference Guide*.

Swapping the Primary CMM for the Secondary CMM

If the primary CMM is having problems, or if it needs to be shut down, then the secondary CMM can be instructed to "take over" the switch operation as the primary CMM is shut down. It's normal for the NIs to indicate a DOWN status for approximately 10 seconds while establishing communication to the secondary CMM, however this does not affect the flow of traffic.

Note. It is important that the software for the secondary CMM has been synchronized with the primary CMM before you initiate a secondary CMM takeover. If the CMMs are not synchronized, the takeover could result in the switch running old or out-of-date software. Synchronizing the primary and secondary CMMs is described in "Synchronizing the Primary and Secondary CMMs" on page 3-18.

To instruct the secondary CMM to takeover switch functions from the primary CMM, enter the following command at the prompt:

```
-> takeover
```

The **takeover** command is described in detail in the *OmniSwitch CLI Reference Guide*.

Note. The saved **boot.cfg** file will be overwritten if the **takeover** command is executed after the **copy flash-synchro** command on a switch set up with redundant CMMs.

Show Currently Used Configuration

In a chassis with a redundant CMMs, the display for the currently running configuration tells the user if the primary and secondary CMMs are synchronized.

To check the directory from where the switch is currently running and if the primary and secondary CMMs are synchronized, enter the following command on a stack:

```
-> show running-directory

CONFIGURATION STATUS
   Running CMM              : PRIMARY,
   CMM Mode                 : DUAL CMMs,
   Current CMM Slot         : 1,
   Running configuration    : WORKING,
   Certify/Restore Status   : CERTIFY NEEDED
SYNCHRONIZATION STATUS
   Flash Between CMMs       : SYNCHRONIZED,
   Running Configuration    : NOT AVAILABLE,
```

The **show running-directory** command is described in detail in the *OmniSwitch CLI Reference Guide*.

Using the USB Flash Drive

An Alcatel-Lucent certified USB flash drive can be connected to the CMM and used to transfer images to and from the flash memory on the switch. This can be used for upgrading switch code, backing up files or recovering a failed CMM. For the automatic upgrades and disaster recovery the USB flash drive must be configured with the proper directory structure, depending on the platform, as noted in the table below. Once the flash drive is properly mounted a directory named /*uflash* is automatically created. Files can then be copied to and from the /*uflash* directory.

The directories below must be created on the USB flash drive for feature support and in lower case.

Product Family Name	Auto-Copy Support	Disaster-Recovery Support
OmniSwitch 10K	10000/working	10000/working 10000/certified
OmniSwitch 6900	6900/working	6900/working 6900/certified

Transferring Files Using a USB Flash Drive

The following is an example of how to mount and transfer files using the USB flash drive using the **usb** and **mount** commands.

```
-> usb enable
-> mount /uflash
-> cp /flash/working/boot.cfg /uflash/boog.cfg
-> umount /uflash
```

Once the USB flash drive is mounted most common file and directory commands can be performed on the /*uflash* directory.

Automatically Copying Code Using a USB Flash Drive

The switch can be configured to automatically mount and copy image files from the USB flash drive as soon as it's connected. This can be used to automatically upgrade code. In order to prevent an accidental upgrade, a file named *aossignature* must be stored on the USB flash drive as well as having a directory with the same name as the product family as noted in the table above. The following is an example using the **usb auto-copy** command

Note: The ***aossignature*** file can be an empty text file**.**

1. Create a file named *aossignature* in the root of the USB flash drive.
2. Create a directory named *10000/working* on the USB flash drive with all the proper image files.
3. -> usb enable
4. -> usb auto-copy enable
5. Connect the USB flash drive to the CMM; the images will be validated and copied to the /*flash*/*working* directory of the CMM and the *boot.cfg* file in the */flash/working* directory will be updated or created

using the running setup. The switch will then reboot from the *working* directory applying the code upgrade.

6 Once the switch reboots the auto-copy feature will automatically be disabled to prevent another upgrade.

Disaster Recovery Using a USB Flash Drive

A USB flash drive can be loaded with the necessary files to recover a failed CMM. This can be used if the image files on the CMM become corrupted, deleted, or the switch is unable to boot from the CMM for other reasons. Perform the following steps to run Disaster Recovery:

Note: It's recommended to prepare the USB flash drive prior to needing it for disaster recovery**.** This example is for an OS10K, use the proper directory names based on the platform (i.e. *10000* or *6900*)

---

1 Create the directory structure *10000/certified* and *10000/working* on the USB flash drive with the backup system and configuration files.

2 Copy the **Rrescue.img** file to the root directory on the USB flash drive.

3 Connect the USB flash drive to the CMM and reboot. The switch will automatically stop and display the option to perform a disaster recovery.

4 Enter the '**run rescue**' command from miniboot/uboot and follow the recovery prompts.

Once complete, the CMM will reboot and be operational again.

In-Service Software Upgrade

The In-Service Software Upgrade (ISSU) feature is used to upgrade switch software with minimal disruption to data traffic.

ISSU Specifications

CMMs	Must be synchronized and certified redundant CMMs
Image Files	**Ros.img** **Reni.img**
Validation File	***issu_version***
ISSU Directory	Any user-defined directory to store the image files.
NI Reset Timer	120 minutes
Control LED	Blinks amber during ISSU upgrade

The Validation File

The Validation File contains the information required to validate that an ISSU upgrade is possible. An ISSU upgrade is dependant upon the current version of software on the switch and the version of software the switch is being upgraded to. If the version of code on the switch is not ISSU compatible with the version being upgraded, the ISSU upgrade will not be allowed and an error message similar to the one below will be displayed:

```
Tue Dec 14 14:19:15 : ChassisSupervisor issuMgr alert message:
+++ ISSU Image Validation Failed - aborting ISSU
ERROR: ISSU Validation Error: Images not issu compatible
```

- Image files and the '*issu_version*' file are available from Service & Support.
- View the *'issu_version'* text file to determine ISSU compatibility between code versions before attempting an ISSU upgrade or contact Service and Support.

Resetting NIs

After performing an ISSU upgrade the NIs must be reset to complete the ISSU upgrade. They can be reset manually using the '**issu slot**' or '**reload slot**'commands. If the NIs are not reset by the time the NI reset timer expires (Refer to "ISSU Specifications" on page 3-23), they will be reset individually by the system in ascending order beginning with slot 1. Once the reset NI reaches a ready state, the next one is reset. This process continues until all NIs have been reset.

ISSU Guidelines

- The current and new versions of code must be ISSU compatible.
- To complete an ISSU upgrade NIs must be reset either manually or automatically.
- NIs can be manually reset one right after the other. However, it's recommended to wait until the previous NI is operational before resetting the next NI.

- No configuration is allowed until the entire ISSU upgrade process is complete, including NI reset.

Performing an ISSU Upgrade

Note. The example below uses a directory named '***issu_dir***'. However, the directory can be any *user-defined* directory.

1 Ensure that the switch is fully synchronized and certified.

2 Create the **flash/issu_dir** directory and copy the image files and the ***issu_version*** file to the **/flash/issu_dir** directory.

3 Enter '**issu from issu_dir**' to begin the ISSU upgrade.

4 The switch copies the **/flash/issu_dir** directory to the secondary CMM, reloads the secondary CMM with the upgraded code, and the secondary becomes the new primary CMM.

5 The old primary CMM becomes the secondary CMM and reloads using the upgraded code in the **flash/issu_dir** directory.

As a result of this process both CMMs are now running with the upgraded code and the primary and secondary CMMs will have changed roles (i.e., primary will act as secondary and the secondary as primary).

6 Reset the individual NIs using the **issu slot** command.

7 Enter '**copy running certified flash-synchro'** to certify the configuration.

8 Verify the status of the ISSU upgrade using the **show issu status** command.

Displaying CMM Conditions

To show various CMM conditions, such as where the switch is running from and which files are installed, use the following CLI show commands:

show running-directory	Shows the directory from where the switch was booted.
show reload	Shows the status of any time delayed reboot(s) that are pending on the switch.
show microcode	Displays microcode versions installed on the switch.
usb	Enables access to the device connected to the USB port.

For more information on the resulting displays from these commands, see the *OmniSwitch CLI Reference Guide*.

4 Using the CLI

Alcatel-Lucent's Command Line Interface (CLI) is a text-based configuration interface that allows you to configure switch applications and to view switch statistics. Each CLI command applicable to the switch is defined in the *OmniSwitch CLI Reference Guide*. All command descriptions listed in the Reference Guide include command syntax definitions, defaults, usage guidelines, example screen output, and release history.

This chapter describes various rules and techniques that will help you use the CLI to its best advantage. This chapter includes the following sections:

CLI Specifications

The following table lists specifications for the Command Line Interface.

Platforms Supported	OmniSwitch 10K, 6900
Configuration Methods	• Online configuration via real-time sessions using CLI commands. • Offline configuration using text file holding CLI commands.
Command Capture Feature	Snapshot feature captures switch configurations in a text file.
User Service Features	• Command Line Editing • Command Prefix Recognition • CLI Prompt Option • Command Help • Keyword Completion • Command History • Command Logging • Syntax Error Display • More Command

CLI Overview

The CLI uses single-line text commands that are similar to other industry standard switch interfaces. However, the OmniSwitch CLI is different from industry standard interfaces in that it uses a single level command hierarchy.

Unlike other switch interfaces, the CLI has no concept of command modes. Other CLIs require you to step your way down a tree-type hierarchy to access commands. Once you enter a command mode, you must step your way back to the top of the hierarchy before you can enter a command in a different mode. The OmniSwitch will accept any CLI command at any time because there is no hierarchy.

Online Configuration

To configure parameters and view statistics you must connect the switch to a terminal, such as a PC or UNIX workstation, using terminal emulation software. This connection can be made directly to the switch's serial port or over a network via Telnet.

Once you are logged in to the switch, you may configure the switch directly using CLI commands. Commands executed in this manner normally take effect immediately. The majority of CLI commands are independent, single-line commands and therefore can be entered in any order. However, some functions may require you to configure specific network information before other commands can be entered. For example, before you can assign a port to a VLAN, you must first create the VLAN. For information about CLI command requirements, refer to the *OmniSwitch CLI Reference Guide*.

Offline Configuration Using Configuration Files

CLI configuration commands can be typed into a generic text file. When the text file is placed on the switch its commands are applied to the switch when the **configuration apply** command is issued. Files used in this manner are called configuration files.

A configuration file can be viewed or edited offline using a standard text editor. It can then be uploaded and applied to additional switches in the network. This allows you to easily clone switch configurations. This ability to store comprehensive network information in a single text file facilitates troubleshooting, testing, and overall network reliability.

See Chapter 5, "Working With Configuration Files," for detailed information about configuration files.

Command Entry Rules and Syntax

When you start a session on the switch, you can execute CLI commands as soon as you are logged in. The following rules apply:

- Enter only one command per line.
- Passwords are case sensitive.
- Commands are *not* case sensitive. The switch accepts commands entered in upper case, lower case or a combination of both.
- Press Enter to complete each command line entry.
- To use spaces within a user-defined text string, you must enclose the entry in quotation marks (" ").
- If you receive a syntax error (i.e., ERROR: Invalid entry:), double-check your command as written and re-enter it exactly as described in the *OmniSwitch CLI Reference Guide*. Be sure to include all syntax option parameters.
- To exit the CLI, type **exit** and press Enter.

Text Conventions

The following table contains text conventions and usage guidelines for CLI commands as they are documented in this manual.

bold text	Indicates basic command and keyword syntax. Example: **show snmp station**
" " (Quotation Marks)	Used to enclose text strings that contain spaces Example: **vlan 2 name "new test vlan"**

Using "Show" Commands

The CLI contains **show** commands that allow you to view configuration and switch status on your console screen. The **show** syntax is used with other command keywords to display information pertaining to those keywords.

For example, the **show vlan** command displays a table of all VLANs currently configured, along with pertinent information about each VLAN. Different forms of the **show vlan** command can be used to display different subsets of VLAN information. For example the **show vlan rules** command displays all rules defined for a VLAN.

Using the "No" Form

The *OmniSwitch CLI Reference Guide* defines all CLI commands and explains their syntax. Whenever a command has a "no" form, it is described on the same page as the original command. The "no" form of a command will mean one of the following:

- It can remove the configuration created by a command. For example, you create a VLAN with the **vlan** command, and you delete a VLAN with the **no vlan** command.

Partial Keyword Completion

The CLI has a partial keyword recognition feature that allows the switch to recognize partial keywords to CLI command syntax. Instead of typing the entire keyword, you may type only as many characters as is necessary to uniquely identify the *keyword*, then press the Tab key. The CLI will complete the keyword and place the cursor at the end of the keyword.

When you press Tab to complete a command keyword, one of four things can happen:

- You enter enough characters (prior to Tab) to uniquely identify the command keyword.

In this case, pressing Tab will cause the CLI to complete the keyword and place a space followed by the cursor at the end of the completed keyword.

- You do not enter enough characters (prior to Tab) to uniquely identify the command keyword.

In this case pressing Tab will list all of the possible parameters. .

- You enter characters that do not belong to a keyword that can be used in this instance.

In this case, pressing Tab will have no effect.

- You enter enough characters (prior to Tab) to uniquely identify a group of keywords such that all keywords in the group share a common prefix.

In this case, pressing Tab will cause the CLI to complete the common prefix and place the cursor at the end of the prefix. Note that in this case, no space is placed at the end of the keyword.

Command Help

The CLI has an internal help feature you can invoke by using the question mark (**?**) character as a command. The CLI help feature provides progressive information on how to build your command syntax, one keyword at a time.

If you do not know the first keyword of the command you need, you can use a question mark character at the CLI system prompt. The CLI responds by listing command keywords divided into command sets. You can find the first keyword for the command you need by referring to the list on your screen. The following is a partial display:

```
-> ?
  WHOAMI WHO VERBOSE USB USER UPDATE UMOUNT TTY SYSTEM SWLOG SHOW SESSION NTP
  NSLOOKUP NO NEWFS MOUNT MODIFY KILL IPV6 IP FSCK FREESPACE DEBUG
  COMMAND-LOG CHMOD
(System Service & File Mgmt Command Set)

  POWER POWERSUPPLY WRITE TEMP-THRESHOLD TAKEOVER SYSTEM SHOW RRM RLS RELOAD
  RDF RCP NO MULTI-CHASSIS MODIFY ISSU HASH-CONTROL DEBUG COPY CLEAR <cr>
(CMM Chassis Supervision Command Set)
```

(Additional output not shown)

Note that the command keywords are shown in all capital letters. The name of the command set is listed parenthetically *below* the keywords in initial caps.

Recalling the Previous Command Line

To recall the last command executed by the switch, press either the Up Arrow key or the **!!** (bang, bang) command at the prompt and the previous command will display on your screen.

In the following example, the **ls** command is used to list the contents of the switch's **/flash/switch** directory.

```
-> ls

Listing Directory /flash/switch:

drw      2048 Jan  1  1980 ./
drw      2048 Jan  3 19:23 ../
-rw       308 Jan  1  1980 banner_default.txt

         9850880 bytes free
->
```

To enter this same command again, use the Up Arrow key. The **ls** command appears at the prompt. To issue the **ls** command, press Enter.

```
-> ls
```

The **!!** (bang, bang) command will display the last command line entered and automatically run the command.

Inserting Characters

To insert a character between characters already typed, use the Left and Right Arrow keys to place the cursor into position, then type the new character. Once the command is correct, execute it by pressing Enter. In the following example, the user enters the wrong syntax to execute the command. The result is an error message.

```
-> show mirocode
ERROR: Invalid entry: "mirocode"
```

To correct the syntax without retyping the entire command line, use the up arrow to recall the previous syntax. Then, use the Left Arrow key to edit the command as needed.

```
-> show microcode
```

To execute the corrected command, press Enter.

Command History

The **history** command allows you to view commands you have recently issued to the switch. The switch has a history buffer that stores the most recently executed commands.

Note. The **command history** feature differs from the **command logging** feature in that command logging stores the most recent commands in a separate **command.log** file. Also, the command logging feature includes additional information, such as full command syntax, login user name, entry date and time, session IP address, and entry results. For more information on command logging, refer to "Logging CLI Commands and Entry Results" on page 4-7.

You can display the commands in a numbered list by using the **history** command. The following is a sample list:

```
-> history
1 show cmm
2 show fantray
3 show vlan
4 show temperature
5 ip load dvmrp
6 show arp
7 clear arp
8 show ip config
9 ip helper max hops 5
10 show ip interface
11 show vlan
12 history
```

You can recall commands shown in the history list by using the exclamation point character (**!**) also called "bang". To recall the command shown in the history list at number 4, enter **!4** (bang, 4). The CLI will respond by printing the number four command at the prompt. Using the history list of commands above, the following would display:

```
-> !4
-> show ip interface
```

Logging CLI Commands and Entry Results

The switch provides command logging via the **command-log** command. This feature allows users to record the most recent commands entered via Telnet, Secure Shell, and console sessions. In addition to a list of commands entered, the results of each command entry are recorded. Results include information such as whether a command was executed successfully, or whether a syntax or configuration error occurred.

Refer to the sections below for more information on configuring and using CLI command logging. For detailed information related to command logging commands, refer to the *OmniSwitch CLI Reference Guide*.

Enabling Command Logging

By default, command logging is *disabled*. To enable command logging on the switch, enter the following command:

```
-> command-log enable
```

When command logging is enabled via the **command-log enable** syntax, a file called **command.log** is automatically created in the switch's **flash** directory. Once enabled, configuration commands entered on the command line will be recorded to this file until command logging is disabled.

Note. The **command.log** file cannot be deleted while the command logging feature is enabled. Before attempting to remove the file, be sure to disable command logging. To disable command logging, refer to the information below.

Disabling Command Logging

To disable the command logging, simply enter the following command:

```
-> command-log disable
```

Disabling command logging *does not* automatically remove the **command.log** file from the **flash** directory. All commands logged *before* the **command-log disable** syntax was entered remains available for viewing. For information on viewing logged commands, along with the command entry results, refer to "Viewing Logged CLI Commands and Command Entry Results" on page 4-8.

Viewing the Current Command Logging Status

As mentioned above, the command logging feature is disabled by default. To view whether the feature is currently enabled or disabled on the switch, use the **show command-log status** command. For example:

```
-> show command-log status
CLI command logging: Enable
```

In this case, the feature has been enabled by the user via the **command-log** command. For more information on enabling and disabling command logging, refer to the sections above.

Viewing Logged CLI Commands and Command Entry Results

To view a list of logged commands, along with the corresponding information (including entry results), enter the **show command-log** command. For example:

```
-> show command-log
Command : ip interface vlan-68 address 168.14.12.120 vlan 68
  UserName : admin
  Date     : MON APR 28 01:42:24
  Ip Addr  : 128.251.19.240
  Result   : SUCCESS

Command : ip interface vlan-68 address 172.22.2.13 vlan 68
  UserName : admin
  Date     : MON APR 28 01:41:51
  Ip Addr  : 128.251.19.240
  Result   : ERROR: Ip Address must not belong to IP VLAN 67 subnet

Command : ip interface vlan-67 address 172.22.2.12 vlan 67
  UserName : admin
  Date     : MON APR 28 01:41:35
  Ip Addr  : 128.251.19.240
  Result   : SUCCESS

Command : command-log enable
  UserName : admin
  Date     : MON APR 28 01:40:55
  Ip Addr  : 128.251.19.240
  Result   : SUCCESS
```

The **show command-log** command lists commands in *descending order* (the most recent commands are listed first). In the example above, the **command-log enable** syntax is the least recent command logged; the **ip interface vlan-68 address 168.14.12.120 vlan 68** syntax is the most recent.

- **Command.** Shows the exact syntax of the command, as entered by the user.
- **UserName.** Shows the name of the user session that entered the command. For more information on different user session names, refer to Chapter 6, “Managing Switch User Accounts.”
- **Date.** Shows the date and time, down to the second, when the command was originally entered.
- **IP Addr.** The IP address of the terminal from which the command was entered.
- **Result.** The outcome of the command entry. If a command was entered successfully, the syntax **SUCCESS** displays in the Result field. If a syntax or configuration error occurred at the time a command was entered, details of the error display. For example:

  ```
  Result   : ERROR: Ip Address must not belong to IP VLAN 67 subnet
  ```

Customizing the Screen Display

The CLI has several commands that allow you to customize the way switch information is displayed to your screen. You can make the screen display smaller or larger. You can also adjust the size of the table displays and the number of lines shown on the screen.

Note. Screen display examples in this chapter assume the use of a VT-100/ASCII emulator.

Changing the Screen Size

You may specify the size of the display shown on your terminal screen by using the **tty** command. This command is useful when you have a small display screen or you want to limit the number of lines scrolled to the screen at one time. For example, to limit the number of lines to 10 and the number of columns to 150, enter the following:

```
-> tty 10 150
```

The first number entered after **tty** defines the number of lines on the screen. It must be a number between 10 and 150. The second number after **tty** defines the number of columns on the screen. It must be a number between 20 and 150. You may view the current setting for your screen by using the **tty** command.

Changing the CLI Prompt

You can change the system prompt that displays on the screen when you are logged into the switch. The default prompt consists of a dash, greater-than (->) text string. To change the text string that defines the prompt from -> to ##=> use the **session prompt** command as follows:

```
->
-> session prompt default ##=>
##=>
```

The switch displays the new prompt string after the command is entered.

Verifying CLI Usage

To display information about CLI commands and the configuration status of your switch, use the **show** commands listed here:

show session config	Displays session manager configuration information (e.g., default prompt, banner file name, and inactivity timer).
show prefix	Shows the command prefix (if any) currently stored by the CLI. Prefixes are stored for command families that support the prefix recognition feature.
history	Displays commands you have recently issued to the switch. The commands are displayed in a numbered list.
telnet	Shows the enable status of the more mode along with the number of lines specified for the screen display.

For more information about the resulting displays from these commands, see the *OmniSwitch CLI Reference Guide*. Additional information can also be found in "Using "Show" Commands" on page 4-4.

5 Working With Configuration Files

Commands and settings needed for the OmniSwitch can be contained in an ASCII-based configuration text file. Configuration files can be created in several ways and are useful in network environments where multiple switches must be managed and monitored.

This chapter describes how configuration files are created, how they are applied to the switch, and how they can be used to enhance OmniSwitch 10K and OmniSwitch 6900 usability.

In This Chapter

Configuration procedures described in this chapter include:

Configuration File Specifications

The following table lists specifications applicable to Configuration Files.

Platforms Supported	OmniSwitch 10K, 6900
Creation Methods for Configuration Files	• Create a text file on a word processor and upload it to the switch. • Invoke the switch's snapshot feature to create a text file. • Create a text file using the switch's text editor.
Timer Functions	Files can be applied immediately or by setting a timer on the switch.
Command Capture Feature	Snapshot feature captures switch configurations in a text file.
Error Reporting	Snapshot feature includes error reporting in the text file.
Text Editing on the Switch	Vi standard editor.
Default Error File Limit	1

Tutorial for Creating a Configuration File

This example creates a configuration file that includes CLI commands to configure the DHCP Relay application on the switch. For this example, the forward delay value is set to 15 seconds, the maximum number of hops is set to 3 and the IP address of the DHCP server is 128.251.16.52.

This tutorial shows you how to accomplish the following tasks:

1 Create a configuration text file containing CLI commands needed to configure DHCP Relay application.

This example used MS Notepad to create a text file on a PC workstation. The text file named **dhcp_relay.txt** contains three CLI commands needed to configure the forward delay value to 15 seconds and the maximum number of hops to 3. The IP address of the DHCP server is 128.251.16.52.

```
ip helper address 128.251.16.52
ip helper forward-delay 15
ip helper maximum-hops 3
```

2 Transfer the configuration file to the switch's file system.

For more information about transferring files onto the switch see Chapter 2, "Managing System Files."

3 Apply the configuration file to the switch by using the **configuration apply** command as shown here:

```
-> configuration apply dhcp_relay.txt
File configuration <dhcp_relay.txt>: completed with no errors
```

4 Use the **show configuration status** command to verify that the **dhcp_relay.txt** configuration file was applied to the switch. The display is similar to the one shown here:

```
-> show configuration status
File syntax check <text.txt>: completed with no errors

Error file limit: 1

Running configuration and saved configuration are different
```

For more information about these displays, refer to the *OmniSwitch CLI Reference Guide*.

5 Use the **show ip helper** command to verify that the DHCP Relay parameters defined in the configuration files were actually implemented on the switch. The display is similar to the one shown here:

```
-> show ip helper

Ip helper :

    Forward Delay(seconds)         = 15,
    Max number of hops             = 3,
    Relay Agent Information        = Disabled,
    PXE support                    = Disabled,
    Forward option                 = standard mode,
    Bootup Option                  = Disable
         Forwarding address list (Standard mode):
         192.168.10.10
```

These results confirm that the commands specified in the file **dhcp_relay.txt** configuration file were successfully applied to the switch.

Quick Steps for Applying Configuration Files

Setting a File for Immediate Application

In this example, the configuration file **configfile_1** exists on the switch in the **/flash** directory. When these steps are followed, the file will be immediately applied to the switch.

1 Verify that there are no timer sessions pending on the switch.

```
File configuration: none scheduled

Error file limit: 1
```

2 Apply the file by executing the **configuration apply** command, followed by the path and file name. If the configuration file is accepted with no errors, the CLI responds with a system prompt.

```
-> configuration apply /flash/configfile_1.txt
```

Note. Optional. You can specify *verbose mode* when applying a configuration file to the switch. When the keyword **verbose** is specified in the command line, all syntax contained in the configuration file is printed to the console. (When verbose is *not* specified in the command line, cursory information—number of errors and error log file name—will be printed to the console only if a syntax or configuration error is detected.)

