

**Part No. 060391-10, Rev. A
May 2014**

OmniSwitch AOS Release 8 Switch Management Guide

Alcatel-Lucent 

www.alcatel-lucent-lucent.com

**This user guide