To verify that the file was applied, enter the **show configuration status** command. The display is similar to the one shown here.

```
-> show configuration status
File configuration </flash/configfile_1.txt>: completed with 0 errors
```

For more information about this display, see "Configuration File Manager Commands" in the *OmniSwitch CLI Reference Guide*.

Setting an Application Session for a Date and Time

You can set a timed session to apply a configuration file at a specific date and time in the future. The following example applies the **bncom_cfg.txt** file at 9:00 a.m. on July 4 of the current year.

1 Verify that there are no current timer sessions pending on the switch.

```
-> show configuration status
File configuration: none scheduled

Error file limit: 1
```

2 Apply the file by executing the **configuration apply** using the **at** keyword with the relevant date and time.

```
-> configuration apply bncom_cfg.txt at 09:00 july 4
```

Note. Optional. To verify that the switch received this **configuration apply** request, enter the **show configuration status** command. The display is similar to the one shown here.

```
-> show configuration status
File configuration <bncom_cfg.txt>: scheduled at 07/04/10 09:00

Error file limit: 1

Running configuration and saved configuration are different
```

For more information about this display see "Configuration File Manager Commands" in the *OmniSwitch CLI Reference Guide*.

Setting an Application Session for a Specified Time Period

You can set a future timed session to apply a configuration file after a specified period of time has elapsed. In the following example, the **amzncom_cfg.txt** will be applied after 6 hours and 15 minutes have elapsed.

1 Verify that there are no current timer sessions pending on the switch.

```
-> show configuration status
File configuration: none scheduled
```

2 Apply the file by executing the **configuration apply** command using the in keyword with the relevant time frame specified.

```
-> configuration apply amzncom_cfg.txt in 6:15
```

Note. Optional. To verify that the switch received this **configuration apply** request, enter the **show configuration status** command. The display is similar to the one shown here.

```
-> show configuration status
File configuration </flash/working/amzncom_cfg.txt>: scheduled at 03/07/10 05:02
```

The "scheduled at" date and time show when the file will be applied. This value is 6 hours and 15 minutes from the date and time the command was issued.

For more information about this display see "Configuration File Manager Commands" in the *OmniSwitch CLI Reference Guide*.

Configuration Files Overview

Instead of using CLI commands entered at a workstation, you can configure the switch using an ASCII-based text file. You may type CLI commands directly into a text document to create a *configuration file* that will reside in your switch's **/flash** directory. Configuration files are created in the following ways:

- You may create, edit, and view a file using a standard text editor (such as MS WordPad or Notepad) on a workstation. The file can then be uploaded to the switch's **/flash** file directory.
- You can invoke the switch's CLI **configuration snapshot** command to capture the switch's current configuration into a text file. This causes a configuration file to be created in the switch's **/flash** directory.
- You can use the switch's text editor to create or edit a configuration file located in the switch's **/flash** file directory.

Applying Configuration Files to the Switch

Once you have a configuration file located in the switch's file system you must load the file into running memory to make it run on the switch. You do this by using **configuration apply** command.

You may apply configuration files to the switch immediately, or you can specify a timer session. In a timer session, you schedule a file to be applied in the future at a specific date and time or after a specific period of time has passed (like a countdown). Timer sessions are very useful for certain management tasks, especially synchronized batch updates.

- For information on applying a file immediately, refer to "Setting a File for Immediate Application" on page 5-4.
- For information on applying a file at a specified date and time, refer to "Setting an Application Session for a Date and Time" on page 5-4.
- For information on applying a file after a specified period of time has elapsed, refer to "Setting an Application Session for a Specified Time Period" on page 5-5.

Verifying a Timed Session

To verify that a timed session is running, use the **show configuration status** command. The following displays where the timed session was set using the **configuration apply qos_pol at 11:30 october 31** syntax.

```
-> show configuration status
File configuration <qos_pol>: scheduled at 11:30 october 31
```

Note. Only one session at a time can be scheduled on the switch. If two sessions are set, the last one will overwrite the first. Before you schedule a timed session you should use the **show configuration status** command to see if another session is already running.

The following displays where the timed session was set on March 10, 2002 at 01:00 using the **configuration apply group_config in 6:10** syntax.

```
-> show configuration status
File configuration <group_config>: scheduled at 03/10/02 07:10
```

Cancelling a Timed Session

You may cancel a pending timed session by using the **configuration cancel** command. To confirm that your timer session has been cancelled, use the **show configuration status** command. The following will display.

```
-> configuration cancel
-> show configuration status
File configuration: none scheduled
```

For more details about the CLI commands used to apply configuration files or to use timer sessions, refer to "Configuration File Manager Commands" in the *OmniSwitch CLI Reference Guide*.

Configuration File Error Reporting

If you apply a configuration file to the switch that contains significant errors, the application may not work. In this case, the switch will indicate the number of errors detected and print the errors into a text file that will appear in the **/flash** directory. The following display will result where the **cfg_txt** file contains three errors.

```
-> configuration apply cfg_file
Errors: 3
Log file name: cfg_txt.1.err
```

In this case, the error message indicates that the application attempt was unsuccessful. It also indicates that the switch wrote log messages into a file named **cfg_txt.1.err**, which now appears in your **/flash** directory. To view the contents of a generated error file, use the **view** command. For example, **view cfg_txt.1.err**.

Setting the Error File Limit

The number of files ending with the **.err** extension present in the switch's **/flash** directory is set with the **configuration error-file-limit** command. You can set the switch to allow a maximum number of error files in the **/flash** directory. Once the error file limit has been reached, the next error file generated will cause the error file with the oldest time stamp to be deleted. The following command sets the error file limit to 5 files:

```
-> configuration error-file limit 5
```

If you need to save files with the **.err** extension, you can either rename them so they no longer end with the **.err** extension or you may move them to another directory.

Syntax Checking

The configuration syntax check command is used to detect potential syntax errors contained in a configuration file *before* it is applied to the switch. It is recommended that you check *all* configuration files for syntax errors before applying them to your switch.

To run a syntax check on a configuration file, use the **configuration syntax-check** command. For example:

```
-> configuration syntax asc.1.snap
Errors: 3
Log file name: check asc.1.snap.1.err
```

In this example, the proposed **asc.1.snap** configuration file contains three errors. As with the **configuration apply** command, an error file (**.err**) is automatically generated by the switch whenever an error is detected. By default, this file is placed in the root **/flash** directory.

If a configuration file is located in another directory, be sure to specify the full path. For example:

```
-> configuration syntax check /flash/working/asc.1.snap
```

Viewing Generated Error File Contents

For error details, you can view the contents of a generated error file. To view the contents of an error file, use the **more** command. For example:

```
-> more asc.1.snap.1.err
```

For more information, refer to "Text Editing on the Switch" on page 5-8.

Verbose Mode Syntax Checking

When **verbose** is specified in the command line, all syntax contained in the configuration file is printed to the console, even if no error is detected. (When **verbose** is not specified in the command line, cursory information—number of errors and error log file name—will be printed to the console only if a syntax or configuration error is detected.)

To specify verbose mode, enter the **verbose** keyword at the end of the command line. For example:

```
-> configuration syntax check asc.1.snap verbose
```

Text Editing on the Switch

The switch software includes a standard line editor called "Vi". The Vi editor is available on most UNIX systems. No attempt is being made to document Vi in this manual because information on it is freely available on the Internet.

Invoke the "Vi" Editor

You can invoke the Vi editor from the command line. Use the following syntax to view the **switchlog.txt** file located in the **/flash/working** directory:

```
-> vi /flash/working switchlog.txt
```

Creating Snapshot Configuration Files

You can generate a list of configurations currently running on the switch by using the **configuration snapshot** command. A snapshot is a text file that lists commands issued to the switch during the current login session.

Note. A user must have read and write permission for the configuration family of commands to generate a snapshot file for those commands. See the "Switch Security" chapter of this manual for further information on permissions to specific command families.

Snapshot Feature List

You can specify the snapshot file so that it will capture the CLI commands for one or more switch features or for all network features. To generate a snapshot file for all network features, use the following syntax:

```
-> configuration snapshot all
```

To generate a snapshot file for specific features, select the appropriate syntax from the following list.

Snapshot Keywords		
WEBMGT	**QOS**	**IPSEC**
VRRP	**PORT-MAPPING**	**IPMS**
VLAN	**POLICY**	**IPMR**
VFC	**PMM**	**IP-ROUTING**
UDLD	**OSPF3**	**IP-HELPER**
SYSTEM	**OSPF**	**IP**
STP	**NTP**	**INTERFACE**
STACK-MANAGER	**NETSEC**	**HEALTH**
SNMP	**MULTI-CHASSIS**	**ERP**
SLB	**MODULE**	**CHASSIS**
SESSION	**LLDP**	**CAPABILITY**
RIPNG	**LINKAGG**	**BFD**
RIP	**BGP**	**AAA**
BRIDGE	**IPV6**	**ALL**

You may enter more than one network feature in the command line. Separate each feature with a space (and no comma). The following command will generate a snapshot file listing current configurations for the vlan, qos, and snmp command families.

```
-> configuration snapshot vlan qos snmp
```

User-Defined Naming Options

When the snapshot syntax does not include a file name, the snapshot file is created using the default file name asc.*n*.snap. Here, the *n* character holds the place of a number indicating the order in which the snapshot file name is generated. For example, the following syntax may generate a file named **asc.1.snap**.

```
-> configuration snapshot all
```

Subsequent snapshot files without a name specified in the command syntax will become **asc.2.snap**, **asc.3.snap**, etc.

The following command produces a snapshot file with the name **testfile.snap**.

```
-> configuration snapshot testfile.snap
```

Editing Snapshot Files

Snapshot files can be viewed, edited and reused as a configuration file. You also have the option of editing the snapshot file directly using the switch's Vi text editor or you may upload the snapshot file to a text editing software application on your workstation.

The snapshot file contains both command lines and comment lines. You can identify the comment lines because they each begin with the exclamation point (**!**) character. Comment lines are ignored by the switch when a snapshot file is being applied. Comment lines are located at the beginning of the snapshot file to form a sort of header. They also appear intermittently throughout the file to identify switch features or applications that apply to the commands that follow them.

Example Snapshot File Text

The following is the text of a sample snapshot file created with the **configuration snapshot all** command.

```
!=======================================!
! File: asc.1.snap                      !
!=======================================!
! Chassis :
system name OS10K
! Configuration:
! VLAN :
! IP :
ip service all
icmp unreachable net-unreachable disable
ip interface "vlan-1" address 10.255.211.70 mask 255.255.255.192 vlan 1 mtu 1500
ifindex 1
! IPMS :
! AAA :
aaa authentication default "local"
aaa authentication console "local"
! PARTM :
! AVLAN :
! 802.1x :
! QOS :
! Policy manager :
! Session manager :
! SNMP :
snmp security no security
snmp community map mode off
! IP route manager :
ip static-route 0.0.0.0 mask 0.0.0.0 gateway 10.255.211.65 metric 1
! RIP :
```

```
! OSPF :
! BGP :
! IP multicast :
! IPv6 :
! RIPng :
! Health monitor :
! Interface :
! Link Aggregate :
! VLAN AGG:
! 802.1Q :
! Spanning tree :
bridge mode 1x1
! Bridging :
source-learning chassis hardware
! Bridging :
! Port mirroring :
! UDP Relay :
! Server load balance :
! System service :
! VRRP :
! Web :
! Module :
! NTP :
! RDP :
```

This file shows configuration settings for the Chassis, IP, AAA, SNMP, IP route manager, Spanning tree, and Bridging services. Each of these services have configuration commands listed under their heading. All other switch services and applications are either not being using or are using default settings.

Verifying File Configuration

You can verify the content and the status of the switch's configuration files with commands listed in the following table.

show configuration status	Displays whether there is a pending timer session scheduled for a configuration file and indicates whether the running configuration and the saved configuration files are *identical* or *different*. This command also displays the number of error files that will be held in the flash directory.
show configuration snapshot	Generates a snapshot file of the switch's non-default current running configuration. A snapshot can be generated for all current network features or for one or more specific network features. A snapshot is a single text file that can be viewed, edited, and reused as a configuration file.
write terminal	Displays the switch's current running configuration for all features.

6 Managing Switch User Accounts

Switch user accounts may be set up locally on the switch for users to log into and manage the switch. The accounts specify login information (combinations of usernames and passwords) and privileges.

The switch has several interfaces (e.g. console, Telnet, HTTP, FTP) through which users may access the switch. The switch may be set up to allow or deny access through any of these interfaces. See Chapter 7, "Managing Switch Security," for information about setting up management interfaces.

In This Chapter

This chapter describes how to set up user accounts locally on the switch through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

This chapter provides an overview of user accounts. In addition, configuration procedures described in this chapter include:

User information may also be configured on external servers in addition to, or instead of, user accounts configured locally on the switch. For information about setting up external servers that are configured with user information, see the "Managing Authentication Servers" chapter in the *OmniSwitch AOS Release 7 Network Configuration Guide*.

User Database Specifications

Platforms Supported	OmniSwitch 10K, 6900
Maximum number of alphanumeric characters in a username	63
Maximum number of alphanumeric characters in a user password	30
Maximum number of local user accounts	50

User Account Defaults

- Two user accounts are available on the switch by default: **admin** and **default**. For more information about these accounts, see "Startup Defaults" on page 6-4 and "Default User Settings" on page 6-7.
- New users inherit the privileges of the **default** user if the specific privileges for the user are not configured; the default user is modifiable.
- Password defaults are as follows:

Description	Command	Default
Minimum password length	**user password-size min**	6 characters
Default password expiration for any user	**user password-expiration**	disabled
Password expiration for particular user	**expiration** keyword in the **user** command	none
Username is not allowed in password.	**user password-policy cannot-contain-username**	disabled
Minimum number of uppercase characters allowed in a password.	**user password-policy min-uppercase**	0 (disabled)
Minimum number of lowercase characters allowed in a password.	**user password-policy min-lowercase**	0 (disabled)
Minimum number of base-10 digits allowed in a password.	**user password-policy min-digit**	0 (disabled)
Minimum number of non-alphanumeric characters allowed in a password.	**user password-policy min-nonalpha**	0 (disabled)
Maximum number of old passwords to retain in the password history.	**user password-history**	4
Minimum number of days user is blocked from changing password.	**user password-min-age**	0 (disabled)

- Global user account lockout defaults are as follows:

Parameter Description	Command	Default
Length of time during which failed login attempts are counted.	**user lockout-window**	0—failed login attempts are never aged out.
Length of time a user account remains locked out of the switch before the account is automatically unlocked.	**user lockout-duration**	0—account remains locked until manually unlocked
Maximum number of failed login attempts allowed during the lockout window time period.	**user lockout-threshold**	0—no limit to the number of failed login attempts

Overview of User Accounts

A user account includes a login name, password, and user privileges. These privileges determine whether the user has read or write access to the switch and which command **domains** and command **families** the user is authorized to execute on the switch.

The designation of particular command families/domains or command families for user access is sometimes referred to as *partitioned management*. The privileges and profiles are sometimes referred to as *authorization*.

Note. For information about setting up user information on an authentication (AAA) server, see the "Managing Authentication Servers" chapter of the *OmniSwitch AOS Release 7 Network Configuration Guide*.

Users typically log into the switch through one of the following methods:

- **Console port**—A direct connection to the switch through the console port.
- **Telnet**—Any standard Telnet client may be used for logging into the switch.
- **FTP**—Any standard FTP client may be used for logging into the switch.
- **HTTP**—The switch has a Web browser management interface for users logging in via HTTP. This management tool is called WebView.
- **Secure Shell**—Any standard Secure Shell client may be used for logging into the switch.
- **SNMP**—Any standard SNMP browser may be used for logging into the switch.

Startup Defaults

By default, a single user management account is available at the first bootup of the switch. This account has the following user name and password:

- user name—**admin**
- password—**switch**

Initially, the **admin** user can only be authorized on the switch through the console port. Management access through any other interface is disabled. The Authenticated Switch Access commands may be used to enable access through other interfaces/services (Telnet, HTTP, etc.); however, SNMP access is not allowed for the admin user. Also, the admin user cannot be modified, except for the password.

Password expiration for the admin user is disabled by default. See "Configuring Password Expiration" on page 6-12.

In addition, another account, **default**, is available on the switch for default settings only; this account cannot be used to log into the switch. It is used to store and modify default settings for new users.

To set up a user account, use the **user** command, which specifies the following:

- *Password*—The password is required for new users or when modifying a user's SNMP access. The password will not appear in an ASCII configuration file created via the **snapshot** command.

- *Privileges*—The user's read and write access to command domains and families. See "Configuring Privileges for a User" on page 6-16 for more details.
- *SNMP access*—Whether or not the user is permitted to manage the switch via SNMP. See "Setting Up SNMP Access for a User Account" on page 6-17 for more details.

Typically, options for the user are configured at the same time the user is created. An example of creating a user and setting access privileges for the account is given here:

```
-> user thomas password techpubs read-write domain-policy
```

Quick Steps for Network Administrator User Accounts

1 Configure the user with the relevant username and password. For example, to create a user called **thomas** with a password of **techpubs**, enter the following:

```
-> user thomas password techpubs
```

For information about creating a user and setting up a password, see "Creating a User" on page 6-9.

2 Configure the user privileges (and SNMP access) if the user should have privileges that are different than those set up for the **default** user account. For example:

```
-> user thomas read-write domain-network ip-helper telnet
```

For information about the default user settings, see the next section. For information about setting up privileges, see "Configuring Privileges for a User" on page 6-16.

Note. *Optional*. To verify the user account, enter the **show user** command. The display is similar to the following:

```
-> show user thomas

User name = thomas,
Password expiration     = None,
Password allow to be modified date     = None,
Account lockout     = None,
Password bad attempts     = 0,
Read Only for domains   = None,
Read/Write for domains  = Network ,
Snmp allowed     = NO
Console-Only    = Disabled
```

For more information about the **show user** command, see the *OmniSwitch CLI Reference Guide*.

Default User Settings

The **default** user account on the switch is used for storing new user defaults for privileges and profile information. This account does not include a password and cannot be used to log into the switch.

At the first switch startup, the default user account is configured for:

- No read or write access.
- No SNMP access.

Any new users created on the switch will inherit the privileges of the default user unless the user is configured with specific privileges.

The default user settings may be modified. Enter the **user** command with **default** as the user name. Note that the default user may only store default functional privileges.

The following example modifies the **default** user account with **read-write** access to all CLI commands:

```
-> user default read-write all
```

In this example, any new user that is created will have read and write access to all CLI commands (unless a specific privilege or SNMP access is configured for the new user).

Account and Password Policy Settings

The switch includes global password settings that are used to implement and enforce password complexity when a password is created, modified, and used. These user-configurable settings apply the following password requirements to all user accounts configured for the switch:

- Minimum password size.
- Whether or not a password can contain the account username.
- Minimum password character requirements.
- Password expiration.
- Password history.
- Minimum password age.

In addition to global password settings, the switch also includes global user lockout settings that determine when a user account is locked out of the switch and the length of time the user account remains locked.

See “Configuring Password Policy Settings” on page 6-11 and “Configuring Global User Lockout Settings” on page 6-14 for more information.

How User Settings Are Saved

Unlike other settings on the switch, user settings configured through the **user** and **password** commands are saved to the switch configuration automatically. These settings are saved in real time in the local user database.

At bootup, the switch reads the database file for user information (rather than the **boot.cfg** file).

Note. Password settings configured through the **user password-policy** commands are not automatically saved to the switch configuration.

Creating a User

To create a new user, enter the **user** command with the desired username and password. Use the **password** keyword. For example:

```
-> user thomas password techpubs
```

In this example, a user account with a user name of **thomas** and a password of **techpubs** is stored in the local user database.

Note. The exclamation point (!) is not a valid password character. In addition, specifying an asterisk (*) as one or more characters in a password is allowed as long as every character is not an asterisk. For example, **password** **123456** is allowed; **password ********* is not allowed.

If privileges are not specified for the user, the user will inherit all of the privileges of the default user account. See "Default User Settings" on page 6-7.

Note that the password will not display in clear text in an ASCII configuration file produced by the **snapshot** command. Instead, it will display in encrypted form.

Removing a User

To remove a user from the local database, use the **no** form of the command:

```
-> no user thomas
```

The user account for **thomas** is removed from the local user database.

User-Configured Password

Users may change their own passwords by using the **password** command. In this example, the current user wants to change the password to **my_passwd**. Follow these steps to change the password:

1 Enter the **password** command. The system displays a prompt for the new password:

```
-> password
  enter old password:
```

2 Enter the old password. (The password is concealed with asterisks.) A prompt displays for the new password.

```
-> password
  enter old password:********
  enter new password:
```

3 Enter the desired password. The system then displays a prompt to verify the password.

```
-> password
  enter old password:********
  enter new password: *********
  reenter new password:
```

4 Enter the password again.

```
-> password
  enter old password:********
  enter new password: *********
  reenter new password: *********
->
```

The password is now reset for the current user. At the next switch login, the user must enter the new password.

Configuring Password Policy Settings

The global password policy settings for the switch define the following requirements that are applied to all user accounts:

- Minimum password size.
- Whether or not the password can contain the username.
- The minimum number of uppercase characters required in a password.
- The minimum number of uppercase characters required in a password.
- The minimum number of base-10 digits required in a password.
- The minimum number of non-alphanumeric characters (symbols) required in a password.
- Password expiration.
- The maximum number of old passwords that are saved in the password history.
- The minimum number of days during which a user is not allowed to change their password.

Password policy settings are applied when a password is created or modified. The following subsections describe how to configure these settings using CLI commands.

To view the current policy configuration, use the **show user password-policy** command. For more information about this command and those used in the configuration examples throughout this section, see the *OmniSwitch CLI Reference Guide*.

Setting a Minimum Password Size

To configure a minimum password size, enter the **user password-size min** command. For example:

```
-> user password-size min 10
```

The minimum length for any passwords configured for users is now 10 characters.

Configuring the Username Password Exception

Use the **user password-policy cannot-contain-username** command to block the ability to configure a password that contains the username. For example:

```
-> user password-policy cannot-contain-username enable
```

Enabling this functionality prevents the user from specifying the username in the password that is configured for the same user account. For example, the password for the account username of **public** can not contain the word **public** in any part of the password. However, the username of another account is still allowed.

Configuring Password Character Requirements

The character requirements specified in the global password policy determine the minimum number of uppercase, lowercase, non-alphanumeric, and 10-base digit characters required in all passwords. These requirements are configured using the following **user password-policy** commands:

Command	Configures ...
user password-policy min-uppercase	The minimum number of uppercase characters required in all passwords.
user password-policy min-lowercase	The minimum number of lowercase characters required in all passwords.
user password-policy min-digit	The minimum number of base-10 digits required in all passwords.
user password-policy min-nonalpha	The minimum number of non-alphanumeric characters (symbols) required in all passwords.

Specifying zero with any of the these commands disables the requirement. For example, if the number of minimum uppercase characters is set to zero (the default), then there is no requirement for a password to contain any uppercase characters.

Configuring Password Expiration

By default, password expiration is disabled on the switch. A global default password expiration may be specified for all users or password expiration may be set for an individual user.

Note. When the current user's password has less than one week before expiration, the switch will display an expiration warning after login.

If a user's password expires, the user will be unable to log into the switch through any interface; the **admin** user must reset the user's password. If the **admin** user's password expires, the admin user will have access to the switch through the console port with the currently configured password.

Default Password Expiration

To set password expiration globally, use the **user password-expiration** command with the desired number of days; the allowable range is 1 to 150 days. For example:

```
-> user password-expiration 3
```

The default password expiration is now set to three days. All user passwords on the switch will be set or reset with the three-day expiration. If an individual user was configured with a different expiration through the **user** command, the expiration will be reset to the global value.

The expiration is based on the switch system date/time and date/time the **user password-expiration** command is entered. For example, if a user is configured with a password expiration of 10 days, but the global setting is 20 days, that user's password will expire in 10 days.

To disable the default password expiration, use the **user password-expiration** command with the **disable** option:

```
-> user password-expiration disable
```

Specific User Password Expiration

To set password expiration for an individual user, use the **user** command with the expiration keyword and the desired number of days or an expiration date. For example:

```
-> user bert password techpubs expiration 5
```

This command gives user **bert** a password expiration of five days.

To set a specific date for password expiration, include the date in *mm/dd/yyyy hh:mm* format. For example:

```
-> user bert password techpubs expiration 02/19/2003 13:30
```

This command sets the password expiration to February 19, 2003, at 1:30pm; the switch will calculate the expiration based on the system date/time. The system date/time may be displayed through the **system date** and **system time** commands.

Note. The expiration will be reset to the global default setting (based on the **user password-expiration** command) if the user password is changed or the **user password-expiration** command is entered again.

Configuring the Password History

The password history refers to the number of old passwords for each user account that are saved by the switch. This functionality prevents the user from using the same password each time their account password is changed. For example, if the password history is set to 10 and a new password entered by the user matches any of the 10 passwords saved, then an error message is displayed notifying the user that the password is not available.

By default, the password history is set to save up to 4 old passwords for each user account. To configure the number of old passwords to save, use the **user password-history** command. For example:

```
-> user password-history 2
```

To disable the password history function, specify 0 as the number of old passwords to save. For example:

```
-> user password-history 0
```

Note that a password is dropped from the password history when it no longer falls within the number of passwords that are retained by the switch.

Configuring the Minimum Age for a Password

The password minimum age setting specifies the number of days during which a user is not allowed to change their password. Note that it is necessary to configure a password minimum age value that is less than the password expiration value.

The default minimum age is set to zero, which means that there is no minimum age requirement for a password. To configure a minimum password age, use the **user password-min-age** command. For example:

```
-> user password-min-age 7
```

This command specifies that the user is prevented from changing their password for seven days from the time the password was created or modified.

Configuring Global User Lockout Settings

The following user lockout settings configured for the switch apply to all user accounts:

- Lockout window—the length of time a failed login attempt is aged before it is no longer counted as a failed attempt.
- Lockout threshold—the number of failed login attempts allowed within a given lockout window period of time.
- Lockout duration—the length of time a user account remains locked until it is automatically unlocked.

In addition to the above lockout settings, the network administrator also has the ability to manually lock and unlock user accounts. The following subsections describe how to configure user lockout settings and how to manually lock and unlock user accounts.

Note. Only the **admin** user is allowed to configure user lockout settings. The **admin** account is protected from lockout; therefore, it is always available.

Lockout settings are saved *automatically*; that is, these settings do not require the **issu slot** command to save user settings over a reboot. To view the current lockout settings configured for the switch, use the **show user lockout-setting** command.

For more information about this command and those used in the configuration examples throughout this section, see the *OmniSwitch CLI Reference Guide*.

Configuring the User Lockout Window

The lockout window is basically a moving observation window of time in which failed login attempts are counted. If the number of failed login attempts exceeds the lockout threshold setting (see "Configuring the User Lockout Threshold Number" on page 6-14) during any given observation window period of time, the user account is locked out of the switch.

Note that if a failed login attempt ages beyond the observation window of time, that attempt is no longer counted towards the threshold number. For example, if the lockout window is set for 10 minutes and a failed login attempt occurred 11 minutes ago, then that attempt has aged beyond the lockout window time and is not counted. In addition, the failed login count is decremented when the failed attempt ages out.

If the lockout window is set to 0 this means that there is no observation window and failed login attempts are never aged out and will never be decremented. To configure the lockout window time, in minutes, use the **user lockout-window** command. For example:

```
-> user lockout-window 30
```

Do not configure an observation window time period that is greater than the lockout duration time period (see "Configuring the User Lockout Duration Time" on page 6-15).

Configuring the User Lockout Threshold Number

The lockout threshold number specifies the number of failed login attempts allowed during any given lockout window period of time (see "Configuring the User Lockout Window" on page 6-14). For example, if the lockout window is set for 30 minutes and the threshold number is set for 3 failed login attempts, then the user is locked out when 3 failed login attempts occur within a 30 minute time frame.

By default, the lockout threshold number is set to 0; this means that there is no limit to the number of failed login attempts allowed, even if a lockout window time period exists. To configure a lockout threshold number, use the **user lockout-threshold** command. For example:

```
-> user lockout-threshold 3
```

Note that a locked user account is automatically unlocked when the lockout duration time (see "Configuring the User Lockout Duration Time" on page 6-15) is reached or the **admin** user manually unlocks the user account.

Configuring the User Lockout Duration Time

The user lockout duration time specifies the number of minutes a user account remains locked until it is automatically unlocked by the switch. This period of time starts when the user account is locked out of the switch. Note that at any point during the lockout duration time, the **admin** user can still manually unlock the user account.

By default, the user lockout duration time is set to 0; this means that there is no automatic unlocking of a user account by the switch. The locked user account remains locked until it is manually unlocked by the **admin** user. To configure a lockout duration time, use the **user lockout-duration** command. For example:

```
-> user lockout-duration 60
```

Do not configure a lockout duration time that is less than the lockout window time period (see "Configuring the User Lockout Window" on page 6-14).

Manually Locking and Unlocking User Accounts

The **user lockout unlock** command is used to manually lock or unlock a user account. This command is only available to the **admin** user or a user who has read/write access privileges to the switch.

To lock a user account, enter **user lockout** and the username for the account. For example,

```
-> user j_smith lockout
```

To unlock a user account, enter **user unlock** and the username for the locked account. For example,

```
-> user j_smith unlock
```

In addition to this command, the **admin** user or users with read/write access privileges can change the user account password to unlock the account.

Note that if a lockout duration time (see "Configuring the User Lockout Duration Time" on page 6-15) is not configured for the switch, then it is only possible to manually unlock a user account with the **user lockout** command or by changing the user password.

Configuring Privileges for a User

To configure privileges for a user, enter the **user** command with the **read-only** or **read-write** option and the desired CLI command domain names or command family names. The **read-only** option provides access to **show** commands; the **read-write** option provides access to configuration commands and show commands. Command families are subsets of command domains.

If you create a user without specifying any privileges, the user's account will be configured with the privileges specified for the default user account.

Command domains and families are listed here:

Domain	Corresponding Families
domain-admin	file telnet debug
domain-system	system aip snmp rmon webmgt config
domain-physical	chassis module interface pmm health
domain-network	ip rip ospf bgp vrrp ip-routing ipx ipmr ipms rdp ospf3 ipv6
domain-layer2	vlan bridge stp 802.1q linkagg ip-helper
domain-service	dns
domain-policy	qos policy slb
domain-security	session avlan aaa

In addition to command families, the keywords **all** or **none** may be used to set privileges for all command families or no command families respectively.

An example of setting up user privileges:

```
-> user thomas read-write domain-network ip-helper telnet
```

User **thomas** will have write access to all the configuration commands and **show** commands in the network domain, as well as Telnet and IP helper (DHCP relay) commands. The user will not be able to execute any other commands on the switch.

Use the keyword **all** to specify access to all commands. In the following example, the user is given read access to all commands:

```
-> user lindy read-only all
```

Note. When modifying an existing user, the user password is not required. If you are configuring a new user with privileges, the password is required.

The default user privileges may also be modified. See "Default User Settings" on page 6-7.

Setting Up SNMP Access for a User Account

By default, users can access the switch based on the SNMP setting specified for the default user account. The **user** command, however, may be used to configure SNMP access for a particular user. SNMP access may be configured without authentication and encryption required (supported by SNMPv1, SNMPv2, or SNMPv3). Or it may be configured with authentication or authentication/encryption required (SNMPv3 only).

SNMP authentication specifies the algorithm that should be used for computing the SNMP authentication key. It may also specify DES encryption. The following options may be configured for a user's SNMP access with authentication or authentication/encryption:

- SHA—The SHA authentication algorithm is used for authenticating SNMP PDU for the user.
- MD5—The MD5 authentication algorithm is used for authenticating SNMP PDU for the user.
- SHA and DES—The SHA authentication algorithm and DES encryption standard is used for authenticating and encrypting SNMP PDU for the user.
- MD5 and DES—The MD5 authentication algorithm and the DES encryption standard is used for authenticating and encrypting SNMP PDU for the user.

The user's level of SNMP authentication is superseded by the SNMP version allowed globally on the switch. By default, the switch allows all SNMP requests. Use the **snmp security** command to change the SNMP security level on the switch.

Note. At least one user with SHA/MD5 authentication and/or DES encryption must be configured on the switch for SNMPv3 communication with OmniVista.

The community string carried in the SNMP PDU identifies the request as an SNMPv1 or SNMPv2 request. The way the community string is handled on the switch is determined by the setting of the **snmp community-map mode** command. If the community map mode is enabled, the community string is checked against the community strings database (populated by the **snmp community-map** command). If the community map mode is disabled, then the community string value is checked against the user database. In either case, if the check fails, the request is dropped.

For more information about configuring SNMP globally on the switch, see Chapter 9, "Using SNMP."

The next sections describe how to configure SNMP access for users. Note the following:

- SNMP access cannot be specified for the **admin** user.
- When modifying a user's SNMP access, the user password must be re-entered (or a new one configured). This is required because the hash algorithm used to save the password in the switch depends on the SNMP authentication level.

SNMP Access Without Authentication/Encryption

To give a user SNMP access without SNMP authentication required, enter the **user** command with the **no auth** option. For example, to give existing user **thomas** SNMP access without SNMP authentication, enter the following:

```
-> user thomas password techpubs no auth
```

For this user, if the SNMP community map mode is enabled (the default), the SNMP community map must include a mapping for this user to a community string. In this example, the community string is **our_group**:

```
-> snmp community map our_group user thomas
```

In addition, the global SNMP security level on the switch must allow non-authenticated SNMP frames through the switch. By default, the SNMP security level is **privacy all**; this is the highest level of SNMP security, which allows only SNMPv3 frames through the switch. Use the **snmp security** command to change the SNMP security level. For more information about configuring SNMP globally on the switch, see Chapter 9, "Using SNMP."

SNMP Access With Authentication/Encryption

To configure a user with SNMP access and authentication, enter the **user** command with the desired authentication type (**sha**, **md5**, **sha+des**, and **md5+des**).

```
-> user thomas password techpubs sha+des
```

When SNMP authentication is specified, an SNMP authentication key is computed from the user password based on the authentication/encryption setting. In this example, the switch would use the SHA authentication algorithm and DES encryption on the **techpubs** password to determine the SNMP authentication key for this user. The key is in hexadecimal form and is used for encryption/de-encryption of the SNMP PDU.

The authentication key is only displayed in an ASCII configuration file if the **snapshot** command is entered. The key is indicated in the file by the syntax **authkey** *key*. See Chapter 5, "Working With Configuration Files," for information about using the **snapshot** command. The key is not displayed in the CLI.

Removing SNMP Access From a User

To deny SNMP access, enter the **user** command with the **no snmp** option:

```
-> user thomas no snmp
```

This command results in **thomas** no longer having SNMP access to manage the switch.

Multiple User Sessions

Several CLI commands give you information about user sessions that are currently operating on the OmniSwitch, including your own session. These commands allow you to list the number and types of sessions that are currently running on the switch. You can also terminate another session, provided you have administrative privileges.

Listing Other User Sessions

The **who** command displays all users currently logged into the OmniSwitch. The following example shows use of the **who** command and a resulting display:

```
-> who
Session number = 0
  User name   = (at login),
  Access type = console,
  Access port = Local,
  IP address  = 0.0.0.0,
  Read-only domains   = None,
  Read-only families  = ,
  Read-Write domains  = None,
  Read-Write families = ,
Session number = 1
  User name   = admin,
  Access type = http,
  Access port = Ethernet,
  IP address  = 123.251.12.51,
  Read-only domains   = None,
  Read-only families  = ,
  Read-Write domains  = All ,
  Read-Write families = ,
Session number = 3
  User name   = admin,
  Access type = telnet,
  Access port = Ethernet,
  IP address  = 123.251.12.61,
  Read-only domains   = None,
  Read-only families  = ,
  Read-Write domains  = All ,
  Read-Write families = ,
```

The above display indicates that three sessions are currently active on the OmniSwitch. Session number 0 always shows the console port whenever that port is active and logged in. The other sessions are identified by session number, user name, the type of access, port type, IP address, and user privileges.

Listing Your Current Login Session

In order to list information about your current login session, you may either use the **who** command and identify your login by your IP address or you may enter the **whoami** command. The following will display:

```
-> whoami
Session number = 4
  User name   = admin,
  Access type = telnet,
  Access port = NI,
  IP address  = 148.211.11.02,
  Read-only domains   = None,
  Read-only families  = ,
  Read-Write domains  = All ,
  Read-Write families = ,
```

This display indicates that the user is currently logged in as session number 4, under the username "admin," using a Telnet interface, from the IP address of 148.211.11.02.

Terminating Another Session

If you are logged in with administrative privileges, you can terminate the session of another user by using the **kill** command. The following command will terminate login session number 4.

```
-> kill 4
```

The command syntax requires you to specify the number of the session you want to kill. You can use the **who** command for a list of all current user sessions and their numbers. The **kill** command takes effect immediately.

Verifying the User Configuration

To display information about user accounts configured locally in the user database, use the **show** commands listed here:

show user	Displays information about all users or a particular user configured in the local user database on the switch.
show user password-policy	Displays the minimum number of characters that are required for a user password.
show user password-policy	Displays the expiration date for passwords configured for user accounts stored on the switch.
show user password-policy	Displays the global password settings configured for the switch.
show user lockout-setting	Displays the global user lockout settings configured for the switch.
show aaa priv hexa	Displays hexadecimal values for command domains/families.

For more information about the resulting displays from these commands, see the *OmniSwitch CLI Reference Guide*. An example of the output for the **show user** command is also given in "Quick Steps for Network Administrator User Accounts" on page 6-6.

7 Managing Switch Security

Switch security is provided on the switch for all available management interfaces. The switch may be set up to allow or deny access through any of these interfaces.

In This Chapter

This chapter describes how to set up switch management interfaces through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

An overview of switch security is given in this chapter. In addition, configuration procedures described in this chapter include:

A user login procedure requires that users are authenticated for switch access via an external authentication server or the local user database. For information about setting up user accounts locally on the switch, see Chapter 6, "Managing Switch User Accounts." For information about setting up external servers that are configured with user information, see the "Managing Authentication Servers" chapter in the *Network Configuration Guide*.

This chapter describes how to enable/disable access for management interfaces. For information about basic login on the switch, see Chapter 1, "Logging Into the Switch."

Switch Security Defaults

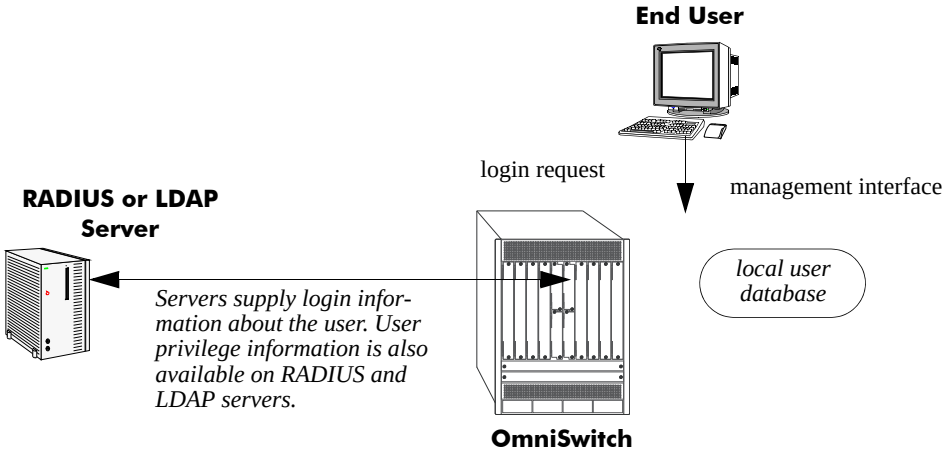
Access to managing the switch is always available for the **admin** user through the console port, even if management access to the console port is disabled for other users.

Description	Command	Default
Console Access	**aaa authentication**	Enabled
Remote Access	**aaa authentication**	Disabled

Switch Security Overview

Switch security features increase the security of the basic switch login process by allowing management only through particular interfaces for users with particular privileges. Login information and privileges may be stored on the switch and/or an external server, depending on the type of external server you are using and how you configure switch access.

The illustration here shows the components of switch security:



Authenticated Switch Access Setup

An external RADIUS or LDAP server can supply both user login and authorization information. External servers may also be used for accounting, which includes logging statistics about user sessions. For information about configuring the switch to communicate with external servers, see the "Managing Authentication Servers" chapter in the *Network Configuration Guide*.

If an external server is not available or is not configured, user login information and user authorization may be provided through the local user database on the switch. The user database is described in Chapter 6, "Managing Switch User Accounts."

Logging may also be accomplished directly on the switch. For information about configuring local logging for switch access, see "Configuring Accounting for ASA" on page 7-11. For complete details about local logging, see the "Using Switch Logging" chapter in the *Network Configuration Guide*.

Authenticated Switch Access

Authenticated Switch Access (ASA) is a way of authenticating users who want to manage the switch. With authenticated access, all switch login attempts require authentication via the local user database or via a third-party server.

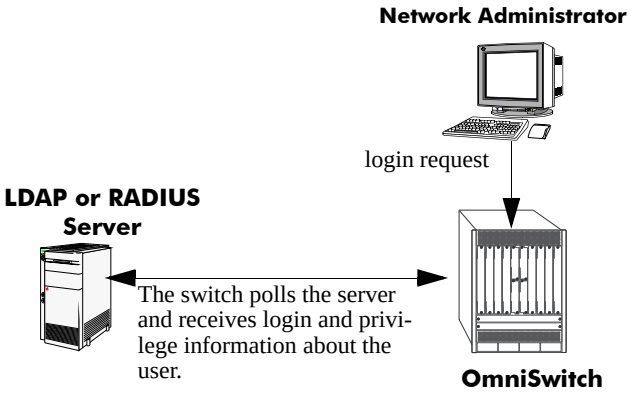
This section describes how to configure management interfaces for authenticated access as well as how to specify external servers that the switch can poll for login information. The type of server may be an authentication-only mechanism or an authentication, authorization, and accounting (AAA) mechanism.

AAA Servers—RADIUS or LDAP

AAA servers are able to provide authorization for switch management users as well as authentication (they also may be used for accounting). The AAA servers supported on the switch are Remote Authentication Dial-In User Service (RADIUS) or Lightweight Directory Access Protocol (LDAP) servers. User login information and user privileges may be stored on the servers.

Privileges are used for *network administrator accounts*. Instead of user privileges an end-user profile may be associated with a user for *customer login accounts*. User information configured on an external server may include a profile name attribute. The switch will attempt to match the profile name to a profile stored locally on the switch.

The following illustration shows the two different user types attempting to authenticate with a AAA server:



AAA Server (LDAP or RADIUS)

For more information about types of users, see Chapter 6, “Managing Switch User Accounts.”

Interaction With the User Database

By default, switch management users may be authenticated through the console port via the local user database. If external servers are configured for other management interfaces (such as Telnet, or HTTP), but the servers become unavailable, the switch will poll the local user database for login information.

Access to the console port provides secure failover in case of misconfiguration or if external authentication servers become unavailable. The **admin** user is always authorized through the console port via the local database (provided the correct password is supplied), even if access to the console port is disabled.

The database includes information about whether or not a user is able to log into the switch and which kinds of privileges or rights the user has for managing the switch. The database may be set up by the **admin** user or any user with write privileges to the AAA commands.

See Chapter 6, "Managing Switch User Accounts," for more information about setting up the user database.

Configuring Authenticated Switch Access

Setting up Authenticated Switch Access involves the following general steps:

1 Set Up the Authentication Servers. This procedure is described briefly in this chapter. See the "Managing Authentication Servers" chapter of the *Network Configuration Guide* for complete details.

2 Set Up the Local User Database. Set up user information on the switch if user login or privilege information will be pulled from the switch. See Chapter 6, "Managing Switch User Accounts."

3 Set Up the Management Interfaces. This procedure is described in "Setting Up Management Interfaces for ASA" on page 7-9.

4 Set Up Accounting. This step is optional and is described in "Configuring Accounting for ASA" on page 7-11.

Additional configuration is required in order to set up the switch to communicate with external authentication servers. This configuration is briefly mentioned in this chapter and described in detail in the "Managing Authentication Servers" chapter of the *Network Configuration Guide*.

If you are using the local switch database to authenticate users, user accounts must be set up on the switch. Procedures for creating user accounts are described in this chapter. See Chapter 6, "Managing Switch User Accounts."

Note that by default:

- Authenticated switch access is available only through the console port.
- Users are authenticated through the console port via the local user database on the switch.

These defaults provide "out-of-the-box" security at initial startup. Other management interfaces (Telnet, HTTP, etc.) must be specifically enabled before they can access the switch.

A summary of the commands used for configuring ASA is given in the following table:

Commands	Used for...
user	Configuring the local user database on the switch.
aaa radius-server **aaa tacacs+-server**	Setting up the switch to communicate with external RADIUS or LDAP authentication servers.
aaa authentication	Configuring the management interface and specifying the servers and/or local user database to be used for the interface.
aaa accounting session	*Optional*. Specifies servers to be used for accounting.

Quick Steps for Setting Up ASA

1 If the local user database is used for user login information, set up user accounts through the **user** command. In this example, user privileges are configured:

```
-> user thomas password mypassword read-write all
```

2 If an external RADIUS or LDAP server is used for user login information, use the **aaa radius-server** or **aaa tacacs+-server** commands to configure the switch to communicate with these servers. For example:

```
-> aaa radius-server rad1 host 10.10.1.2 timeout 3
```

For more information, see the "Managing Authentication Servers" chapter in the *Network Configuration Guide*.

3 Use the **aaa authentication** command to specify the management interface through which switch access is permitted (such as **console**, **telnet**, **ftp**, **http**, or **ssh**). Specify the server and backup servers to be used for checking user login and privilege information. Multiple servers of different types may be specified. For example:

```
-> aaa authentication telnet rad1 ldap2 local
```

The order of the server names is important. The switch uses the first available server in the list. In this example, the switch would use **rad1** to authenticate Telnet users. If **rad1** becomes unavailable, the switch will use **ldap2**. If **ldap2** then becomes unavailable, the switch will use the local user database to authenticate users.

4 Repeat step 3 for each management interface to which you want to configure access; or use the **default** keyword to specify access for all interfaces for which access is not specifically denied. For example, if you want to configure access for all management interfaces except HTTP, you would enter:

```
-> no aaa authentication http
-> aaa authentication default rad1 local
```

Note the following:

- SNMP access may only use LDAP servers or the local user database. If you configure the default management access with only RADIUS SNMP will not be enabled.
- It is recommended that Telnet and FTP be disabled if Secure Shell (**ssh**) is enabled.
- If you want to use WebView to manage the switch, make sure HTTP is enabled.

5 Specify an accounting server if a RADIUS or LDAP server will be used for accounting. Specify **local** if accounting may be done on the switch through the Switch Logging feature. Multiple servers may be specified as backups.

```
-> aaa accounting session ldap2 local
```

The order of the server names is important here as well. In this example, the switch will use **ldap2** for logging switch access sessions. If **ldap2** becomes unavailable, the switch will use the local Switch Logging facility. For more information about Switch Logging, see the *Network Configuration Guide*.

Note. To verify the switch access setup, enter the **show aaa authentication** command. The display is similar to the one shown here:

```
Service type = Default
   1rst authentication server   = rad1
   2nd authentication server    = local
Service type = Console
   Authentication = Use Default,
   1rst authentication server   = rad1
   2nd authentication server    = local
Service type = Telnet
   Authentication = Use Default,
   1rst authentication server   = rad1
   2nd authentication server    = local
Service type = Ftp
   Authentication = Use Default,
   1rst authentication server   = rad1
   2nd authentication server    = local
Service type = Http
   Authentication = denied
Service type = Snmp
   Authentication = Use Default,
   1rst authentication server   = rad1
   2nd authentication server    = local
Service type = Ssh
   Authentication = Use Default,
   1rst authentication server   = rad1
   2nd authentication server    = local
```

For more information about this command, see the *OmniSwitch CLI Reference Guide*.

Setting Up Management Interfaces for ASA

By default, authenticated access is available through the console port. Access through other management interfaces is disabled. This chapter describes how to set up access for management interfaces. For more details about particular management interfaces and how they are used, see Chapter 1, "Logging Into the Switch."

To give switch access to management interfaces, use the **aaa authentication** command to allow or deny access to each interface type; the **default** keyword may be used to configure access for all interface types. Specify the server(s) to be used for authentication through the indicated management interface.

To specify an external authentication server or servers, use the RADIUS or LDAP server name. To specify that the local user database should be used for authentication, use the **local** keyword.

RADIUS and LDAP servers are set up to communicate with the switch via the **aaa radius-server** and **aaa ldap-server** commands. For more information about configuring the switch to communicate with these servers, see the "Managing Authentication Servers" chapter of the *Network Configuration Guide*.

The order of the specified servers is important. The switch uses only one server for authentication—the first available server in the list. All authentication attempts will be tried on that server. Other servers are not tried, even if they are available. If **local** is specified, it must be last in the list since the local user database is always available when the switch is up.

Servers may also be used for accounting, or logging, of authenticated sessions. See "Configuring Accounting for ASA" on page 7-11.

The following table describes the management access interfaces or methods and the types of authentication servers that may be used with them:

Server Type	Management Access Method
RADIUS	Telnet, FTP, HTTP, SSH
LDAP	Telnet, FTP, HTTP, SSH, SNMP
local	console, FTP, HTTP, SSH, SNMP

Enabling Switch Access

Enter the **aaa authentication** command with the relevant keyword that indicates the management interface and specify the servers to be used for authentication. In this example, Telnet access for switch management is enabled. Telnet users will be authenticated through a chain of servers that includes a RADIUS server and an LDAP server that have already been configured through the **aaa radius-server** and **aaa ldap-server** commands respectively. For example:

```
-> aaa authentication telnet rad1 ldap2 local
```

After this command is entered, Telnet users will be authenticated to manage the switch through the **rad1** RADIUS server. If that server is unavailable, the LDAP server, **ldap2**, will be polled for user information. If that server is unavailable, the local user database will be polled for user information. Note that if the local user database is specified, it must be last in the list of servers.

To disable authenticated access for a management interface use the **no** form of the command with the keyword for the interface. For example:

```
-> no aaa authentication ftp
```

FTP access is now denied on the switch.

Note. The **admin** user always has switch access through the console port even if access is denied through the console port.

To remove a server from the authenticated switch access configuration, enter the **aaa authentication** command with the relevant server names (s) and leave out the names of any servers you want to remove. For example:

```
-> aaa authentication telnet rad1 local
```

The server **ldap2** is removed for Telnet access and will not be polled for user information when users attempt to log into the switch through Telnet.

Note. SNMP can only use LDAP servers or the local user database for authentication.

Configuring the Default Setting

The **default** keyword may be used to specify the default setting for all management interfaces except those that have been explicitly denied. For example:

```
-> no aaa authentication ftp
-> aaa authentication default ldap2 local
```

In this example, all management interfaces except FTP are given switch access through **ldap2** and the local user database.

The **default** keyword may also be used to reset a specified interface to the default interface setting. For example:

```
-> aaa authentication ftp default
```

In this example, FTP users will now be authenticated through the servers that are specified for the default interface.

Configuring Accounting for ASA

Accounting servers track network resources such as time, packets, bytes, etc., and user activity (when a user logs in and out, how many login attempts were made, session length, etc.). The accounting servers may be located anywhere in the network.

Note the following:

- The servers may be different types.
- The keyword **local** must be specified if you want accounting to be performed via the Switch Logging feature in the switch. If **local** is specified, it must be the last server in the list.

Note that external accounting servers are configured through the **aaa radius-server** and **aaa tacacs+-server** commands. These commands are described in "Managing Authentication Servers" in the *Network Configuration Guide*.

To enable accounting (logging a user session) for Authenticated Switch Access, use the **aaa accounting session** command with the relevant server name(s). In this example, the RADIUS and LDAP servers have already been configured through the **aaa radius-server** and **aaa ldap-server** commands.

```
-> aaa accounting session rad1 ldap2 local
```

After this command is entered, accounting will be performed through the **rad1** RADIUS server. If that server is unavailable, the LDAP server, **ldap2**, will be used for accounting. If that server is unavailable, logging will be done locally on the switch through the Switch Logging feature. (For more information about Switch Logging, see the *Network Configuration Guide*.)

To remove an individual server from the list of servers, enter the **aaa accounting session** command with the relevant server name(s), removing the desired server from the list. For example:

```
-> aaa accounting session rad1 local
```

The server **ldap2** is removed as an accounting server.

To disable accounting for Authenticated Switch Access, use the **no** form of the **aaa accounting session** command:

```
-> no aaa accounting session
```

Accounting will not be performed for Authenticated Switch Access sessions.

Verifying the ASA Configuration

To display information about management interfaces used for Authenticated Switch Access, use the **show** commands listed here:

show aaa authentication	Displays information about the current authenticated switch session.
show aaa accounting	Displays information about accounting servers configured for Authenticated Switch Access or Authenticated VLANs.
show aaa server	Displays information about a particular AAA server or AAA servers.

For more information about the resulting displays from these commands, see the *OmniSwitch CLI Reference Guide*. An example of the output for the **show aaa authentication** command is also given in "Quick Steps for Setting Up ASA" on page 7-7.

8 Using WebView

The switch can be monitored and configured using WebView, Alcatel-Lucent's web-based device management tool. The WebView application is embedded in the switch and is accessible via the following web browsers:

In This Chapter

This chapter provides an overview of WebView and WebView functionality, and includes information about the following procedures:

- WebView CLI (see "WebView CLI Defaults" on page 8-2)
- WebView Quick Steps (see "WebView Page Layout" on page 8-4)

WebView CLI Defaults

Web Management Command Line Interface (CLI) commands allow you to enable/disable WebView, enable/disable Secure Socket Layer (SSL), and view basic WebView parameters. These configuration options are also available in WebView. The following table lists the defaults for WebView configuration.

Description	Command	Default
WebView Server	**webview server**	enabled
WebView Access	**webview access**	enabled
Force SSL	**webview force-ssl**	enabled
HTTPS port	**webview https-port**	443
HTTP port	**webview http-port**	80

Browser Setup

Your browser preferences (or options) should be set up as follows:

- Cookies should be enabled. Typically this is the default.
- JavaScript must be enabled/supported.
- Java must be enabled.
- Style sheets must be enabled; that is, the colors, fonts, backgrounds, etc. of web pages should always be used (rather than any user-configured settings).
- Checking for new versions of pages should be set to “Every time” when your browser opens.
- If you are using a proxy server, the proxy settings should be configured to bypass the switch on which you are running WebView (the local switch).

Typically many of these settings are configured as the default. Different browsers (and different versions of the same browser) may have different dialogs for these settings. Check your browser help pages if you need help.

WebView CLI Commands

The following configuration options can be performed using the CLI. These configuration options are also available in WebView; but changing the web server port or secured port may only be done through the CLI (or SNMP).

Enabling/Disabling WebView

WebView is enabled on the switch by default. If necessary, use the **webview server** and **webview access** commands to enable/disable WebView. For example:

```
-> webview server enable
-> webview access enable
```

If web management is disabled, you will not be able to access the switch using WebView. Use the **show webview** command to view WebView status.

Changing the HTTP Port

You can change the port using the **webview http-port** command.

Note. All WebView sessions must be terminated before the switch will accept the command.

For example:

```
-> webview http-port 20000
```

To restore an HTTP port to its default value, use the **default** keyword as shown below:

```
-> webview http-port default
```

Enabling/Disabling SSL

Use the **webview force-ssl** command to enable Force SSL on the switch. For example:

```
-> webview force-ssl
```

Changing the HTTPS Port

You can change the port using the **webview https-port** command.

Note. All WebView sessions must be terminated before the switch accepts the command.

For example:

```
-> webview https-port 20000
```

To restore an HTTPS port to its default value, use the **default** keyword as shown below:

```
-> webview https-port default
```

Quick Steps for Setting Up WebView

1 Make sure you have an Ethernet connection to the switch.

2 Configure switch management for HTTP using the **aaa authentication** command. Enter the command, the port type that you are authenticating (**http**), and the name of an external or local server that is being used for authentication. For example, to configure switch management for HTTP using the "local" authentication server you would enter:

```
-> aaa authentication http local
```

3 Open a web browser.

4 Enter the IP address of the switch you want to access in the Address field of the browser and press Enter. The WebView login screen appears.

5 Enter the appropriate user ID and password (the initial user name is **admin** and the initial password is **switch**). After successful login, the Chassis Management Home Page appears

---

Note. The WebView self-signed certificate will generate a certificate warning on the browser.

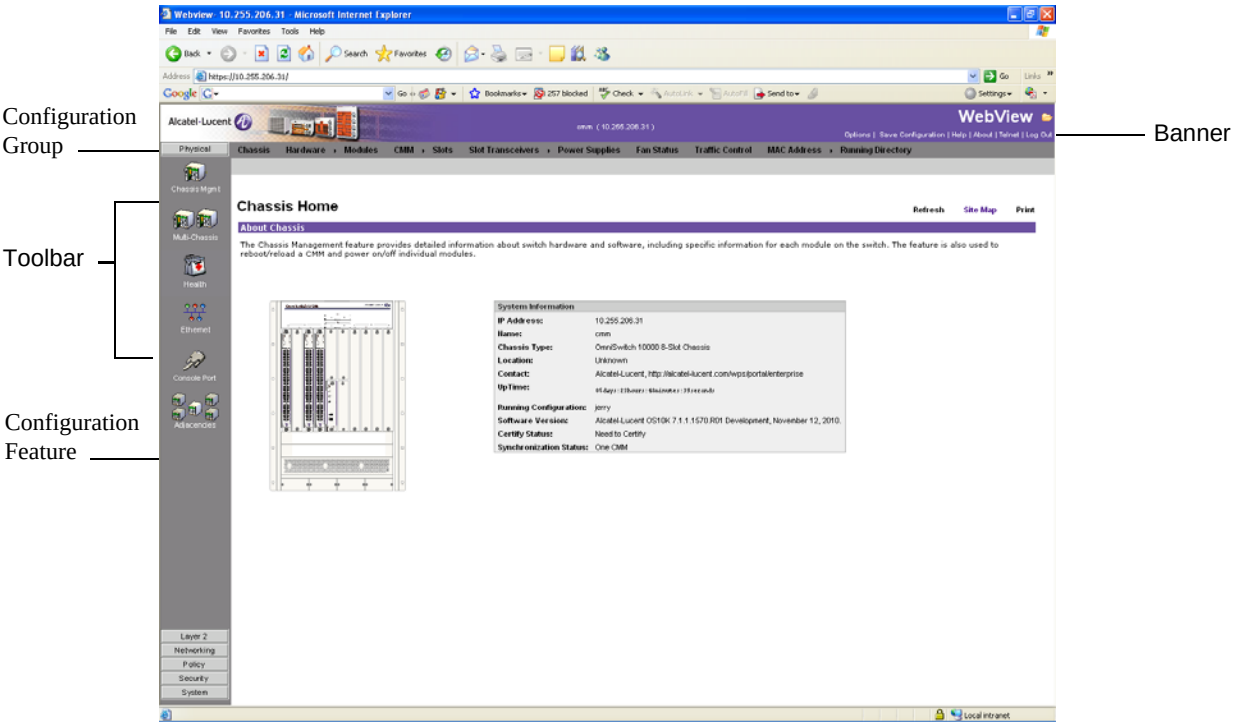
WebView Overview

The following sections provide an overview of WebView page layouts.

WebView Page Layout

As shown below, each WebView page is divided into four areas:

- **Banner**—Used to access global options (e.g., global help, telnet, and log out). An icon is also displayed in this area to indicate the current directory.
- **Toolbar**—Used to access WebView features.
- **Feature Options**—Used to access specific configuration options for each feature (displayed in drop-down menus at the top of the page).
- **View/Configuration Area**—Used to view/configure a feature.



WebView Chasis Home Page

Banner

The banner provides quick access to common tasks such as setting options, saving the switch configuration and using telnet to access the switch.

Toolbar

Switch configuration is divided into configuration groups in the toolbar (for example, Physical, Layer 2, etc.). Under each configuration group are switch features, identified by a name and an icon.

Feature Options

Feature configuration options are displayed as drop-down menus at the top of each feature page.

View/Configuration Area

The View/Configuration area is where switch configuration information is displayed and where configuration pages appear. After logging into WebView, a real-time graphical representation of the switch displays all of the switch's current components. The feature configuration options on this page are used to configure the switch.

9 Using SNMP

The Simple Network Management Protocol (SNMP) is an application-layer protocol that allows communication between SNMP managers and SNMP agents on an IPv4 as well as on an IPv6 network. Network administrators use SNMP to monitor network performance and to manage network resources.

In This Chapter

This chapter describes SNMP and how to use it through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

This chapter also includes lists of Industry Standard and Enterprise (Proprietary) MIBs used to manage the OmniSwitch.

SNMP Specifications

The following table lists specifications for the SNMP protocol.

Platforms Supported	OmniSwitch 10K, 6900
RFCs Supported for SNMPv2	1902 through 1907 - SNMPv2c Management Framework 1908 - Coexistence and transitions relating to SNMPv1 and SNMPv2c
RFCs Supported for SNMPv3	2570 – Version 3 of the Internet Standard Network Management Framework 2571 – Architecture for Describing SNMP Management Frameworks 2572 – Message Processing and Dispatching for SNMP 2573 – SNMPv3 Applications 2574 – User-based Security Model (USM) for version 3 SNMP 2575 – View-based Access Control Model (VACM) for SNMP 2576 – Coexistence between SNMP versions
Platforms Supported	OmniSwitch 10K, 6900
SNMPv1, SNMPv2, SNMPv3	The SNMPv3 protocol is ascending compatible with SNMPv1 and v2 and supports all the SNMPv1 and SNMPv2 PDUs
SNMPv1 and SNMPv2 Authentication	Community Strings
SNMPv1, SNMPv2 Encryption	None
SNMPv1 and SNMPv2 Security requests accepted by the switch	Sets and Gets
SNMPv3 Authentication	SHA, MD5
SNMPv3 Encryption	DES
SNMPv3 Security requests accepted by the switch.	Non-authenticated Sets, Non-authenticated Gets and Get-Nexts, Authenticated Sets, Authenticated Gets and Get-Nexts, Encrypted Sets, Encrypted Gets and Get-Nexts
SNMP traps	Refer to the table on page 9-10 for a complete list of traps and their definitions.

SNMP Defaults

The following table describes the default values of the SNMP protocol parameters.

Parameter Description	Command	Default Value/Comments
SNMP Management Station	**snmp station**	UDP port 162, SNMPv3, Enabled
Community Strings	**snmp community-map**	Enabled
SNMP Security setting	**snmp security**	Privacy all (highest) security
Trap filtering	**snmp-trap filter-ip**	Disabled
Trap Absorption	**snmp-trap absorption**	Enabled
Enables the forwarding of traps to WebView.	**snmp-trap to-webview**	Enabled

Parameter Description	Command	Default Value/Comments
Enables or disables SNMP authentication failure trap forwarding.	**snmp authentication-trap**	Disabled

Quick Steps for Setting Up An SNMP Management Station

An SNMP Network Management Station (NMS) is a workstation configured to receive SNMP traps from the switch. To set up an SNMP NMS by using the switch's CLI, proceed as follows:

1 Specify the user account name and the authentication type for that user. For example:

```
-> user NMSuserV3MD5DES md5+des password ********
```

2 Specify the UDP destination port number (in this case 8010), the IP address of the management station (199.199.100.200), a user account name (NMSuserV3MD5DES), and the SNMP version number (v3). For example:

Note: The user account must already be created as documented in Step 1 above.

```
-> snmp station 199.199.100.200 8010 NMSuserV3MD5DES v3 enable
```

Use the same command as above for specifying the IPv6 address of the management station. For example:

```
-> snmp station 300::1 enable
```

Note. *Optional.* To verify the SNMP Management Station, enter the **show snmp station** command. The display is similar to the one shown here:

```
-> show snmp station
ipAddress/udpPort           status    protocol user
----------------------------+---------+--------+------------------------------
199.199.100.200/8010        enable    v3       NMSuserV3MD5DES
199.199.101.201/111         disable   v2       NMSuserV3MD5
199.199.102.202/8002        enable    v1       NMSuserV3SHADES


-> show snmp station
ipAddress/udpPort                                    status    protocol user
-----------------------------------------------------+---------+--------+------
172.21.160.32/4000                                   enable    v3       abc
172.21.160.12/5000                                   enable    v3       user1
0300:0000:0000:0000:0211:d8ff:fe47:470b/4001         enable    v3       user2
0300:0000:0000:0000:0211:d8ff:fe47:470c/5001         enable    v2       abc
```

For more information about this display, see the "SNMP Commands" chapter in the *OmniSwitch CLI Reference Guide.*

Quick Steps for Setting Up Trap Filters

You can filter traps by limiting user access to trap command families. You can also filter according to individual traps.

Filtering by Trap Families

The following example will create a new user account. This account will be granted read-only privileges to three CLI command families (snmp, chassis, and interface). Read-only privileges will be withheld from all other command families.

1 Set up a user account named “usermark2” by executing the **user** CLI command.

```
-> user usermark2 password ******
```

2 Remove all read-only privileges from the user account.

```
-> user usermark2 read-only none
```

3 Add read-only privileges for the snmp, chassis, and interface command families.

```
-> user usermark2 read-only snmp chassis interface
```

Note. *Optional.* To verify the user account, enter the **show user** command. A partial display is shown here:

```
-> show user
  User name = usermark2
  Read right          = 0x0000a200 0x00000000,
  Write right         = 0x00000000 0x00000000,
  Read for domains    = ,
  Read for families   = snmp chassis interface ,
  Write for domains   = None ,
  Snmp authentication = NONE, Snmp encryption = NONE
```

The usermark2 account has read-only privileges for the snmp, chassis, and interface command families.

---

4 Set up an SNMP station with the user account “usermark2” defined above.

```
-> snmp station 210.1.2.1 usermark2 v3 enable
```

Note. *Optional.*To verify the SNMP Management Station, enter the **show snmp station** command. The display is similar to the one shown here:

```
-> show snmp station
ipAddress/udpPort           status   protocol  user
----------------------------+--------+--------+------------------------------
210.1.2.1/162               enable   v3        usermark2
```

The usermark2 account is established on the SNMP station at IP address 210.1.2.1.

Filtering by Individual Traps

The following example enables trap filtering for the coldstart, warmstart, linkup, and linkdown traps. The identification numbers for these traps are 0, 1, 2, and 3. When trap filtering is enabled, these traps will be filtered. This means that the switch will *not* pass them through to the SNMP management station. All other traps will be passed through.

1 Specify the IP address for the SNMP management station and the trap identification numbers.

```
-> show snmp trap filter 210.1.2.1 0 1 2 3
-> snmp trap filter 300::1 1 3 4
```

Note. *Optional.* You can verify which traps will *not* pass through the filter by entering the **snmp-trap filter-ip** command. The display is similar to the one shown here:

```
-> show snmp trap filter
ipAddress         trapId list
-----------------+--------------------------------------
210.1.2.1          0   1   2   3
```

The SNMP management station with the IP address of 210.1.2.1 will *not* receive trap numbers 0, 1, 2, and 3.

For trap numbers refer to the "Using SNMP For Switch Security" on page 9-10. For more information on the CLI commands and the displays in these examples, refer to the *OmniSwitch CLI Reference Guide.*

SNMP Overview

SNMP provides an industry standard communications model used by network administrators to manage and monitor their network devices. The SNMP model defines two components, the SNMP Manager and the SNMP Agent.

SNMP Network Model

- The *SNMP Manager* resides on a workstation hosting the management application. It can query agents by using SNMP operations. An SNMP manager is commonly called a Network Management System (NMS). NMS refers to a system made up of a network device (such as a workstation) and the NMS software. It provides an interface that allows users to request data or see alarms resulting from traps or informs. It can also store data that can be used for network analysis.
- The *SNMP Agent* is the software entity that resides within the switch on the network. It maintains the management data about a particular network device and reports this data, as needed, to the managing systems. The agent also responds to requests for data from the SNMP Manager.

Along with the SNMP agent, the switch also contains *Management Information Bases (MIBs)*. MIBs are databases of managed objects, written in the SNMP module language, which can be monitored by the NMS. The SNMP agent contains MIB variables, which have values the NMS can request or change using Get, GetNext, GetBulk, or Set operations. The agent can also send unsolicited messages (traps or informs) to the NMS to notify the manager of network conditions.

SNMP Operations

Devices on the network are managed through transactions between the NMS and the SNMP agent residing on the network device (i.e., switch). SNMP provides two kinds of management transactions, manager-request/agent-response and unsolicited notifications (traps or informs) from the agent to the manager.

In a manager-request/agent-response transaction, the SNMP manager sends a request packet, referred to as a Protocol Data Unit (PDU), to the SNMP agent in the switch. The SNMP agent complies with the request and sends a response PDU to the manager. The types of management requests are Get, GetNext, and GetBulk requests. These transactions are used to request information from the switch (Get, GetNext, or GetBulk) or to change the value of an object instance on the switch (Set).

In an unsolicited notification, the SNMP agent in the switch sends a trap PDU to the SNMP manager to inform it that an event has occurred. The SNMP manager normally does not send confirmation to the agent acknowledging receipt of a trap.

Using SNMP for Switch Management

The Alcatel-Lucent switch can be configured using the Command Line Interface (CLI), SNMP, or the WebView device management tool. When configuring the switch by using SNMP, an NMS application (such as Alcatel-Lucent's OmniVista or HP OpenView) is used.

Although MIB browsers vary depending on which software package is used, they all have a few things in common. The browser must compile the Alcatel-Lucent switch MIBs before it can be used to manage the switch by issuing requests and reading statistics. Each MIB must be checked for dependencies and the MIBs must be compiled in the proper order. Once the browser is properly installed and the MIBs are compiled, the browser software can be used to manage the switch. The MIB browser you use depends on the design and management requirements of your network.

Detailed information on working with MIB browsers is beyond the scope of this manual. However, you must know the configuration requirements of your MIB browser or other NMS installation before you can define the system to the switch as an SNMP station.

Setting Up an SNMP Management Station

An SNMP management station is a workstation configured to receive SNMP traps from the switch. You must identify this station to the switch by using the **snmp station** CLI command.

The following information is needed to define an SNMP management station.

- The IP address of the SNMP management station device.
- The UDP destination port number on the management station. This identifies the port to which the switch will send traps.
- The SNMP version used by the switch to send traps.
- A user account name that the management station will recognize.

Procedures for configuring a management station can be found in "Quick Steps for Setting Up An SNMP Management Station" on page 9-4

SNMP Versions

The SNMP agent in the switch can communicate with multiple managers. You can configure the switch to communicate with different management stations by using different versions of SNMP. The switch supports three versions of SNMP—v1, v2, and v3.

SNMPv1

SNMPv1 is the original implementation of the SNMP protocol and network management model. It is characterized by the Get, Set, GetNext, and Trap protocol operations.

SNMPv1 uses a rudimentary security system where each PDU contains information called a *community string*. The community string acts like a combination username and password. When you configure a device for SNMP management you normally specify one community string that provides read-write access to objects within the device and another community string that limits access to read-only. If the community string in a data unit matches one of these strings, the request is granted. If not, the request is denied.

The community string security standard offers minimal security and is generally insufficient for networks where the need for security is high. Although SNMPv1 lacks bulk message retrieval capabilities and security features, it is widely used and is a de facto standard in the Internet environment.

SNMPv2

SNMPv2 is a later version of the SNMP protocol. It uses the same Get, Set, GetNext, and Trap operations as SNMPv1 and supports the same community-based security standard. SNMPv1 is incompatible with SNMPv2 in certain applications due to the following enhancements:

- Management Information Structure

  SNMPv2 includes new macros for defining object groups, traps compliance characteristics, and capability characteristics.

- Protocol Operations

  SNMPv2 has two new PDUs not supported by SNMPv1. The GetBulkRequest PDU enables the manager to retrieve large blocks of data efficiently. In particular, it is well suited to retrieving multiple rows in a table. The InformRequest PDU enables one manager to send trap information to another manager.

SNMPv3

SNMPv3 supports the View-Based Access Control Model (VACM) and User-Based Security Model (USM) security models along with these added security features:

- Message integrity—Ensuring that a packet has not been tampered with in transit.
- Time Frame Protection—Limiting requests to specified time frames. The user can specify a time frame so that any PDU bearing an out of date timestamp will be ignored.
- Encryption—Scrambling the contents of a packet to prevent it from being learned by an unauthorized source.
- Authentication—Determining that the message is from a valid source holding the correct privileges.

Using SNMP For Switch Security

Community Strings (SNMPv1 and SNMPv2)

The switch supports the SNMPv1 and SNMPv2c community strings security standard. When a community string is carried over an incoming SNMP request, the community string must match up with a user account name as listed in the community string database on the switch. Otherwise, the SNMP request will not be processed by the SNMP agent in the switch.

Configuring Community Strings

To use SNMPv1 and v2 community strings, each user account name must be mapped to an SNMP community string. Follow these steps:

1 Create a user account on the switch and define its password. Enter the following CLI syntax to create the account “community_user1”.

```
-> user community_user1 password ******* no auth
```

Note. A community string inherits the security privileges of the user account that creates it.

A user account can be created locally on the switch by using CLI commands. For detailed information on setting up user accounts, refer to the “Using Switch Security” chapter of this manual.

2 Map the user account to a community string.

A community string works like a password so it is defined by the user. It can be any text string up to 32 characters in length. If spaces are part of the text, the string must be enclosed in quotation marks (“ ”). The following CLI command maps the username “community_user1” to the community string “comstring2”.

```
-> snmp community-map comstring2 user community_user1 enable
```

3 Verify that the community string mapping mode is enabled.

By default, the community strings database is enabled. (If community string mapping is not enabled, the community string configuration will not be checked by the switch.) If the community string mapping mode is disabled, use the following command to enable it.

```
-> snmp community-map mode enable
```

Note. *Optional.* To verify that the community string is properly mapped to the username, enter the **show snmp community-map** command. The display is similar to the one shown here:

```
->show snmp community-map
Community mode : enabled

status    community string                user name
--------+--------------------------------+--------------------------------
enabled   comstring2                      community_user1
```

This display also verifies that the community map mode is enabled.

Encryption and Authentication (SNMPv3)

Two important processes are used to verify that the message contents have not been altered and that the source of the message is authentic. These processes are *encryption* and *authentication.*

A typical data *encryption process* requires an encryption algorithm on both ends of the transmission and a secret key (like a code or a password). The sending device encrypts or "scrambles" the message by running it through an encryption algorithm along with the key. The message is then transmitted over the network in its encrypted state. The receiving device then takes the transmitted message and "un-scrambles" it by running it through a decryption algorithm. The receiving device cannot un-scramble the coded message without the key.

The switch uses the Data Encryption Standard (DES) encryption scheme in its SNMPv3 implementation. For DES, the data is encrypted in 64-bit blocks by using a 56-bit key. The algorithm transforms a 64-bit input into a 64-bit output. The same steps with the same key are used to reverse the encryption.

The *authentication process* ensures that the switch receives accurate messages from authorized sources. Authentication is accomplished between the switch and the SNMP management station through the use of a username and password identified via the **snmp station** CLI syntax. The username and password are used by the SNMP management station along with an authentication algorithm (SHA or MD5) to compute a hash that is transmitted in the PDU. The switch receives the PDU and computes the hash to verify that the management station knows the password. The switch will also verify the checksum contained in the PDU.

Authentication and encryption are combined when the PDU is first authenticated by either the SHA or MD5 method. Then the message is encrypted using the DES encryption scheme. The encryption key is derived from the authentication key, which is used to decrypt the PDU on the switch's side.

Configuring Encryption and Authentication

Setting Authentication for a User Account

User account names and passwords must be a minimum of 8 characters in length when authentication and encryption are used. The following syntax sets authentication type MD5 with DES encryption for user account "user_auth1".

```
-> user user_auth1 password ******** md5+des
```

SNMP authentication types SHA and MD5 are available with and without type DES encryption. The **sha**, **md5**, **sha+des**, and **md5+des** keywords may be used in the command syntax.

Note. *Optional.* To verify the authentication and encryption type for the user, enter the **show user** command. The following is a partial display.

```
-> show user
  User name = user_auth1
  Read right          = 0x0000a200 0x00000000,
  Write right         = 0x00000000 0x00000000,
  Read for domains    = ,
  Read for families   = snmp chassis interface ,
  Write for domains   = None ,
  Snmp authentication = MD5, Snmp encryption = DES
```

The user's SNMP authentication is shown as MD5 and SNMP encryption is shown as DES.

Setting SNMP Security

By default, the switch is set to “privacy all”, which means the switch accepts only authenticated and encrypted v3 Sets, Gets, and Get-Nexts. You can configure different levels of SNMP security by entering **snmp security** followed by the command parameter for the desired security level. For example, the following syntax sets the SNMP security level as “authentication all” as defined in the table below:

```
-> snmp security authentication all
```

The command parameters shown in the following table define security from the lowest level (no security) to the highest level (traps only) as shown.

Security Level	SNMP requests accepted by the switch
no security	All SNMP requests are accepted.
authentication set	SNMPv1, v2 Gets Non-authenticated v3 Gets and Get-Nexts Authenticated v3 Sets, Gets, and Get-Nexts Encrypted v3 Sets, Gets, and Get-Nexts
authentication all	Authenticated v3 Sets, Gets, and Get-Nexts Encrypted v3 Sets, Gets, and Get-Nexts
privacy set	Authenticated v3 Gets and Get-Nexts Encrypted v3 Sets, Gets, and Get-Nexts
privacy all	Encrypted v3 Sets, Gets, and Get-Nexts
traps only	All SNMP requests are rejected.

Working with SNMP Traps

The SNMP agent in the switch has the ability to send traps to the management station. It is not required that the management station request them. Traps are messages alerting the SNMP manager to a condition on the network. A trap message is sent via a PDU issued from the switch's network management agent. It is sent to alert the management station to some event or condition on the switch.

Traps can indicate improper user authentication, restarts, the loss of a connection, or other significant events. You can configure the switch so that traps are forwarded to or suppressed from transmission to the management station under different circumstances.

Trap Filtering

You can filter SNMP traps in at least two ways. You can filter traps by limiting user access to trap families or you can filter according to individual traps.

Filtering by Trap Families

Access to SNMP traps can be restricted by withholding access privileges for user accounts to certain command families or domains. (Designation of particular command families for user access is sometimes referred to as *partition management*.)

SNMP traps are divided into functional families as shown in the "Using SNMP For Switch Security" on page 9-10. These families correspond to switch CLI command families. When read-only privileges for a user account are restricted for a command family, that user account is also restricted from reading traps associated with that family.

Procedures for filtering traps according to command families can be found in the Quick Steps for "Filtering by Trap Families" on page 9-5. For a list of trap names, command families, and their descriptions refer to the "Using SNMP For Switch Security" on page 9-10.

Filtering By Individual Trap

You can configure the switch to filter out individual traps by using the **snmp-trap filter-ip** command. This command allows you to suppress specified traps from the management station. The following information is needed to suppress specific traps:

- The IP address of the SNMP management station that will receive the traps.
- The ID number of the individual traps to be suppressed.

Procedures for filtering individual traps can be found in the Quick Steps for "Filtering by Individual Traps" on page 9-6. For a list of trap names, ID numbers, and their descriptions refer to the table "Using SNMP For Switch Security" on page 9-10.

Authentication Trap

The authentication trap is sent when an SNMP authentication failure is detected. This trap is a signal to the management station that the switch received a message from an unauthorized protocol entity. This normally means that a network entity attempted an operation on the switch for which it had insufficient authorization. When the SNMP authentication trap is enabled, the switch will forward a trap to the management station. The following command will enable the authentication trap:

```
-> snmp authentication trap enable
```

The trap will be suppressed if the SNMP authentication trap is disabled.

Trap Management

Several CLI commands allow you to control trap forwarding from the agent in the switch to the SNMP management station.

Replaying Traps

The switch normally stores all traps that have been sent out to the SNMP management stations. You can list the last stored traps by using the **show snmp-trap replay-ip** command. This command lists the traps along with their sequence number. The sequence number is a record of the order in which the traps were previously sent out.

You may want to replay traps that have been stored on the switch for testing or troubleshooting purposes. This is useful in the event when any traps are lost in the network. To replay stored traps, use the **snmp trap replay** command followed by the IP address for an SNMP management station. This command replays (or re-sends) all stored traps from the switch to the specified management station on demand.

If you do not want to replay all of the stored traps, you can specify the sequence number from which the trap replay will start. The switch will start the replay with a trap sequence number greater than or equal to the sequence number given in the CLI command. The number of traps replayed depends on the number of traps stored for this station.

Absorbing Traps

The switch may send the same traps to the management station many, many times. You can suppress the transmission of identical repetitive traps by issuing the **snmp-trap absorption** command. When trap absorption is enabled, traps that are identical to traps previously sent will be suppressed and therefore not forwarded to the SNMP management station. The following command will enable SNMP trap absorption:

```
-> snmp trap absorption enable
```

To view or verify the status of the Trap Absorption service, use the **show snmp-trap config** command.

Sending Traps to WebView

When WebView forwarding is enabled, all traps sent by switch applications are also forwarded to WebView. The following command allows a WebView session to retrieve the trap history log:

```
-> snmp trap to webview enable
```

SNMP MIB Information

MIB Tables

You can display MIB tables and their corresponding command families by using the **show snmp mib-family** command. The MIB table identifies the MIP identification number, the MIB table name and the command family. If a command family is not valid for the entire MIB table, the command family will be displayed on a per-object basis.

For a list and description of system MIBs and Traps refer to "SNMP Trap Information" section on page -1.

The following is a partial display.

```
-> show snmp mib family
MIP ID   MIB TABLE NAME                          FAMILY
-------+---------------------------------------+--------------------
6145   esmConfTrap                             NO SNMP ACCESS
6146   alcetherStatsTable                      interface
6147   dot3ControlTable                        interface
6148   dot3PauseTable                          interface
6149   dot3StatsTable                          interface
6150   esmConfTable                            interface
...
...
77828   healthModuleTable                      rmon
77829   healthPortTable                        rmon
77830   healthThreshInfo                       rmon
78849   vrrpAssoIpAddrTable                    vrrp
78850   vrrpOperTable                          vrrp
78851   vrrpOperations                         vrrp
78852   vrrpRouterStatsTable                   vrrp
...
...
87042   vacmContextTable                       snmp
87043   vacmSecurityToGroupTable               snmp
87044   vacmAccessTable                        snmp
87045   vacmViewTreeFamilyTable                snmp
```

MIB Table Description

If the user account has no restrictions, the display shown by the **show snmp mib-family** command can be very long. For documentation purposes, a partial list is shown above and three entry examples are defined.

- The first entry in the MIB Table shows an MIP identification number of 6145. The MIB table name is esmConfTrap.This table is found in the AlcatelIND1Port MIB, which defines managed objects for the ESM Driver subsystem.
- For MIP Id number 77828, the MIB table name is healthModuleTable. This table is found in the AlcatelIND1Health MIB, which defines managed objects for the health monitoring subsystem.
- For MIB Id number 87042, the MIB table name is vacmContextTable. This table is found in the SNMP-VIEW-BASED-ACM MIB, which serves as the view-based access control model (VACM) for the SNMP.

Verifying the SNMP Configuration

To display information about SNMP management stations, trap management, community strings, and security, use the **show** commands listed in the following table.

show snmp station	Displays current SNMP station information including IP address, UDP Port number, Enabled/Disabled status, SNMP version, and user account names.
show snmp community-map	Shows the local community strings database including status, community string text, and user account name.
show snmp security	Displays current SNMP security status.
show snmp statistics	Displays SNMP statistics. Each MIB object is listed along with its status.
show snmp mib-family	Displays SNMP MIB information. Information includes MIP ID number, MIB table name, and command family.
show snmp-trap replay-ip	Displays SNMP trap replay information. This includes the IP address of the SNMP station manager that replayed each trap and the number of the oldest replayed trap.
show snmp-trap filter-ip	Displays the current SNMP trap filter status. This includes the IP address of the SNMP station that recorded the traps and the identification list for the traps being filtered.
show snmp authentication-trap	Displays the current authentication failure trap forwarding status (i.e., enable or disable).
show snmp-trap config	Displays SNMP trap information including trap ID numbers, trap names, command families, and absorption rate. This command also displays the Enabled/Disabled status of SNMP absorption and the Traps to WebView service.

For more information about the resulting displays from these commands, see the *OmniSwitch CLI Reference Guide*.

10 Configuring Network Time Protocol (NTP)

Network Time Protocol (NTP) is used to synchronize the time of a computer client or server to another server or reference time source, such as a radio or satellite receiver. It provides client time accuracies within a millisecond on LANs, and up to a few tens of milliseconds on WANs relative to a primary server synchronized to Universal Coordinated Time (UTC) (via a Global Positioning Service receiver, for example).

In This Chapter

This chapter describes the basic components of the OmniSwitch implementation of Network Time Protocol and how to configure it through Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch CLI Reference Guide*.

Configuration procedures described in this chapter include:

- Enabling the NTP client and selecting the NTP mode. See "Configuring the OmniSwitch as a Client" on page 10-9.
- Selecting an NTP server for the NTP client and modifying settings for communicating with the server. See "NTP Servers" on page 10-10.
- Enabling authentication in NTP negotiations. See "Using Authentication" on page 10-12.

NTP Specifications

Platforms Supported	OmniSwitch 10K, 6900
RFCs supported	1305–Network Time Protocol
NTP Key File Location	**/flash/network**
Platforms Supported	OmniSwitch 10K, 6900
Maximum number of NTP servers per client	3

NTP Defaults Table

The following table shows the default settings of the configurable NTP parameters:

NTP Defaults

Parameter Description	Command	Default Value/Comments
Specifies an NTP server from which this switch will receive updates	**ntp server**	version: 4 minpoll: 6 prefer: no key: 0
Used to activate client	**ntp client**	disabled
Used to activate NTP client broadcast mode	**ntp src-ip preferred**	disabled
Used to set the advertised broadcast delay, in microseconds	**ntp broadcast-delay**	4000 microseconds

NTP Quick Steps

The following steps are designed to show the user the necessary commands to set up NTP on an OmniSwitch:

1 Designate an NTP server for the switch using the **ntp server** command. The NTP server provides the switch with its NTP time information. For example:

```
-> ntp server 198.206.181.139
```

2 Activate the client side of NTP on the switch using the **ntp client** command. For example:

```
-> ntp client admin-state enable
```

3 You can check the server status using the **show ntp server client-list** command, as shown:

```
-> show ntp server status 198.206.181.139
IP address          = 198.206.181.139,
Host mode           = client,
Peer mode           = server,
Prefer              = no,
Version             = 4,
Key                 = 0,
Stratum             = 2,
Minpoll             = 6 (64 seconds),
Maxpoll             = 10 (1024 seconds),
Delay               = 0.016 seconds,
Offset              = -180.232 seconds,
Dispersion          = 7.945 seconds
Root distance       = 0.026,
Precision           = -14,
Reference IP        = 209.81.9.7,
Status              = configured : reachable : rejected,
Uptime count        = 1742 seconds,
Reachability        = 1,
Unreachable count   = 0,
Stats reset count   = 1680 seconds,
Packets sent        = 1,
Packets received    = 1,
Duplicate packets   = 0,
Bogus origin        = 0,
Bad authentication = 0,
Bad dispersion      = 0,
Last Event          = peer changed to reachable,
```

4 You can check the list of servers associated with this client using the **show ntp client server-list** command, as shown:

```
-> show ntp client server-list
IP Address       Ver   Key    St     Delay        Offset         Disp
================+===+=======+====+==========+================+==========
1.2.5.6           4       0     2      0.06            -0.673       0.017
```

5 You can check the client configuration using the **show ntp status** command, as shown:

```
-> show ntp client
Current time:                   THU SEP 15 2005 17:44:54 (UTC)
Last NTP update:                THU SEP 15 2005 17:30:54
Client mode:                    enabled
Broadcast client mode:          disabled
Broadcast delay (microseconds):  4000
```

NTP Overview

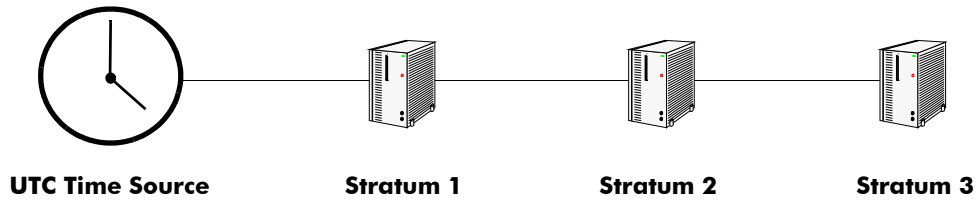
Network Time Protocol (NTP) is used to synchronize the time of a computer client or server to another server or reference time source, such as a radio or satellite receiver. It provides client time accuracies within a millisecond on LANs, and up to a few tens of milliseconds on WANs relative to a primary server synchronized to Universal Coordinated Time (UTC) (via a Global Positioning Service receiver, for example). Typical NTP configurations utilize multiple redundant servers and diverse network paths in order to achieve high accuracy and reliability. Some configurations include cryptographic authentication to prevent accidental or malicious protocol attacks.

It is important for networks to maintain accurate time synchronization between network nodes. The standard timescale used by most nations of the world is based on a combination of UTC (representing the Earth's rotation about its axis), and the Gregorian Calendar (representing the Earth's rotation about the Sun). The UTC timescale is disciplined with respect to International Atomic Time (TAI) by inserting leap seconds at intervals of about 18 months. UTC time is disseminated by various means, including radio and satellite navigation systems, telephone modems, and portable clocks.

Special purpose receivers are available for many time-dissemination services, including the Global Position System (GPS) and other services operated by various national governments. For reasons of cost and convenience, it is not possible to equip every computer with one of these receivers. However, it is possible to equip some computers with these clocks, which then act as primary time servers to synchronize a much larger number of secondary servers and clients connected by a common network. In order to do this, a distributed network clock synchronization protocol is required which can read a server clock, transmit the reading to one or more clients, and adjust each client clock as required. Protocols that do this include NTP.

Stratum

Stratum is the term used to define the relative proximity of a node in a network to a time source (such as a radio clock). Stratum 1 is the server connected to the time source itself. (In most cases the time source and the stratum 1 server are in the same physical location.) An NTP client or server connected to a stratum 1 source would be stratum 2. A client or server connected to a stratum 2 machine would be stratum 3, and so on, as demonstrated in the diagram below:



The farther away from stratum 1 a device is, the more likely there will be discrepancies or errors in the time adjustments done by NTP. A list of stratum 1 and 2 sources available to the public can be found on the Internet.

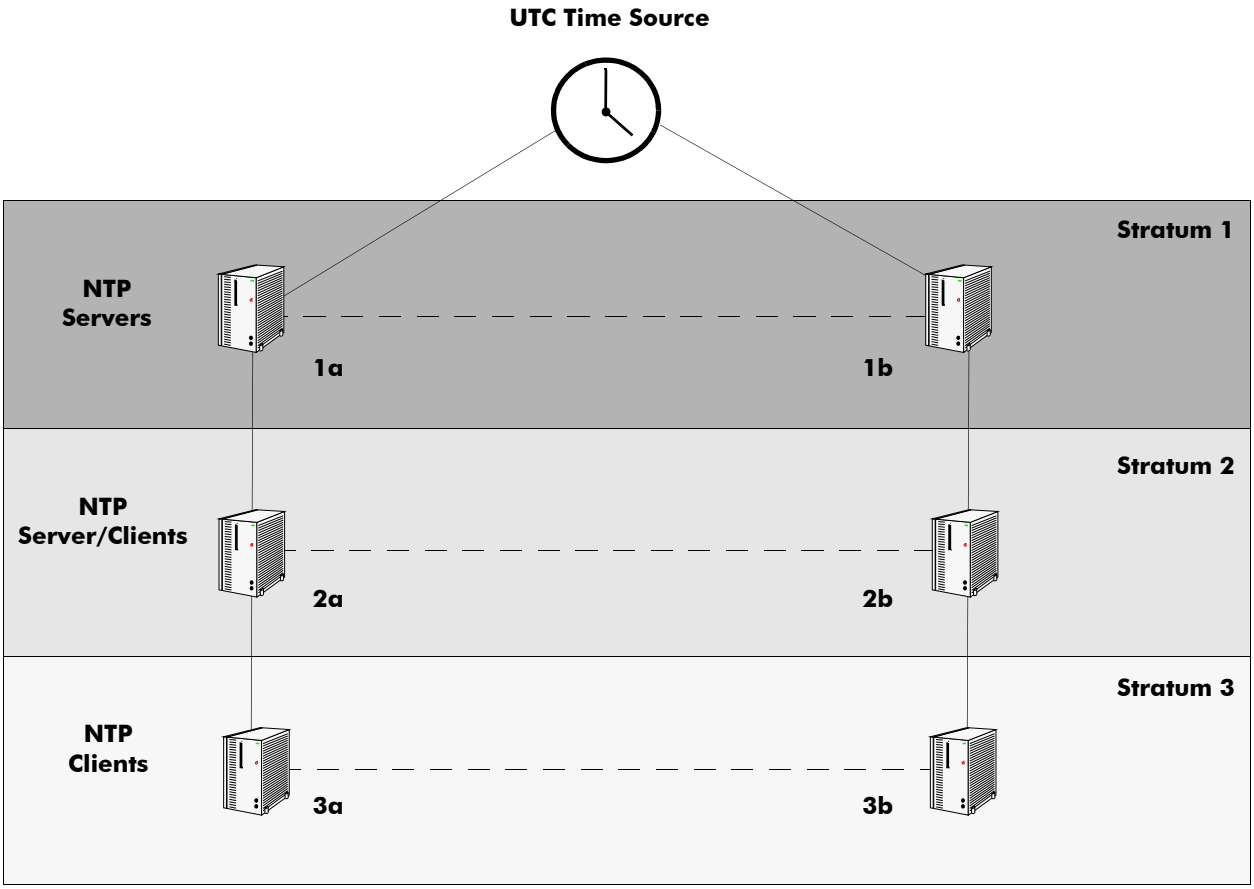
Note. It is not required that NTP be connected to an officially recognized time source (for example, a radio clock). NTP can use any time source to synchronize time in the network.

Using NTP in a Network

NTP operates on the premise that there is one true standard time (defined by UTC), and that if several servers claiming synchronization to the standard time are in disagreement, then one or more of them must be out of synchronization or not functioning correctly. The stratum gradiation is used to qualify the accuracy of a time source along with other factors, such as advertised precision and the length of the network path between connections. NTP operates with a basic distrust of time information sent from other network entities, and is most effective when multiple NTP time sources are integrated together for checks and crosschecks. To achieve this end, there are several modes of operation that an NTP entity can use when synchronizing time in a network. These modes help predict how the entity behaves when requesting or sending time information, listed below:

- A switch can be a client of an NTP server (usually of a lower stratum), receiving time information from the server but not passing it on to other switches.
- A switch can be a client of an NTP server, and in turn be a server to another switch or switches.
- A switch (regardless of its status as either a client or server) must be peered with another switch. Peering allows NTP entities in the network of the same stratum to regard each other as reliable sources of time and exchange time information.
- The OmniSwitch by default will act as an NTP server and be able to respond to NTP client requests, and establish a client/server peering relationship. The OmniSwitch NTP server functionality allows the Omniswitch to establish an active peering relationship with another server, enable broadcast server functionality, disable a given IP for NTP and employ MD5 authentication for clients and active peers.

Examples of these are shown in the simple network diagram below:



Servers 1a and 1b receive time information from, or synchronize with, a UTC time source such as a radio clock. (In most cases, these servers would not be connected to the same UTC source, though it is shown this way for simplicity.) Servers 1a and 1b become stratum 1 NTP servers and are peered with each other, allowing them to check UTC time information against each other. These machines support machines 2a and 2b as clients, and these clients are synchronized to the higher stratum servers 1a and 1b.

Clients 2a and 2b are also peered with each other for time checks, and become stratum 2 NTP servers for more clients (3a and 3b, which are also peered). In this hierarchy, the stratum 1 servers synchronize to the most accurate time source available, then check the time information with peers at the same stratum. The stratum 2 machines synchronize to the stratum 1 servers, but do not send time information to the stratum 1 machines. Machines 2a and 2b in turn provide time information to the stratum 3 machines. It is important to consider the issue of robustness when selecting sources for time synchronization.

It is suggested that at least three sources should be available, and at least one should be “close” to you in terms of network topology. It is also suggested that each NTP client is peered with at least three other same stratum clients, so that time information crosschecking is performed.

When planning your network, it is helpful to use the following general rules:

- It is usually not a good idea to synchronize a local time server with a peer (in other words, a server at the same stratum), unless the latter is receiving time updates from a source that has a lower stratum than from where the former is receiving time updates. This minimizes common points of failure.

- Peer associations should only be configured between servers at the same stratum level. Higher Strata should configure lower Strata, not the reverse.
- It is inadvisable to configure time servers in a domain to a single time source. Doing so invites common points of failure.

Note. NTP does not support year date values greater than 2035 (the reasons are documented in RFC 1305 in the data format section). This should not be a problem (until the year 2035) as setting the date this far in advance runs counter to the administrative intention of running NTP.

Authentication

NTP is designed to use MD5 encryption authentication to prevent outside influence upon NTP timestamp information. This is done by using a key file. The key file is loaded into the switch memory, and consists of a text file that lists key identifiers that correspond to particular NTP entities.

If authentication is enabled on an NTP switch, any NTP message sent to the switch must contain the correct key ID in the message packet to use in decryption. Likewise, any message sent from the authentication enabled switch will not be readable unless the receiving NTP entity possesses the correct key ID.

The key file is a text (.txt) file that contains a list of keys that are used to authenticate NTP servers.

Key files are created by a system administrator independent of the NTP protocol, and then placed in the switch memory when the switch boots. An example of a key file is shown below:

```
2       M       RIrop8KPPvQvYotM     # md5 key as an ASCII random string
14      M       sundial              # md5 key as an ASCII string
```

In a key file, the first token is the key number ID, the second is the key format, and the third is the key itself. (The text following a “#” is not counted as part of the key, and is used merely for description.) The key format indicates an MD5 key written as a 1 to 31 character ASCII string with each character standing for a key octet.

The key file (with identical MD5 keys) must be located on both the local NTP client and the client’s server.

Configuring NTP

The following sections detail the various commands used to configure and view the NTP client software in an OmniSwitch.

Configuring the OmniSwitch as a Client

The NTP software is disabled on the switch by default. To activate the switch as an NTP client, enter the **ntp client** command as shown:

```
-> ntp client admin-status enable
```

This sets the switch to act as an NTP client in the passive mode, meaning the client will receive updates from a designated NTP server.

To disable the NTP software, enter the **ntp client** command as shown:

```
-> ntp client admin-status disable
```

Setting the Client to Broadcast Mode

It is possible to configure an NTP client to operate in the broadcast mode. Broadcast mode specifies that a client switch listens on all interfaces for server broadcast timestamp information. It uses these messages to update its time.

To set an OmniSwitch to operate in the broadcast mode, enter the **ntp broadcast-client** command as shown:

```
-> ntp broadcast-client enable
```

A client in the broadcast mode does not need to have a specified server.

Setting the Broadcast Delay

When set to the broadcast mode, a client needs to advertise a broadcast delay. The broadcast mode is intended for operation on networks with numerous workstations and where the highest accuracy is not required. In a typical scenario, one or more time servers on the network, broadcast NTP messages, which are received by NTP hosts. The correct time is determined from an NTP message based on a pre-configured latency or broadcast delay in the order of a few milliseconds.

To set the broadcast delay, enter the **ntp broadcast-delay** command as shown:

```
-> ntp broadcast-delay 1000
```

NTP Servers

An NTP client needs to receive NTP updates from an NTP server. Each client must have at least one server with which it synchronizes (unless it is operating in broadcast mode). There are also adjustable server options.

Designating an NTP Server

To configure an NTP client to receive updates from an NTP server, enter the **ntp server** command with the server IP address or domain name, as shown:

```
-> ntp server 1.1.1.1
```

or

```
-> ntp server spartacus
```

It is possible to remove an NTP server from the list of servers from which a client synchronizes. To do this, enter the **ntp server** command with the **no** prefix, as shown:

```
-> no ntp server 1.1.1.1
```

Enabling/Disabling NTP Server Synchronization Tests

To enable an NTP client to invoke NTP server synchronization tests as specified by the NTP protocol, enter the **ntp server synchronized** command as shown:

```
-> ntp server synchronized
```

NTP synchronization is enabled by default.

Note. The NTP protocol discards the NTP servers that are unsynchronized.

To disable an NTP client from invoking tests for NTP server synchronization, enter the **ntp server unsynchronized** command, as shown:

```
-> ntp server unsynchronized
```

Disabling peer synchronization tests allows the NTP client to synchronize with either an NTP peer that is not synchronized with an atomic clock or a network of NTP servers that will finally synchronize with an atomic clock.

Setting the Minimum Poll Time

The minimum poll time is the number of seconds that the switch waits before requesting a time synchronization from the NTP server. This number is determined by raising 2 to the power of the number entered using the **ntp server** command with the server IP address (or domain name) and the **minpoll** keyword.

For example, to set the minimum poll time to 128 seconds, enter the following:

```
-> ntp server 1.1.1.1 minpoll 7
```

This would set the minimum poll time to 2^7 = 128 seconds.

Setting the Version Number

There are currently four versions of NTP available (numbered one through four). The version that the NTP server uses must be specified on the client side.

To specify the NTP version on the server from which the switch receives updates, use the **ntp server** command with the server IP address (or domain name), **version** keyword, and version number, as shown:

```
-> ntp server 1.1.1.1 version 3
```

The default setting is version 4.

Marking a Server as Preferred

If a client receives timestamp updates from more than one server, it is possible to mark one of the servers as the preferred server. A preferred server's timestamp will be used before another unpreferred server timestamp.

To specify an NTP as preferred, use the **ntp server** command with the server IP address (or domain name) and the **prefer** keyword, as shown:

```
-> ntp server 1.1.1.1 prefer
```

Using Authentication

Authentication is used to encrypt the NTP messages sent between the client and server. The NTP server and the NTP client must both have a text file containing the public and secret keys. (This file should be obtained from the server administrator. For more information on the authentication file, see "Authentication" on page 10-8.)

Once both the client and server share a common MD5 encryption key, the MD5 key identification for the NTP server must be specified on and labeled as trusted on the client side.

The Omniswitch will use MD5 authentication. Key files reside in /flash/network/ntp.keys.

In order to generate a key file, access to a Solaris/Unix environment is required. Also required is the ntp-keygen utility in Unix to generate the key file.

Setting the Key ID for the NTP Server

Enabling authentication requires the following steps:

1 Make sure the key file is located in the **/networking** directory of the switch. This file must contain the key for the server that provides the switch with its timestamp information.

2 Make sure the key file with the NTP server's MD5 key is loaded into the switch memory by issuing the **ntp key load** command, as shown:

```
-> ntp key load
```

3 Set the server authentication key identification number using the **ntp server** command with the **key** keyword. This key identification number must be the one the server uses for MD5 encryption. For example, to specify key identification number 2 for an NTP server with an IP address of 1.1.1.1, enter:

```
-> ntp server 1.1.1.1 key 2
```

4 Specify the key identification set above as *trusted*. A key that has been labeled as trusted is ready for use in the authentication process. To set a key identification to be trusted, enter the **ntp key** command with the key identification number and **trusted** keyword. For example, to set key ID 5 to trusted status, enter the following:

```
-> ntp key 5 trusted
```

Untrusted keys, even if they are in the switch memory and match an NTP server, will not authenticate NTP messages.

5 A key can be set to untrusted status by using the **ntp key** command with the **untrusted** keyword. For example, to set key ID 5 to untrusted status, enter the following:

```
-> ntp key 5 untrusted
```

Verifying NTP Configuration

To display information about the NTP client, use the **show** commands listed in the following table:

show ntp status	Displays information about the current client NTP configuration.
show ntp server client-list	Displays the basic server information for a specific NTP server or a list of NTP servers.
show ntp client server-list	Displays a list of the servers with which the NTP client synchronizes.
show ntp keys	Displays information about all authentication keys.

For more information about the resulting displays from these commands, see the "NTP Commands" chapter in the *OmniSwitch CLI Reference Guide*.

Examples of the **show ntp client**, **show ntp server status**, and **show ntp client server-list** command outputs are given in the section "NTP Quick Steps" on page 10-3.

A Software License and Copyright Statements

This appendix contains Alcatel-Lucent and third-party software vendor license and copyright statements.

Alcatel-Lucent License Agreement

ALCATEL-LUCENT SOFTWARE LICENSE AGREEMENT

IMPORTANT. Please read the terms and conditions of this license agreement carefully before opening this package.

By opening this package, you accept and agree to the terms of this license agreement. If you are not willing to be bound by the terms of this license agreement, do not open this package. Please promptly return the product and any materials in unopened form to the place where you obtained it for a full refund.

1. **License Grant.** This is a license, not a sales agreement, between you (the "Licensee") and Alcatel-Lucent. Alcatel-Lucent hereby grants to Licensee, and Licensee accepts, a non-exclusive license to use program media and computer software contained therein (the "Licensed Files") and the accompanying user documentation (collectively the "Licensed Materials"), only as authorized in this License Agreement. Licensee, subject to the terms of this License Agreement, may use one copy of the Licensed Files on the Licensee's system. Licensee agrees not to assign, sublicense, transfer, pledge, lease, rent, or share their rights under this License Agreement. Licensee may retain the program media for backup purposes with retention of the copyright and other proprietary notices. Except as authorized under this paragraph, no copies of the Licensed Materials or any portions thereof may be made by Licensee and Licensee shall not modify, decompile, disassemble, reverse engineer, or otherwise attempt to derive the Source Code. Licensee is also advised that Alcatel-Lucent products contain embedded software known as firmware which resides in silicon. Licensee may not copy the firmware or transfer the firmware to another medium.

2. **Alcatel-Lucent's Rights.** Licensee acknowledges and agrees that the Licensed Materials are the sole property of Alcatel-Lucent and its licensors (herein "its licensors"), protected by U.S. copyright law, trademark law, and are licensed on a right to use basis. Licensee further acknowledges and agrees that all rights, title, and interest in and to the Licensed Materials are and shall remain with Alcatel-Lucent and its licensors and that no such right, license, or interest shall be asserted with respect to such copyrights and trademarks. This License Agreement does not convey to Licensee an interest in or to the Licensed Materials, but only a limited right to use revocable in accordance with the terms of this License Agreement.

3. **Confidentiality.** Alcatel-Lucent considers the Licensed Files to contain valuable trade secrets of Alcatel-Lucent, the unauthorized disclosure of which could cause irreparable harm to Alcatel-Lucent. Except as expressly set forth herein, Licensee agrees to use reasonable efforts not to disclose the Licensed Files to any third party and not to use the Licensed Files other than for the purpose authorized by this License Agreement. This confidentiality obligation shall continue after any termination of this License Agreement.

4. **Indemnity.** Licensee agrees to indemnify, defend and hold Alcatel-Lucent harmless from any claim, lawsuit, legal proceeding, settlement or judgment (including without limitation Alcatel-Lucent's reasonable United States and local attorneys' and expert witnesses' fees and costs) arising out of or in connection with the unauthorized copying, marketing, performance or distribution of the Licensed Files.

5. **Limited Warranty.** Alcatel-Lucent warrants, for Licensee's benefit alone, that the program media shall, for a period of ninety (90) days from the date of commencement of this License Agreement (referred to as the Warranty Period), be free from defects in material and workmanship. Alcatel-Lucent further warrants, for Licensee benefit alone, that during the Warranty Period the Licensed Files shall operate substantially in accordance with the functional specifications in the User Guide. If during the Warranty Period, a defect in the Licensed Files appears, Licensee may return the Licensed Files to Alcatel-Lucent for either replacement or, if so elected by Alcatel-Lucent, refund of amounts paid by Licensee under this License Agreement. EXCEPT FOR THE WARRANTIES SET FORTH ABOVE, THE LICENSED MATERIALS ARE LICENSED "AS IS" AND ALCATEL-LUCENT AND ITS LICENSORS DISCLAIM ANY AND ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING (WITHOUT LIMITATION) ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES SO THE ABOVE EXCLUSIONS MAY NOT APPLY TO LICENSEE. THIS WARRANTY GIVES THE LICENSEE SPECIFIC LEGAL RIGHTS. LICENSEE MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

6. **Limitation of Liability.** Alcatel-Lucent's cumulative liability to Licensee or any other party for any loss or damages resulting from any claims, demands, or actions arising out of or relating to this License Agreement shall not exceed the license fee paid to Alcatel-Lucent for the Licensed Materials. IN NO EVENT SHALL ALCATEL-LUCENT BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL, OR EXEMPLARY DAMAGES OR LOST PROFITS, EVEN IF ALCATEL-LUCENT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION TO INCIDENTAL OR CONSEQUENTIAL DAMAGES MAY NOT APPLY TO LICENSEE.

7. **Export Control.** This product is subject to the jurisdiction of the United States. Licensee may not export or reexport the Licensed Files, without complying with all United States export laws and regulations, including but not limited to (i) obtaining prior authorization from the U.S. Department of Commerce if a validated export license is required, and (ii) obtaining "written assurances" from licensees, if required.

8. **Support and Maintenance.** Except as may be provided in a separate agreement between Alcatel-Lucent and Licensee, if any, Alcatel-Lucent is under no obligation to maintain or support the copies of the Licensed Files made and distributed hereunder and Alcatel-Lucent has no obligation to furnish Licensee with any further assistance, documentation or information of any nature or kind.

9. **Term.** This License Agreement is effective upon Licensee opening this package and shall continue until terminated. Licensee may terminate this License Agreement at any time by returning the Licensed Materials and all copies thereof and extracts therefrom to Alcatel-Lucent and certifying to Alcatel-Lucent in writing that all Licensed Materials and all copies thereof and extracts therefrom have been returned or erased by the memory of Licensee's computer or made non-readable. Alcatel-Lucent may terminate this License Agreement upon the breach by Licensee of any term hereof. Upon such termination by

Alcatel-Lucent, Licensee agrees to return to Alcatel-Lucent or destroy the Licensed Materials and all copies and portions thereof.

10. **Governing Law.** This License Agreement shall be construed and governed in accordance with the laws of the State of California.

11. **Severability.** Should any term of this License Agreement be declared void or unenforceable by any court of competent jurisdiction, such declaration shall have no effect on the remaining terms herein.

12. **No Waiver.** The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.

13. **Notes to United States Government Users.** Software and documentation are provided with restricted rights. Use, duplication or disclosure by the government is subject to (i) restrictions set forth in GSA ADP Schedule Contract with Alcatel-Lucent's reseller(s), or (ii) restrictions set forth in subparagraph (c) (1) and (2) of 48 CFR 52.227-19, as applicable.

14.**Third Party Materials.** Licensee is notified that the Licensed Files contain third party software and materials licensed to Alcatel-Lucent by certain third party licensors. Some third party licensors are third part beneficiaries to this License Agreement with full rights of enforcement. Please refer to the section entitled "Third Party Licenses and Notices" on page -4 for the third party license and notice terms.

Third Party Licenses and Notices

Legal Notices applicable to any software distributed alone or in connection with the product to which this document pertains, are contained in files within the software itself located at: **/flash/foss**.

Also, if needed, we provide all FOSS (Free and Open Source Software) source code used into this release at the following URL: https://service.esd.alcatel-lucent.com/portal/page/portal/EService/release

B SNMP Trap Information

This appendix lists the supported SNMP traps along with their descriptions.

SNMP Traps Table

The following table provides information on all SNMP traps supported by the switch. Each row includes the trap name, its ID number, any objects (if applicable), its command family, and a description of the condition the SNMP agent in the switch is reporting to the SNMP management station.

No.	Trap Name	Objects	Family	Description
0	coldStart	none	chassis	The SNMP agent in the switch is reinitiating and its configuration may have been altered.
1	warmStart	none	chassis	The SNMP agent in the switch is reinitiating itself and its configuration is unaltered.
2	linkDown	IfIndex ifAdminStatus ifOperStatus	interface	The SNMP agent in the switch recognizes a failure in one of the communications links configured for the switch.

IfIndex—A unique value, greater than zero, for each interface. It is recommended that values are assigned contiguously starting from 1. The value for each interface sub-layer must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.
ifAdminStatus—The desired state of the interface. The testing (3) state indicates that no operational packets can be passed. When a managed system initializes, all interfaces start with ifAdminStatus in the down (2) state. As a result of either explicit management action or per configuration information retained by the managed system, ifAdminStatus is then changed to either the up (1) or testing (3) states (or remains in the down (2) state).
ifOperStatus—The current operational state of the interface. The testing (3) state indicates that no operational packets can be passed. If ifAdminStatus is down (2) then ifOperStatus should be down(2). If ifAdminStatus is changed to up (1) then ifOperStatus should change to up (1) if the interface is ready to transmit and receive network traffic; it should change to dormant (5) if the interface is waiting for external actions (such as a serial line waiting for an incoming connection); it should remain in the down (2) state if and only if there is a fault that prevents it from going to the up (1) state; it should remain in the notPresent (6) state if the interface has missing (typically, hardware) components.

3	linkUp	ifIndex ifAdminStatus ifOperStatus	interface	The SNMP agent in the switch recognizes that one of the communications links configured for the switch has come up.

IfIndex—A unique value, greater than zero, for each interface. It is recommended that values are assigned contiguously starting from 1. The value for each interface sub-layer must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.
ifAdminStatus—The desired state of the interface. The testing (3) state indicates that no operational packets can be passed. When a managed system initializes, all interfaces start with ifAdminStatus in the down (2) state. As a result of either explicit management action or per configuration information retained by the managed system, ifAdminStatus is then changed to either the up (1) or testing (3) states (or remains in the down (2) state).
ifOperStatus—The current operational state of the interface. The testing(3) state indicates that no operational packets can be passed. If ifAdminStatus is down (2) then ifOperStatus should be down (2). If ifAdminStatus is changed to up (1), then ifOperStatus should change to up (1) if the interface is ready to transmit and receive network traffic; it should change to dormant (5) if the interface is waiting for external actions (such as a serial line waiting for an incoming connection); it should remain in the down (2) state if and only if there is a fault that prevents it from going to the up (1) state; it should remain in the notPresent (6) state if the interface has missing (typically, hardware) components.

4	authenticationFailure	none	snmp	The SNMP agent in the switch has received a protocol message that is not properly authenticated.

No.	Trap Name	Objects	Family	Description
5	entConfigChange	none	module	An entConfigChange notification is generated when a conceptual row is created, modified, or deleted in one of the entity tables.
6	policyEventNotification	policyTrapEventDetailString policyTrapEventCode	qos	The switch notifies the NMS when a significant event happens that involves the policy manager.
policyTrapEventDetailString—Details about the event that took place. **policyTrapEventCode**—The code of the event.				
7	chassisTrapsStr	chassisTrapsStrLevel chassisTrapsStrAppID chassisTrapsStrSnapID chassisTrapsStrfileName chassisTrapsStrfileLineNb chassisTrapsStrErrorNb chassisTrapsStrcomments chassisTrapsStrdataInfo	chassis	A software trouble report (STR) was sent by an application encountering a problem during its execution.
chassisTrapsStrLevel—An enumerated value that provides the urgency level of the STR. **chassisTrapsStrAppID**—The application identification number. **chassisTrapsStrSnapID**—The subapplication identification number. You can have multiple snapIDs per Subapplication (task) but only one is to be used to send STRs. **chassisTrapsStrfileName**—Name of the source file where the fault was detected. This is given by the C ANSI macro __FILE__. The path shouldn't appear. **chassisTrapsStrfileLineNb**—Line number in the source file where the fault was detected. This is given by the C ANSI macro __LINE__. **chassisTrapsStrErrorNb**—The fault identificator. The error number identifies the kind the detected fault and allows a mapping of the data contained in chassisTrapsdataInfo. **chassisTrapsStrcomments**—Comment text explaining the fault. **chassisTrapsStrdataInfo**—Additional data provided to help to find out the origin of the fault. The contained and the significant portion are varying in accordance with chassisTrapsStrErrorNb. The length of this field is expressed in bytes.			

No.	Trap Name	Objects	Family	Description
8	chassisTrapsAlert	physicalIndex chassisTrapsObjectType chassisTrapsObjectNumber chassisTrapsAlertNumber chassisTrapsAlertDescr	chassis	A notification that some change has occurred in the chassis.
physicalIndex—The physical index of the involved object. **chassisTrapsObjectType**—An enumerated value that provides the object type involved in the alert trap. **chassisTrapsObjectNumber**—A number defining the order of the object in the set (e.g., the number of the considered fan or power supply). This is intended to clarify as much as possible the location of the failure or alert. An instance of the appearance of the trap could be “failure on a module. Power supply 3”. **chassisTrapsAlertNumber**—This number that identifies the alert among all the possible chassis alert causes. **chassisTrapsAlertDescr**— The description of the alert matching ChassisTrapsAlertNumber.				
9	chassisTrapsStateChange	physicalIndex chassisTrapsObjectType chassisTrapsObjectNumber chasEntPhysOperStatus	chassis	An NI status change was detected.
physicalIndex—The physical index of the involved object. **chassisTrapsObjectType**—An enumerated value that provides the object type involved in the alert trap. **chassisTrapsObjectNumber**—A number defining the order of the object in the set (e.g., the number of the considered fan or power supply). This intends to clarify as much as possible the location of the failure or alert. An instance of the appearance of the trap could be “failure on a module. Power supply 3”. **chasEntPhysOperStatus**—An enumerated value that indicates the operational status of installed modules (includes empty slots).				
10	chassisTrapsMacOverlap	physicalIndex chasTrapMacRangeIndex	module	A MAC range overlap was found in the backplane eeprom.
physicalIndex—The physical index of the involved object. **chasTrapMacRangeIndex**—The MAC range index of the involved object.				
11	vrrpTrapNewMaster	vrrpOperMasterIpAddr	vrrp	The SNMP agent has transferred from the backup state to the master state.
vrrpOperMasterIpAddr—The master router’s real (primary) IP address. This is the IP address listed as the source in the VRRP advertisement last received by this virtual router.				
12	vrrpTrapAuthFailure	vrrpTrapPacketSrc vrrpTrapAuthErrorType	vrrp	A packet was received from the network whose authentication key conflicts with the switch’s authentication key or type.
vrrpTrapPacketSrc—The IP address of an inbound VRRP packet. **vrrpTrapAuthErrorType**—Potential types of configuration conflicts.				

No.	Trap Name	Objects	Family	Description
13	healthMonModuleTrap	healthModule-Slot healthMonRx-Status healthMonRx-TxStatus healthMon-MemorySta-tus healthMonC-puStatus	health	Indicates a module-level thresh-old was crossed.
healthModuleSlot—The (one-based) front slot number within the chassis. **healthMonRxStatus**—Rx threshold status indicating if threshold was crossed or no change. **healthMonRxTxStatus**—RxTx threshold status indicating if threshold was crossed or no change. **healthMonMemoryStatus**—Memory threshold status indicating if threshold was crossed or no change. **healthMonCpuStatus**—CPU threshold status indicating if threshold was crossed or no change.				
14	healthMonPortTrap	healthPortSlot healthPortIF healthMonRx-Status healthMonRx-TxStatus	health	Indicates a port-level threshold was crossed.
healthPortSlot—The physical slot number for this port. **healthPortIF**—The on-board interface number. **healthMonRxStatus**—Rx threshold status indicating if threshold was crossed or no change. **healthMonRxTxStatus**—RxTx threshold status indicating if threshold was crossed or no change.				
15	healthMonCmmTrap	healthMon-MemorySta-tus, healthMonC-puStatus	health	This trap is sent when an NI memory or CPU threshold is crossed.
healthMonMemoryStatus—Memory threshold status indicating if threshold was crossed or no change. **healthMonCpuStatus**—CPU threshold status indicating if threshold was crossed or no change.				
16	bgpEstablished	bgpPeerLastEr-ror bgpPeerState	bgp	The BGP routing protocol has entered the established state.
bgpPeerLastError—The last error code and subcode seen by this peer on this connection. If no error has occurred, this field is zero. Otherwise, the first byte of this two byte OCTET STRING contains the error code, and the second byte contains the subcode. **bgpPeerState**—The BGP peer connection state.				
17	bgpBackwardTransition	bgpPeerLastEr-ror bgpPeerState	bgp	This trap is generated when the BGP router port has moved from a more active to a less active state.
bgpPeerLastError—The last error code and subcode seen by this peer on this connection. If no error has occurred, this field is zero. Otherwise, the first byte of this two byte OCTET STRING contains the error code, and the second byte contains the subcode. **bgpPeerState**—The BGP peer connection state.				

No.	Trap Name	Objects	Family	Description
18	esmDrvTrapDropsLink	esmPortSlot esmPortIF ifInErrors ifOutErrors esmDrvTrap-Drops	interface	This trap is sent when the Ethernet code drops the link because of excessive errors.

esmPortSlot—The physical slot number for this Ethernet Port. The slot number has been added to be used by the private trap.
esmPortIF—The on-board interface number for this Ethernet port. The port number has been added to be used by the private trap.
ifInErrors—For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter caifIndexn occur at re-initialization of the management system and at other times as indicated by the value of ifCounterDiscontinuityTime.
ifOutErrors—For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times as indicated by the value of ifCounterDiscontinuity-Time.
esmDrvTrapDrops— Partitioned port (separated due to errors).

19	portViolationTrap	ifIndex, portVi-olation-Source, portViolation-Reason	port	This trap is sent when a port violation occurs. The trap will indicate the source of the violation and the reason for the violation

ifIndex—A unique value, greater than zero, for the interface.
portViolationSource—The source of the port violation. The source is the feature or module that has caused the violation - 1. Source Learning, 2. QOS Policy, 3. Net Sec, 4. UDLD, 5. NI Supervison (Fabric Stability). When there is no value the value is “0".
portViolationReason—The reason for the port violation. It is application specific, and indicates first Violation that happened on this port - 1. pvSLLpsShutDown, 2. pvSLLpsRestrict, 3. pvQosPolicy, 4. pvQosSpoofed, 5. pvQosBpdu, 6. pvQosBgp, 7. pvQosOspf, 8. pvQosRip, 9. pvQosVrrp, 10. pvQosDhcp, 11. pvQosPim, 12. pvQosDvmrp, 13. pvQosIsis, 14. pvQosDnsReply, 15. pvUdld.

20	dvmrpNeighborLoss	dvmrpInterface-LocalAddress dvmrpNeigh-borState	ipmr	A 2-way adjacency relationship with a neighbor has been lost. This trap is generated when the neighbor state changes from “active” to “one-way,” “ignoring” or “down.” The trap is sent only when the switch has no other neighbors on the same interface with a lower IP address than itself.

dvmrpInterfaceLocalAddress—The IP address this system will use as a source address on this interface. On unnumbered interfaces, it must be the same value as dvmrpInterfaceLocalAddress for some interfaces on the system.
dvmrpNeighborState—State of the neighbor adjacency.

No.	Trap Name	Objects	Family	Description
21	dvmrpNeighborNotPruning	dvmrpInterface-LocalAddress dvmrpNeigh-borCapabili-ties	ipmr	A non-pruning neighbor has been detected in an implementation-dependent manner. This trap is generated at most once per generation ID of the neighbor. For example, it should be generated at the time a neighbor is first heard from if the prune bit is not set. It should also be generated if the local system has the ability to tell that a neighbor which sets the prune bit is not pruning any branches over an extended period of time. The trap should be generated if the router has no other neighbors on the same interface with a lower IP address than itself.
dvmrpInterfaceLocalAddress—The IP address this system will use as a source address on this interface. On unnumbered interfaces, it must be the same value as dvmrpInterfaceLocalAddress for some interfaces on the system. **dvmrpNeighborCapabilities**—This object describes the neighboring router's capabilities. The leaf bit indicates that the neighbor has only one interface with neighbors. The prune bit indicates that the neighbor supports pruning. The generationID bit indicates that the neighbor sends its generationID in Probe messages. The mtrace bit indicates that the neighbor can handle mtrace requests.				
22	risingAlarm	alarmIndex alarmVariable alarmSample-Type alarmValue alarmRisingTh-reshold	rmon	An Ethernet statistical variable has exceeded its rising threshold. The variable's rising threshold and whether it will issue an SNMP trap for this condition are configured by an NMS station running RMON.
alarmIndex—An index that uniquely identifies an entry in the alarm table. Each such entry defines a diagnostic sample at a particular interval for an object on the device. **alarmVariable**—The object identifier of the particular variable to be sampled. Only variables that resolve to an ASN.1 primitive type of INTEGER (INTEGER, Integer32, Counter32, Counter64, Gauge, or TimeTicks) may be sampled. **alarmSampleType**—The method of sampling the selected variable and calculating the value to be compared against the thresholds. If the value of this object is absoluteValue (1), the value of the selected variable will be compared directly with the thresholds at the end of the sampling interval. If the value of this object is deltaValue (2), the value of the selected variable at the last sample will be subtracted from the current value, and the difference compared with the thresholds. **alarmValue**—The value of the statistic during the last sampling period. For example, if the sample type is deltaValue, this value will be the difference between the samples at the beginning and end of the period. If the sample type is absoluteValue, this value will be the sampled value at the end of the period. **alarmRisingThreshold**—A threshold for the sampled statistic. When the current sampled value is greater than or equal to this threshold, and the value at the last sampling interval was less than this threshold, a single event will be generated. A single event will also be generated if the first sample after this entry becomes valid is greater than or equal to this threshold and the associated alarmStartupAlarm is equal to risingAlarm (1) or risingOrFallingAlarm (3).			

No.	Trap Name	Objects	Family	Description
23	fallingAlarm	alarmIndex alarmVariable alarmSample-Type alarmValue alarmFallingTh-reshold	rmon	An Ethernet statistical variable has dipped below its falling threshold. The variable's falling threshold and whether it will issue an SNMP trap for this condition are configured by an NMS station running RMON.
	alarmIndex—An index that uniquely identifies an entry in the alarm table. Each such entry defines a diagnostic sample at a particular interval for an object on the device. **alarmVariable**—The object identifier of the particular variable to be sampled. Only variables that resolve to an ASN.1 primitive type of INTEGER (INTEGER, Integer32, Counter32, Counter64, Gauge, or TimeTicks) may be sampled. **alarmSampleType**—The method of sampling the selected variable and calculating the value to be compared against the thresholds. If the value of this object is absoluteValue (1), the value of the selected variable will be compared directly with the thresholds at the end of the sampling interval. If the value of this object is deltaValue (2), the value of the selected variable at the last sample will be subtracted from the current value, and the difference compared with the thresholds. **alarmValue**—The value of the statistic during the last sampling period. For example, if the sample type is deltaValue, this value will be the difference between the samples at the beginning and end of the period. If the sample type is absoluteValue, this value will be the sampled value at the end of the period. **alarmFallingThreshold**—A threshold for the sampled statistic. When the current sampled value is less than or equal to this threshold, and the value at the last sampling interval was greater than this threshold, a single event will be generated. A single event will also be generated if the first sample after this entry becomes valid is less than or equal to this threshold and the associated alarmStartupAlarm is equal to fallingAlarm (2) or risingOr-FallingAlarm (3).		
24	stpNewRoot	vStpNumber	stp	Sent by a bridge that became the new root of the spanning tree.
	vStpNumber—The Spanning Tree number identifying this instance.			
25	stpRootPortChange	vStpNumber vStpRootPort-Number	stp	A root port has changed for a spanning tree bridge. The root port is the port that offers the lowest cost path from this bridge to the root bridge.
	vStpNumber—The Spanning Tree number identifying this instance. **vStpRootPortNumber**—The port ifindex of the port which offers the lowest cost path from this bridge to the root bridge for this spanning tree instance.			
26	mirrorConfigError	mirmonPrima-rySlot mirmonPrima-ryPort mirroringSlot mirroringPort mirMonErrorNi mirMonError	pmm	The mirroring configuration failed on an NI. This trap is sent when any NI fails to configure mirroring. Due to this error, port mirroring session will be terminated.
	mirmonPrimarySlot—Slot of mirrored or monitored interface. **mirmonPrimaryPort**—Port of mirrored or monitored interface. **mirroringSlot**—Slot of mirroring interface. **mirroringPort**—Port of mirroring interface. **mirMonErrorNi**—The NI slot number. **mirMonError**—The Error returned by the NI which failed to configure Mirroring/Monitoring.			

No.	Trap Name	Objects	Family	Description
27	mirrorUnlikeNi	mirmonPrimarySlot mirmonPrimaryPort mirroringSlot mirroringPort mirMonErrorNi	pmm	The mirroring configuration is deleted due to the swapping of different NI board type. The Port Mirroring session which was active on a slot cannot continue with the insertion of different NI type in the same slot.
mirmonPrimarySlot—Slot of mirrored or monitored interface. **mirmonPrimaryPort**—Port of mirrored or monitored interface. **mirroringSlot**—Slot of mirroring interface. **mirroringPort**—Port of mirroring interface. **mirMonErrorNi**—The NI slot number. **mirMonError**—The Error returned by the NI which failed to configure Mirroring/Monitoring.				
28	slbTrapOperStatus	slbTrapInfoEntityGroup slbTrapInfoOperStatus slbTrapInfoClusterName slbTrapInfoServerIpAddr	load balancing	A change occurred in the operational status of the server load balancing entity.
slbTrapInfoEntityGroup—The entity group inside SLB management. **slbTrapInfoOperStatus**—The operational status of an SLB cluster or server. **slbTrapInfoClusterName**—A change occurred in the operational status of an SLB entity. **slbTrapInfoServerIpAddr**—The IP address of a server. **Note**: This trap is not supported.				
29	sessionAuthenticationTrap	sessionAccessType sessionUserName sessionUserIpAddress sessionAuthFailure	session	An authentication failure trap is sent each time a user authentication is refused.
sessionAccessType—The access type of the session. **sessionUserName**—The user name of the user logged-in. **sessionUserIpAddress**—The IP address of the user logged-in.				
30	trapAbsorptionTrap	trapAbsorStamp trapAbsorTrapId trapAbsorCounter trapAbsorTime	none	The absorption trap is sent when a trap has been absorbed at least once.
trapAbsorStamp—The time stamp of the absorbed trap. **trapAbsorTrapId**—The trap identifier of the absorbed trap. **trapAbsorCounter**—The number of the iterations of the absorbed trap. **trapAbsorTime**—The time stamp of the last iteration.				

No.	Trap Name	Objects	Family	Description
31	alaDoSTrap	alaDoSType alaDoSDetected	ip	Indicates that the sending agent has received a Denial of Service (DoS) attack.
alaDoSType—Index field for the alaDoSTable. Integer indicating the DoS Type: 0=portscan, 1=tcpsyn, 2=pingofdeath, 3=smurf, 3=pepsi, 5=land and 6=teardropBonkBoink. **alaDoSDetected**—Number of attacks detected				
pethMainPseConsumptionPower—Measured usage power expressed in Watts. **Note**: This trap is not supported on OmniSwitch 6400, 6800, 6850, and 6855 switches.				
32	ospfNbrStateChange	ospfRouterId ospfNbrIpAddr ospfNbrAd-dressLessIn-dex ospfNbrRtrId ospfNbrState	ospf	Indicates a state change of the neighbor relationship.
ospfRouterId—A 32-bit integer uniquely identifying the router in the Autonomous System. By convention, to ensure uniqueness, this should default to the value of one of the router's IP interface addresses. **ospfNbrIpAddr**—The IP address this neighbor is using in its IP Source Address. Note that, on address-less links, this will not be 0.0.0.0, but the address of another of the neighbor's interfaces. **ospfNbrAddressLessIndex**—On an interface having an IP Address, zero. On address-less interfaces, the corresponding value of ifIndex in the Internet Standard MIB. On row creation, this can be derived from the instance. **ospfNbrRtrId**—A 32-bit integer (represented as a type IpAddress) uniquely identifying the neighboring router in the Autonomous System. **ospfNbrState**—The State of the relationship with this Neighbor.				
33	ospfVirtNbrStateChange	ospfRouterId ospfVirtN-brArea ospfVirtN-brRtrId ospfVirtN-brState	ospf	Indicates a state change of the virtual neighbor relationship.
ospfRouterId—A 32-bit integer uniquely identifying the router in the Autonomous System. By convention, to ensure uniqueness, this should default to the value of one of the router's IP interface addresses. **ospfVirtNbrArea**—The Transit Area Identifier. **ospfVirtNbrRtrId**—A 32-bit integer uniquely identifying the neighboring router in the Autonomous System. **ospfVirtNbrState**—The state of the Virtual Neighbor Relationship.				
34	lnkaggAggUp	traplnkaggId traplnkaggPor-tIfIndex	linkaggre-gation	Indicates the link aggregate is active. This trap is sent when any one port of the link aggregate group goes into the attached state.
traplnkaggId—Index value of the Link Aggregate group. **traplnkaggIfIndex**—Port of the Link Aggregate group.				
35	lnkaggAggDown	traplnkaggId traplnkaggPor-tIfIndex	linkaggre-gation	Indicates the link aggregate is not active. This trap is sent when all ports of the link aggregate group are no longer in the attached state.

No.	Trap Name	Objects	Family	Description
traplnkaggId—Index value of the Link Aggregate group. **traplnkaggIfIndex**—Port of the Link Aggregate group.				
36	lnkaggPortJoin	traplnkaggId traplnkaggPortIfIndex	linkaggregation	This trap is sent when any given port of the link aggregate group goes to the attached state.
traplnkaggId—Index value of the Link Aggregate group. **traplnkaggIfIndex**—Port of the Link Aggregate group.				
37	lnkaggPortLeave	traplnkaggId traplnkaggPortIfIndex	linkaggregation	This trap is sent when any given port detaches from the link aggregate group.
traplnkaggId—Index value of the Link Aggregate group. **traplnkaggIfIndex**—Port of the Link Aggregate group.				
38	lnkaggPortRemove	traplnkaggId traplnkaggPortIfIndex	linkaggregation	This trap is sent when any given port of the link aggregate group is removed due to an invalid configuration.
traplnkaggId—Index value of the Link Aggregate group. **traplnkaggIfIndex**—Port of the Link Aggregate group.				
39	monitorFileWritten	mirmonPrimarySlot mirmonPrimaryPort monitorFileName monitorFileSize	pmm	A File Written Trap is sent when the amount of data requested by the user has been written by the port monitoring instance.
mirmonPrimarySlot—Slot of mirrored or monitored interface. **mirmonPrimaryPort**—Port of mirrored or monitored interface. **monitorFileName**—The name of the file in which the traffic will be stored (the default is "PMONITOR.ENC"). **monitorFileSize**—The number of bytes in 16K (16384) increments allowed for the file (default 16384 bytes). The file contains only the last **monitorFileName** bytes of the current port monitoring instance.				
40	alaVrrp3TrapProtoError	alaVrrp3TrapProtoErrReason	vrrp	The error trap indicates that the sending agent has encountered the protocol error.
alaVrrp3TrapProtoErrReason—This indicates the reason for protocol error trap.				
41	alaVrrp3TrapNewMaster	alaVrrp3OperMasterlpAddrType alaVrrp3OperMasterlpAddr alaVrrp3TrapNewMasterReason	vrrp	The newMaster trap indicates that the sending agent has transitioned to Master state.
alaVrrp3OperMasterlpAddrType—This specifies the type of alaVrrp3OperMasterlpAddr in this row. **alaVrrp3OperMasterlpAddr**—The master switch's real (primary for vrrp over IPv4) IP address. This is the Ip address listed as the source in the advertisement last received by this virtual switch. For IPv6, a link local address. **alaVrrp3TrapNewMasterReason**—This indicates the reason for NewMaster trap.				

No.	Trap Name	Objects	Family	Description
42	chassisTrapsPossibleDuplicateMac	physicalIndex baseMacAddress	chassis	This trap is sent when there is a possiblity of duplicate a MAC address in the network.
physicalIndex—The Physical index of the involved object. **baseMacAddress**—The base MAC Address.				
43	lldpRemTablesChange	lldptatsRemTablesInserts lldptatsRemTablesDeletes lldptatsRemTablesDrops lldptatsRemTablesAgeouts	aip	This trap is sent when the value ofthe LLDP Stats Rem Table Last ChangeTime changes. It can be utilized by an NMS to trigger LLDP remote systems table maintenance polls.
lldptatsRemTablesInserts—The number of times the complete set of information advertised by a particular MSAP has been inserted into tables contained in lldpRemoteSystemsData and lldpExtensions objects. **lldptatsRemTablesDeletes**—The number of times the complete set of information advertised by a particular MSAP has been deleted from tables contained in lldpRemoteSystemsData and lldpExtensions objects **lldptatsRemTablesDrops**—The number of times the complete set of information advertised by a particular MSAP could not be entered into tables contained in lldpRemoteSystemsData and lldpExtensions objects because of insufficient resources **lldptatsRemTablesAgeouts**—The number of times the complete set of information advertised by a particular MSAP has been deleted from tables contained in lldpRemoteSystemsData and lldpExtensions objects because the information timeliness interval has expired.				
44	pimNeighborLoss	pimNeighborUpTime	ipmr	This trap is sent when an adjacency with a neighbor is lost. The notification is generated when the neighbor timer expires, and the router has no other neighbors on the same interface with the same IP version and a lower IP address than itself. The notification is generated whenever the PIM NeighborLoss Count is incremented, subject to the rate limit specified by the PIM Neighbor Loss Notification-Period.
pimNeighborUpTime—The time since this PIM neighbor (last) became a neighbor of the local router.				

No.	Trap Name	Objects	Family	Description
45	pimInvalidRegister	PimGroupMappingPimMode pimInvalidRegisterAddressType pimInvalidRegisterOrigin pimInvalidRegisterGroup pimInvalidRegisterRp	ipmr	This trap is sent when an invalid PIM Register message is received. The notification is generated whenever the PIM Invalid Register Message Reveived counter is incremented, subject to the rate limit specified by the Invalid Register NotificationPeriod.

pimGroupMappingPimMode—The PIM mode used for groups in this group prefix.
pimInvalidRegisterAddressType—The address type stored in pimInvalidRegisterOrigin, pimInvalid RegisterGroup and pimInvalidRegisterRp. If no unexpected Register messages are received, the onject is set to "Unknown".
pimInvalidRegisterOrigin—The source address of the last unexpected Register message received by thisdevice
pimInvalidRegisterGroup—The IP multicast group address to which the last unexpected Register message received by this device was addressed.
pimInvalidRegisterRp—The RP address to which the last unexpected Register message received by this device was delivered.

46	pimInvalidJoinPrune	pimGroupMappingPimMode pimInvalidJoinPruneAddressType pimInvalidJoinPruneOrigin pimInvalidJoinPruneGroup pimInvalidJoinPruneRp pimNeighborUpTime	ipmr	This trap is sent when an invalid PIM Join/Prune message is received. The notification is generated whenever the PIM Invalid Join Prune Messages Recieved counter is incremented, subject to the rate limit specified by the PIM Invalid Join/Prune Notification Period.

pimGroupMappingPimMode—The PIM mode used for groups in this group prefix.
pimInvalidRegisterAddressType—The address type stored in pimInvalidRegisterOrigin, pimInvalid RegisterGroup and pimInvalidRegisterRp. If no unexpected Register messages are received, the onject is set to "Unknown".
pimInvalidJoinPruneOrigin—The source address of the last unexpected Join/Prune message received
pimInvalidJoinPruneGroup—The IP multicast group address carried in the last unexpected Join/Prune message received
pimInvalidJoinPruneRp—The RP address carried in the last unexpected Join/Prune message received
pimNeighborUpTime—The time since this PIM neighbor (last) became a neighbor of the local router.

No.	Trap Name	Objects	Family	Description
47	PimRPMappingChange	pimGroupMappingPimMode pimGroupMappingPrecedence	ipmr	This trap is sent when a change is detected to the active RP mapping on the device. The notification is generated whenever the PIM RP Mapping Change Count is incremented, subject to the rate limit specified by PIM RP Mapping Change Notification Period
pimGroupMappingPimMode—The PIM mode used for groups in this group prefix. **pimGroupMappingPrecedence**—The value for pimGroupMappingPrecedence to be used for this static RP configuration. This allows fine control over which configuration is overridden by this static configuration				
48	PimInterfaceElection	pimInterfaceAddressType pimInterfaceAddress	ipmr	This trap is sent when a new DR or DR has been elected on a network. The notification is generated whenever the counter PIM Interface Elections Win Count is incremented, subject to the rate limit specified by PIM Interface Election Notification Period.
pimInterfaceAddressType—The address type of the PIM interface. **pimInterfaceAddress**—The primary IP address of this router on this PIM interface.				
49	pimBsrElectedBSRLostElection	pimBsrElectedBSRAddressType, pimBsrElectedBSRAddress, pimBsrElected	ipmr	This trap is sent when the current E-BSR loses an election to a new Candidate-BSR.
pimBsrElectedBSRAddressType—The address type of the elected BSR. **pimBsrElectedBSRAddress**—The unicast address of the elected BSR. **pimBsrElectedBSRPriority**—The priority value for the elected BSR for this address type. Higher values for this object indicate higher priorities (0 - 255).				
50	pimBsrCandidateBSRWinElection pim	pimBsrCandidateBSRElectedBSR	ipmr	This trap is sent when a C-BSR wins a BSR Election.
pimBsrCandidateBSR ElectedBSR—Indicates whether the local router is the elected BSR for this zone.				

No.	Trap Name	Objects	Family	Description
51	lpsViolationTrap	lpsTrapSwitch-Name lpsTrapSwitchI-pAddr lpsTrapSwitch-Slice lpsTrapSwitch-Port lpsTrapViolat-ingMac lpsTrapViola-tionType systemServices-Date systemServices-Time	bridge	A Learned Port Security (LPS) violation has occurred.
	lpsTrapSwitchName—The name of the switch. **lpsTrapSwitchIpAddr**—The IP address of switch. **lpsTrapSwitchSlice**— The physical slice number for the LPS port on which the violation occurred. **lpsTrapSwitchPort**—The physical port number on which the violation occurred. **lpsTrapViolatingMac**—The violating MAC address. **lpsTrapViolationType**—The type of violation that occurred on the LPS port. **systemServicesDate**—This object contains the current System Date in the following format: MM/DD/YYYY. **systemServicesTime**—This object contains the current System Time in the following format: HH:MM:SS.			
52	lpsPortUpAfterLearningWindowEx-piredT	lpsTrapSwitch-Name lpsTrapSwitch-Slice lpsTrapSwitch-Port systemServices-Date systemServices-Time	bridge	This trap is sent when an LPS port joins or is enabled after the Learning Window is expired, disabling the MAC address learning on the port. This trap is also generated at the time the Learning Window expires, with a slice and port value of 0.
	lpsTrapSwitchName—The name of the switch. **lpsTrapSwitchSlice**—The slot number for the LPS port on which the violation occured **lpsTrapSwitchPort**—The port number for the LPS port on which the violation occured **systemServicesDate**—The current System Date in the following format: MM/DD/YYYY. **systemServicesTime**—The current System Time in the following format: HH:MM:SS.			
53	lpsLearnTrap	lpsLearn-TrapThreshold	bridge	This trap is sent when the number of bridged MACs learned matches the configured Learned Trap Threshhold. A trap is then generated or every additional MAC that is learned.
	lpsLearnTrapThreshold—The number of bridged MAC addresses that can be learned before a trap is sent.			
54	gvrpVlanLimitReachedEvent	alaGvrpMaxV-lanLimit	bridge	This trap is sent when the number of dynamically-learned VLANs has reached the configured limit.

No.	Trap Name	Objects	Family	Description
alaGvrpMaxVlanLimit—The maximum number of dynamic VLANs that can be created on the system by GVRP before a trap is sent.				
55	alaNetSecPortTrapAnomaly	alaNetSecPort-TrapInfoIfId, alaNetSecPort-TrapInfoA-nomaly, alaNetSecPort-TrapInfoType	netsec	This trap is sent when and anom-alout port quarantine is detected.
alaNetSecPortTrapInfoIfId—The interface index of port on which anomaly is detected. **alaNetSecPortTrapInfoAnomaly—**The yype of anomaly detected on the interface. **alaNetSecPortTrapInfoType—**The ature of anomaly. Informs if system attached to interface is source or target of the anomaly.				
56	alaNetSecPortTrapQuarantine	alaNetSecPort-TrapInfoIfId	netsec	This trap is sent when and anom-alout port quarantine is detected.
alaNetSecPortTrapInfoIfId—The interface index of port on which anomaly is detected.				
57	ifMauJabberTrap	ifMauJabber-State	interface	This trap is sent whenever a man-aged interface MAU enters the jabber state.
ifMauJabberState—The value other(1) is returned if the jabber state is not 2, 3, or 4. The agent MUST always return other(1) for MAU type dot3MauTypeAUI. The value unknown(2) is returned when the MAU's true state is unknown; for example, when it is being initialized. If the MAU is not jabbering the agent returns noJabber(3). This is the "normal" state. If the MAU is in jabber state the agent returns the jabbering(4) value.				
58	udldStateChange	alaUdldPortI-fIndex alaUdldPre-vState alaUdldCurrent-State	interface	This trap is sent when the UDLD state of a port has changed.
alaUdldPortIfIndex—The interface index of the port which troggered the UDLD trap. **alaUdldPrevState**—The previous UDLD state of the port - notapplicable (0), shutdown (1), undetermined (2), bidirectional (3). **alaUdldCurrentState**—he current UDLD state of the port - notapplicable (0), shutdown (1), undetermined (2), bidirectional (3).				
59	ndpMaxLimitReached	none	ipv6	This trap is sent when the hard-ware table has reached the maximum number of entries sup-ported. The OS6400 will not generate new ARP request for new nexthops.
60	ripRouteMaxLimitReached	none	rip	This trap is sent when the RIP database reaches the supported maximum number of entries. When the maximum number is reached, RIP discards any new updates.

No.	Trap Name	Objects	Family	Description
61	ripngRouteMaxLimitReached	none	ripng	This trap is sent when the RIPng database reaches the supported maximum number of entries. When the maximum number is reached, RIPng discards any new updates.
62	alaErpRingStateChanged	alaErpRingId alaErpRingState	erp	This trap is sent when the ERP Ring State has changed from "Idle" to "Protection".
alaErpRingId—The unique Ring identifier. **alaErpRingState**—The current state of the Ring (0=Idle, 1=Protection).				
63	alaErpRingMultipleRpl	alaErpRingId	erp	This trap is sent when multiple RPLs are detected in the Ring.
alaErpRingId—The unique Ring identifier.				
64	alaErpRingRemoved	alaErpRingId	erp	This trap is sent when the Ring is removed dynamically.
alaErpRingId—The unique Ring identifier.				
65	ntpMaxAssociation		ntp	This trap is generated when the the maximum number of peer and client associations configured for the switch is exceeded.
NtpMaxAssociation—The maximum number of peer and client associations that the switch will serve.				
66	ddmTemperatureThresholdViolated	ifIndex ddmNotificationType ddmTemperature		This trap is sent when an SFP/XFP/SFP+ temperature has crossed any threshold or reverted from previous threshold violation for a port represented by ifIndex. It also provides the current realtime value of SFP/XFP/SFP+ temperature.
ifIndex—The interface index. **ddmNotificationType—**The trap type for monitored DDM parameters (clearViolation(1), highAlarm(2), highWarning(3), lowWarning(4), lowAlarm(5). **ddmTemperature—**The temperature, in tenths of a degree celcius.				
67	ddmVoltageThresholdViolated	ifIndex ddmNotificationType ddmSupplyVoltage	port	This trap is sent when SFP/XFP/SFP+ supply voltage has crossed any threshold or reverted from previous threshold violation for a port represented by ifIndex. It also provides the current realtime value of SFP/XFP/SFP+ supply voltage.
ifIndex—The interface index. **ddmNotificationType—**The trap type for monitored DDM parameters (clearViolation(1), highAlarm(2), highWarning(3), lowWarning(4), lowAlarm(5) **ddmSupplyVoltage—**The voltage, in tenths of a volt.				

No.	Trap Name	Objects	Family	Description
68	ddmCurrentThresholdViolated	ifIndex, ddmNotificationType ddmTxBiasCurrent	port	This trap is sent when if an SFP/XFP/SFP+ Tx bias current has crossed any threshold or reverted from previous threshold violation for a port represented by ifIndex. It also provides the current real-time value of SFP/XFP/SFP+ Tx bias current.
ifIndex—The interface index. **ddmNotificationType—**The trap type for monitored DDM parameters (clearViolation(1), highAlarm(2), highWarning(3), lowWarning(4), lowAlarm(5). **ddmTxBiasCurrent—**The current Transmit Bias Current of the SFP/XFP in 10s of milli-Amperes (mA).				
69	ddmTxPowerThresholdViolated	ifIndex ddmNotificationType ddmTxOutputPower	port	This trap is sent when an SFP/XFP/SFP+ Tx output power has crossed any threshold or reverted from previous threshold violation for a port represented by ifIndex. It also provides the current real-time value of SFP/XFP/SFP+ Tx output power.
ifIndex—The interface index. **ddmNotificationType—**The trap type for monitored DDM parameters (clearViolation(1), highAlarm(2), highWarning(3), lowWarning(4), lowAlarm(5). **ddmTxOutputPower—**The current Output Power of the SFP/XFP in 10s of milli-Watts (mW).				
70	ddmRxPowerThresholdViolated	ifIndex, ddmNotificationType ddmRxOpticalPower	port	This trap is sent when an SFP/XFP/SFP+ Rx optical power has crossed any threshold or reverted from previous threshold violation for a port represented by ifIndex. It also provides the current real-time value of SFP/XFP/SFP+ Rx optical power
ifIndex—The interface index. **ddmNotificationType—**The trap type for monitored DDM parameters (clearViolation(1), highAlarm(2), highWarning(3), lowWarning(4), lowAlarm(5). **ddmRxOpticalPower—**The current Received Optical Power of the SFP/XFP in 10s of milli-Watts (mW).				
71	webMgtServerErrorTrap	webMgtServerError	webmgt	This trap is sent when the Web Management server goes into error state after crashing twice within a minute.
webMgtServerError—Error code string when WebView Server is in error status. Format is 'Error Num: {Number}. {String message}.' where {Number} is an integer representing the error code and {String message} is the error string message.				
72	multiChassisIpcVlanUp	multiChassisTrapIpcVlan	multi-chassis	Indicates the operational status for the multi-chassis communication VLAN is Up.
multiChassisTrapIpcVlan—The multi-chassis IPC VLAN number.				
73	multiChassisIpcVlanDown	multiChassisTrapIpcVlan	multi-chassis	Indicates the operational status for the multi-chassis communication VLAN is Down.

No.	Trap Name	Objects	Family	Description
multiChassisTrapIpcVlan—The multi-chassis IPC VLAN number.				
74	multiChassisMisconfigurationFailure	multiChassis-TrapFailure	multi-chas-sis	This trap is sent when there is an MCM misconfiguration (e.g., inconsistent chassis ID or IPC VLAN).
multiChassisTrapFailure—Indicates multi-chassis failure.				
75	multiChassisHelloIntervalConsisFail-ure	multiChassis-TrapFailure	multi-chas-sis	This trap is sent when there is an MCM Hello Interval consistency falure.
multiChassisTrapFailure—Indicates multi-chassis failure.				
76	multiChassisStpModeConsisFailure	multiChassis-TrapFailure	multi-chas-sis	This trap is sent when ther is an STP mode consistency falure.
multiChassisTrapFailure—Indicates multi-chassis failure.				
77	multiChassisStpPathCostModeConsis-Failure	multiChassis-TrapFailure	multi-chas-sis	This trap is sent when ther is an STP path cost mode consistency falure.
multiChassisTrapFailure—Indicates multi-chassis failure..				
78	multiChassisVflinkStatusConsisFail-ure	multiChassis-TrapFailure	multi-chas-sis	This trap is sent when there is an MCM Virtual Fabric Link status consistency falure
multiChassisTrapFailure—Indicates multi-chassis failure.				
79	multiChassisStpBlockingStatus	multiChassis-TrapStpBlock-ingVlanList	multi-chas-sis	This trap is sent when the STP status for some VLANs on the Virtual Fabric Link is in blocking state.
multiChassisTrapStpBlockingVlanList—The VLANS with STP in the Blocking State. Up to 16 VLANs are displayed, seperated by comas.				
80	multiChassisLoopDetected	multiChassis-TrapFailure	multi-chas-sis	This trap is sent when a loop is detected.
multiChassisTrapFailure—Indicates multi-chassis failure.				
81	multiChassisHelloTimeout	multiChassis-TrapFailure	multi-chas-sis	This trap is sent when the Hellow Timer expires.
multiChassisTrapFailure—Indicates multi-chassis failure.				
82	multiChassisVflinkDown	multiChassis-TrapFailure	multi-chas-sis	This trap is sent when the Virtual Fabric Link goes down.
multiChassisTrapFailure—Indicates multi-chassis failure.				
83	multiChassisVFLMemberJoinFailure	multiChassis-TrapVFL, mul-tiChassisTrapV FLMember-Port, multiChas-sisTrapDiagnisti c	multi-chas-sis	This trap is sent when a port configured as virtual fabric member is unable to join the vir-tual fabric link

No.	Trap Name	Objects	Family	Description
multiChassisTrapVFL—The multi-chassis Virtual Fabric Link interface. **multiChassisTrap VFLMemberPort**—The multi-chassis VFL member port number. **multiChassisTrapDiagnistic**—The reason a port configured as virtual-fabric member is unable to join the virtual-fabric link - 1. Duplex Mode, 2. Speed.				
84	alaDHLVlanMoveTrap	alaDHLSessionID, alaDHLPortFrom, alaDHLPortTo, alaDHLVlanMoveReason	vlan	When linkA or linkB goes down or comes up and both ports are are part of some vlan-map, this trap is sent to the Management Entity, with the DHL port information.
alaDHLSessionID—The DHL Session ID for which alaDHLVlanMoveTrap needs to be sent to the Management Entity. **alaDHLPortFrom—**The the port, either linkA or linkB, from whichvlan-mapped vlans have joined to other port due to linkUp or linkDown as specified by alaDHLVlanMoveReason. **alaDHLPortTo—**The the port, either linkA or linkB, to which vlan-mapped vlans have joined from other port due to linkUp or linkDown as specified by alaDHLVlanMoveReason **alaDHLVlanMoveReason—**The reason for Vlan Movement from one port to another port.				
85	alaDhcpClientAddressAddTrap	alaDhcpClientAddress	udp relay	This trap is sent when a new IP address is assigned to DHCP Client interface.
alaDhcpClientAddress—The current IP address of the DHCP client.				
86	alaDhcpClientAddressExpiryTrap	ialaDhcpClientAddress	ip-helper	This trap is sent when the lease time expires or when a DHCP client unable to renew/rebind an IP address.
alaDhcpClientAddress—The current IP address of the DHCP client.				
87	alaDhcpClientAddressModifyTrap	alaDhcpClientAddress, alaDhcpClientNewAddress	ip-helper	This trap is sent when the DHCP client unable to obtain the existing IP address and a new IP address is assigned to the DHCP client.
alaDhcpClientAddress—The current IP address of the DHCP client. **alaDhcpClientNewAddress—**The new IP address assigned to the DHCP client.				
88	vRtrIsisDatabaseOverload	vRtrIsisSystemLeve lisisSysL1 State isisSysL2 State	isis	This trap is sent when the system enters or leaves the Overload state.
vRtrIsisSystemLevel—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm. **isisSysL1State**—Level 1 Routing (1) **isisSysL2State**—Level 2 Routing (2)				

No.	Trap Name	Objects	Family	Description
89	vRtrIsisManualAddressDrops	isisManAr-eaAddrExist-State	isis	This trap is sent when one of the manual area addresses assigned to this system is ignored when computing routes. The object vRtrIsisManAreaAddrExistState describes the area that has been dropped. This trap is edge triggered, and should not be regenerated until an address that was used in the previous computation has been dropped.
isisManAreaAddrExistState—The area ID that was ignored when computing routes.				
90	vRtrIsisCorruptedLSPDetected	vRtrIsisSystem-Level vRtrIsisTrapL-SPID	isis	This trap is sent when an LSP that was stored in memory has become corrupted. The LSP ID is forwarded. The ID may be known, but in some implementations there is a chance that the ID itself will be corrupted.
vRtrIsisSystemLevel—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm. **vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU.				
91	vRtrIsisMaxSeqExceedAttempt	vRtrIsisSys-temLevel vRtrIsisTrapL-SPID	isis	This trap is sent when the sequence number on an LSP wraps the 32 bit sequence counter.
vRtrIsisSystemLevel—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm. **vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU.				
92	vRtrIsisIDLenMismatch	vRtrIsis-FieldLen vRtrIsisIfIndex vRtrIsisPDUF-ragment	isis	This trap is sent when when a PDU with a different System ID Length is received. The notification includes the index to identify the circuit for the PDU and the header of the PDU, which may help a network manager identify the source of the problem.
vRtrIsisFieldLen—The System ID Field length. **vRtrIsisIfIndex**—The ISIS interface on which the PDU was received. **vRtrIsisPDUFragment**—The first 64 bytes of a PDU that triggered the trap.				

No.	Trap Name	Objects	Family	Description
93	vRtrIsisMaxAreaAddrsMismatch	vRtrIsisMax-AreaAddress, vRtrIsisIfIndex vRtrIsisPDUF-ragment	isis	This trap is sent when a PDU with a different Maximum Area Addresses value is recieved. The notification includes the header of the packet, which may help a network manager identify the source of the problem.
vRtrIsisMaxAreaAddress—The maximum number of area addresses in the PDU. **vRtrIsisIfIndex**—The ISIS interface on which the PDU was received. **vRtrIsisPDUFragment**—The first 64 bytes of a PDU that triggered the trap.				
94	vRtrIsisOwnLSPPurge	vRtrIsisIfIndex, vRtrIsisTrapL-SPID vRtrIsisSystem-Level	isis	This trap is sent when sent when a PDU is received with the system ID and zero age. This notification includes the circuit Index if available, which may help a network manager identify the source of theproblem.
vRtrIsisIfIndex—The ISIS interface on which the PDU was received. **vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU. **vRtrIsisSystemLevel**—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm.				
95	vRtrIsisSequenceNumberSkip	vRtrIsisTrapL-SPID vRtrIsisIfIndex vRtrIsisSystem-Level	isis	If an LSP without System ID and different contents is received, the LSP may be reissued with a higher sequence number. If two Intermediate Systems are configured with the same System ID, the sequence number is increased and this notification is sent.
vRtrIsisTrapLSPID—An Octet String that uniquely identifies a Link State PDU. **vRtrIsisIfIndex**—The ISIS interface on which the PDU was received. **vRtrIsisSystemLevel**—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm.				
96	vRtrIsisAutTypeFail	vRtrIsisSystem-Level, vRtrIsisPDUF-ragment, vRtrIsisIfIndex	isis	This trap is sent when a PDU with the wrong authentication type is received. The notification includes the header of the packet, which may help a network manager identify the source of the problem.
vRtrIsisSystemLevel—Identifies the level to which the notification applies.Routing within an area is referred. to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm. **vRtrIsisPDUFragment**—Contains up to the first 64 bytes of a PDU that triggered the trap. **vRtrIsisIfIndex**—The ISIS interface on which the PDU was received.				

No.	Trap Name	Objects	Family	Description
97	vRtrIsisAuthFail	vRtrIsisSystem-Level, vRtrIsisPDUF-ragment, vRtrIsisIfIndex	isis	This trap is sent when a PDU with incorrent authentication information is received. The notification includes the header of the packet, which may help a network manager identify the source of the problem.
	vRtrIsisSystemLevel—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm. **vRtrIsisPDUFragment**—Contains up to the first 64 bytes of a PDU that triggered the trap. **vRtrIsisIfIndex**—The ISIS interface on which the PDU was received..			
98	vRtrIsisVersionSkew	vRtrIsisProto-colVersion vRtrIsisSystem-Level vRtrIsisPDUF-ragment vRtrIsisIfIndex	isis	This trap is sent when a Hello PDU is received from an IS running a different version of the protocol. This notification includes the header of the packet, which may help a network manager identify the source of the problem.
	vRtrIsisProtocolVersion—The PDU protocol version. **vRtrIsisSystemLevel**—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm. **vRtrIsisPDUFragment**—Contains up to the first 64 bytes of a PDU that triggered the trap. **vRtrIsisIfIndex**—The ISIS interface on which the PDU was received.			
99	vRtrIsisAreaMismatch	vRtrIsisLSP-Size vRtrIsisSystem-Level vRtrIsisIfIndex vRtrIsisPDUF-ragment	isis	This trap is sent when a Hello PDU from an IS that does not share any area address is received. This notification includes the header of the packet, which may help a network manager identify the source of the confusion.
	vRtrIsisLSPSize—The size of the LSP received. **vRtrIsisSystemLevel**—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm. **vRtrIsisIfIndex**—The ISIS interface on which the PDU was received. **vRtrIsisPDUFragment**—Contains up to the first 64 bytes of a PDU that triggered the trap.			
100	vRtrIsisRejectedAdjacency	vRtrIsisSys-temLevel vRtrIsisIfIndex	isis	This trap is sent when a Hello PDU is received from an IS, but an adjacency is not established due to a lack of resources.
	vRtrIsisSystemLevel—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm. **vRtrIsisIfIndex**—The ISIS interface on which the PDU was received.			

No.	Trap Name	Objects	Family	Description
101	vRtrIsisLSPTooLargeToPropagate	vRtrIsisLSP-Size vRtrIsisSystem-Level vRtrIsisTrapL-SPID vRtrIsisIfIndex	isis	This trap is sent when an LSP is larger than the Data Link Block Size for a circuit.
vRtrIsisLSPSize—The size of the LSP received. **vRtrIsisSystemLevel**—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm. **vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU. **vRtrIsisIfIndex**—The ISIS interface on which the LSP was received.				
102	vRtrIsisOrigLSPBufSizeMismatch	vRtrIsisOrigi-natingBuffer-Size vRtrIsisSystem-Level vRtrIsisTrapL-SPID vRtrIsisIfIndex	isis	This trap is sent when a Level 1 or 2 LSP is received that is larger than the local value for the originating LSP Buffer Size; or when a Level 1 or 2 LSP is received containing the originating LSP Buffer Size option but the value in the PDU option field does not match the local value for the originating LSP Buffer Size.
vRtrIsisOriginatingBufferSize—The buffer size advertised by the peer. **vRtrIsisSystemLevel**—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm. **vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU. **vRtrIsisIfIndex**—The ISIS interface on which the LSP was received.				
103	vRtrIsisProtoSuppMismatch	vRtrIsisProto-colsSup-ported vRtrIsisSystem-Level vRtrIsisTrapL-SPID vRtrIsisIfIndex	isis	This trap is sent when a non-pseudonode segment 0 LSP is received that has no matching protocols supported. This may be because the system does not generate the field, or because there are no common elements. The list of protocols supported should be included in the notification: it may be empty if the TLV is not supported, or if the TLV is empty.
vRtrIsisProtocolsSupported—The protocols supported by an adjacent system. This may be empty **vRtrIsisSystemLevel**—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm. **vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU. **vRtrIsisIfIndex**—The ISIS interface on which the LSP was received.				

No.	Trap Name	Objects	Family	Description
104	vRtrIsisAdjacencyChange	vRtrIsisSys-temLevel vRtrIsisIfIndex vRtrIsisTrapL-SPID isisISAdjState	isis	This trap is sent when adjacency changes state, entering or leaving state up. The first 6 bytes of the vRtrIsis-TrapLSPID are the SystemID of the adjacent IS. The isisISAd-jState is the new state of the adja-cency.
vRtrIsisSystemLevel—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a sep-arate copy of the basic link-state routing algorithm. **vRtrIsisIfIndex**—The ISIS interface on which the trap was received. **vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU. **isisISAdjState**—The state of the adjacent router.				
105	vRtrIsisCircIdExhausted	vRtrIsisIfIndex	isis	This trap is sent when sent when ISIS cannot be started on a LAN interface because a unique circid could not be assigned due to the exhaustion of the Circuit ID space. This can only happen on broadcast interfaces. When this happens, the interface is marked operationally down. When an operationally up inter-face is deleted, the Circuit ID can be reused by any interface wait-ing to receive a unique Circuit ID.
vRtrIsisIfIndex—The ISIS interface.				
106	vRtrIsisAdjRestartStatusChange	vRtrIsisSys-temLevel vRtrIsisIfIndex vRtrIsisISAd-jRestartStatus	isis	This trap is sent when an adjan-cency's graceful restart status changes.
vRtrIsisSystemLevel—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a sep-arate copy of the basic link-state routing algorithm. **vRtrIsisIfIndex**—The ISIS interface. **vRtrIsisISAdjRestartStatus**—The new graceful restart state of the adjacency.				
107	alaMvrpVlanLimitReachedEvent	alaMvrpMaxV-lanLimit	bridge	This trap is sent when the num-ber of VLANs learned dynami-cally by MVRP reaches the configured limit.
alaMvrpMaxVlanLimit—The the maximum number of dynamic VLANs that can be created on the system by MVRP. If the number of VLANs created by MVRP reaches this limit, the system will prevent MVRP from cre-ating additional VLANs (32 - 4094, Default = 256).				

No.	Trap Name	Objects	Family	Description
108	alaHAVlanClusterPeerMismatch	alaHAVlanClusterId	ha-vlan	This trap is sent when parameteras configured for this cluster ID (Level 1 check) does not match accross the MCLAG peers.
alaHAVlanClusterId—The Cluster ID Number.				
109	alaHAVlanMCPeerMismatch	alaHAVlanClusterId alaHAVlanMultiChassisId alaHAVlanClusterPortIfIndex	ha-vlan	This trap is sent when the cluster parameters are matching on the peers, but MCLAG is not configured or clusters are not in operational state.
alaHAVlanClusterId—The Cluster ID Number. **alaHAVlanMultiChassisId—**The Multi Chassis ID identifying the Multi Chassis Peer. **alaHAVlanClusterPortIfIndex—**The ifindex identifying the cluster port. An ifindex of 1 shall be used for all ports.				
110	alaHAVlanDynamicMAC	alaHAVlanClusterId alaHAVlanClusterInetAddress alaHAVlanClusterMacAddress alaHAVlanClusterPortIfIndex	ha-vlan	The trap is sent when the dynamic MAC is learned on non-server cluster port
alaHAVlanClusterId—The Cluster ID Number. **alaHAVlanClusterInetAddress—**The type of IP address associated with the L3 cluster (e.g., ipv4). **alaHAVlanClusterMacAddress—**The type of ARP resolution used in L3 cluster (static, dynamic, invalid). **alaHAVlanClusterPortIfIndex—**The ifindex identifying the cluster port. An ifindex of 1 shall be used for all ports.				
111	unpMcLagMacIgnored	alaDaUnpMacAddr alaDaUnpSourceIpAddr alaDaUnpNativeVlan alaDaUnpVlan alaDaUnpMCLAGId	da-unp	This trap is sent when a MAC/User is dropped because the VLAN does not exist or UNP is not enabled on the MCLAG..
alaDaUnpMacAddr—The MAC that failed to get configured on peer chassis. **alaDaUnpSourceIpAddr—**The IP address of the MAC that failed to get configured on peer chassis. **alaDaUnpNativeVlan—**The native VLAN of MCLAG on which the MAC ingressed. **alaDaUnpVlan—**The VLAN on which the MAC was classified on the local chassis. **alaDaUnpMCLAGId—**The Link Agg Id for MCLAG..				

No.	Trap Name	Objects	Family	Description
112	unpMcLagConfigInconsistency	alaDaUnpCommandType alaDaUnpName alaDaUnpMacAddr1 alaDaUnpMacAddr2 alaDaUnpIpAddr alaDaUnpIpMask alaDaUnpVlanTag alaDaUnpMCLAGId	da-unp	This trap is sent when a configuration becomes “Out of Sync".
	alaDaUnpCommandType—Indicates which configuration command is out-of-sync: unpConfigCmd (1), macRuleConfigCmd (2), macRangeRuleConfigCmd (3), ipRuleConfigCmd (4), vlanTagRuleConfigCmd (5), authServerUnpConfigCmd (6), authServerTimerConfigCmd (7), dynamicVlanConfigCmd (8), lagConfigCmd (9), dynamicProfileConfigCmd (10). **alaDaUnpName—**Indicates which UNP Profile is out-of-sync. If there is no UNP Profile associated, a zero length string is sent. **alaDaUnpMacAddr1—**The MAC for MAC rule or the lower limit of MAC Range Rule. **alaDaUnpMacAddr2—**The upper limit of MAC Range Rule. **alaDaUnpIpAddr—**The IP address in the IP Rule. **alaDaUnpIpMask—**The IP Mask of the IP address in the IP Rule. **alaDaUnpVlanTag—**The VLAN VLAN Tag Rule. A zero value means it is not applicable. **alaDaUnpMCLAGId—**The Link Agg ID for MCLAG..			
113	multiChassisGroupConsisFailure	multiChassisTrapFailure	mcm	This trap is sent when there is an inconsistency between local and peer chassis group.
	multiChassisTrapFailure—Indicate multi-chassis failure.			
114	multiChassisTypeConsisFailure		mcm	
	xxx—Description.			
115	alaPimNonBidirHello	pimNeighborAdressType, pimNeighboAddress	pim	This trap is sent when a bidir-capable router has received a PIM hello from a non-bidir-capable router. It is generated whenever the counter alaPimsmNonBidirHelloMsgsRcvd is incremented, subject to the rate limit specified by alaPimsmNonBidirHelloNotificationPeriod.
	pimNeighborAdressType—The address type of the PIM neighbor. **pimNeighborAddress—**The primary IP address of the PIM neighbor. The InetAddressType is given by the pimNeighborAddressType object.			
116	dot1agCfmFaultAlarm	dot1agCfmMepHighestPrDefect	802.1AG	This trap is sent when a MEP has a persistent defect condition. A notification (fault alarm) is sent to the management entity with the OID of the MEP that has detected the fault.

No.	Trap Name	Objects	Family	Description
dot1agCfmMepHighestPrDefect—The highest priority defect that has been present since the MEPs Fault Notification Generator State Machine was last in the FNG_RESET state.				
117	alaSaaIPIterationCompleteTrap	alaSaaCtrlOwnerIndex, alaSaaCtrlTestIndex, alaSaaIpResultsTestRunIndex, alaSaaCtrlLastRunResult, alaSaaCtrlLastRunTime	saa	This trap is sent when an IP SAA iteration is completed.
alaSaaCtrlOwnerIndex—The Owner name to identify entries in the table. This is currently not supported and its value will always be the string 'USER'. **alaSaaCtrlTestIndex—**A Unique name to identify the entries in the table. The name is unique across various SNMP users. **alaSaaIpResultsTestRunIndex—**The row entry that reports results for a single OAM test run.The value of this object starts at 1 and can go upto a maximum of alaSaaCtrlMaxHistoryRows. **alaSaaCtrlLastRunResult—**The result of the latest SAA test iteration (Undetermined/Success/Failed/ Aborted). **alaSaaCtrlLastRunTime—**The time the last iteration of the SAA was run.				
118	alaSaaEthIterationCompleteTrap	alaSaaCtrlOwnerIndex, alaSaaCtrlTestIndex, alaSaaEthoamResultsTestRunIndex, alaSaaCtrlLastRunResult, alaSaaCtrlLastRunTime	saa	This trap is sent when when an eth-LB or Eth-DMM SAA iteration is completed.
alaSaaCtrlOwnerIndex—The Owner name to identify entries in the table. This is currently not supported and its value will always be the string 'USER'. **alaSaaCtrlTestIndex—**A Unique name to identify the entries in the table. The name is unique across various SNMP users. **alaSaaEthoamResultsTestRunIndex—**The row entry that reports results for a single Eth-LB/DMM test run. The value of this object starts from 1 and can go upto a maximum of alaSaaCtrlMaxHistoryRows. **alaSaaCtrlLastRunResult—**The result of the latest SAA test iteration (Undetermined/Success/Failed/ Aborted). **alaSaaCtrlLastRunTime—**The time the last iteration of the SAA was run.				
119	alaSaaMacIterationCompleteTrap		saa	This trap is sent when a MAC iteration is complete.
ala—The ?				
120	virtualChassisStatusChange	virtualChassisOperChasId, virtualChassisStatus	virtual chassis	This trap is sent when a chassis status change is detected.
virtualChassisOperChasId—The operational Virtual Chassis ID. **virtualChassisStatus—**The Virtual Chassis status.				

No.	Trap Name	Objects	Family	Description
121	virtualChassisRoleChange	virtualChassisOperChasId, virtualChassisRole	virtual chassis	This trap is sent when a chassis role change is detected.
virtualChassisOperChasId—The operational Virtual Chassis ID. **virtualChassisRole—**The Virtual Chassis role: **unassigned(0):** Initial chassis role and election not complete. **master(1):** Chassis is in master role after election. **slave(2):** Chasis is in slave role after election. **inconsistent(3): C**hassis is not consistent after election.				
122	virtualChassisVflStatusChange	virtualChassisOperChasId, virtualChassisVflIfIndex, virtualChassisVflOperStatus	virtual chassis	This trap is sent when a vflink status change is detected.
virtualChassisOperChasId—The operational Virtual Chassis ID. **virtualChassisVflIfIndex—**The Virtual Fabric Link ID. **virtualChassisVflOperStatus—**The Virtual Fabric Link Operational Status (Up/Down/Disabled).				
123	virtualChassisVflMemberPortStatusCh	virtualChassisOperChasId, virtualChassisVflIfIndex, virtualChassisVflMemberPortIfindex, virtualChassisVflMemberPortOperStatus	virtual chassis	This trap is sent when a vflink member port has a change of status.
virtualChassisOperChasId—The operational Virtual Chassis ID. **virtualChassisVflIfIndex—**The Virtual Fabric Link ID **virtualChassisVflMemberPortIfindex—**The Virtual Fabric Link Member Port ifIndex. **virtualChassisVflMemberPortOperStatus—**Virtual Fabric Link Member Port operational status (Up, Down, Disabled).				
124	virtualChassisVflMemberPortJoinFail	virtualChassisOperChasId, virtualChassisVflIfIndex, virtualChassisVflMemberPortIfindex, virtualChassisDiagnostic	virtual chassis	This trap is sent when a port configured as virtual-fabric member is unable to join the virtual-fabric link.

No.	Trap Name	Objects	Family	Description
virtualChassisOperChasId—The operational Virtual Chassis ID. **virtualChassisVflIfIndex—**The Virtual Fabric Link ID **virtualChassisVflMemberPortIfindex—**The Virtual Fabric Link Member Port ifIndex. **virtualChassisDiagnostic—**Indicates why a port configured as virtual-fabric member is unable to join the virtual-fabric link (Duplex Mode, Speed).				
125	lldpRemTablesChange	lldpStatsRemTablesInserts, lldpStatsRemTablesDeletes, lldpStatsRemTablesDrops, lldpStatsRemTablesAgeouts	lldp	This trap is sent when the value of lldpStatsRemTablelastChangeTime changes. It can be utilized by an NMS to trigger LLDP remote systems table maintenance polls.
lldpStatsRemTablesInserts—The number of times the complete set of information advertised by a particular MSAP has been inserted into tables contained in lldpRemoteSystemsData and lldpExtensions objects. **lldpStatsRemTablesDeletes—**The number of times the complete set of information advertised by a particular MSAP has been deleted from tables contained in lldpRemoteSystemsData and lldpExtensions objects. **lldpStatsRemTablesDrops—**The number of times the complete set of information advertised by a particular MSAP could not be entered into tables contained in lldpRemoteSystemsData and lldpExtensions objects because of insufficient resources. **lldpStatsRemTablesAgeouts—**The number of times the complete set of information advertised by a particular MSAP has been deleted from tables contained in lldpRemoteSystemsData and lldpExtensions objects because the information timeliness interval has expired.				
126	vRtrLdpInstanceStateChange	vRtrLdpGenAdminState, vRtrLdpGenOperState, vRtrLdpInstanceNotifyReasonCode	ldp	This trap is sent when the LDP module changes state either administratively or operationally.
vRtrLdpGenAdminState—The desired administrative state for this LDP instance. **vRtrLdpGenOperState—**The current operational state of this LDP instance. **vRtrLdpInstanceNotifyReasonCode—**The reason for the LDP instance state change (Admin Up, Admin Down, Oper Up, Oper Down)				
127	evbFailedCdcpTlvTrap	evbPortId	evb	This trap is sent when bridge receives a CDCP packet with: - Wrong TLV type, or - Wrong OUI, or - Role is set to Bridge, or - Wrong default channel(scid), or - Incorrect channel number(scid).
evbPortId—The IfIndex that uniquely identifies this port.				
128	evbFailedEvbTlvTrap	evbPortId, ieee8021BridgeEvbVSIVlanId	evb	This trap is sent when bridge receives an EVBTLV packet with: - Wrong TLV type. or - Incorrect TLV length, or - Wrong OUI.

No.	Trap Name	Objects	Family	Description
evbPortId—The IfIndex that uniquely identifies this port**.** **ieee8021BridgeEvbVSIVlanId—**The bridge EVB VSI VLAN.				
129	evbUnknownVsiManagerTrap	evbPortId, ieee8021Bridge EvbSbpPort-Number	evb	This trap is sent when bridge receives a VDP packet with: - Unknown Manager ID type, or - Wrong Manager ID length.
evbPortId—The IfIndex that uniquely identifies this port**.** **ieee8021BridgeEvbSbpPortNumber—**The bridge EVN SBP Port.				
130	evbVdpAssocTlvTrap	evbPortId, ieee8021Bridge EvbSbpPort-Number, ieee8021Bridge EvbVSIID, ieee8021Bridge EvbVSIID-Type, ieee8021Bridge EvbVSIType-Version	evb	This trap is sent when bridge receives an ASSOC TLV in a VDP packet with: - Null VID found and number of entry field is not 1, or - Unknown filter format, - Null VID on De-Assoc TLV type, or - VSI included more than Max number of filter info entries
evbPortId—The IfIndex that uniquely identifies this port**.** **ieee8021BridgeEvbSbpPortNumber—**The EVB port number. **ieee8021BridgeEvbVSIID—**The VSIID that uniquely identifies the VSI in the DCN. **ieee8021BridgeEvbVSIIDType—**The VSIID Type for the VSIID in the DCN: - vsiidIpv4 (1) - vsiidIpv6 (2) - vsiidMAC (3) - vsiidLocal (4) - vsiidUUID (5) **ieee8021BridgeEvbVSITypeVersion—**An integer identifier designating the expected/desired VTID version. The VTID version allows a VSI Manager Database to contain multiple versions of a given VSI Type, allowing smooth migration to newer VSI types. **ieee8021BridgeEvbSbpPortNumber—**The EVB SPB port.				
131	evbCdcpLldpExpiredTrap		evb	This trap is sent when an LLDP Timer expired in bridge. The timer expires when LLDP doesn't not receive CDCP TLV within a specified interval.
132	evbTlvExpiredTrap		evb	This trap is sent when an LLDP Timer expired in bridge. The timer expires when LLDP doesn't not receive EVB TLV within a specified interval.
133	evbVdpKeepaliveExpiredTrap		evb	This trap is sent when a VDP Keep Alive Timer expired in bridge. The timer expires when the bridge doesn't not receive VDP Keep Alive message within a specified interval.

No.	Trap Name	Objects	Family	Description

Index

Symbols

A

B

C

D

E

F

H

I

K

L

M

N

P

R

S

T

U

V

W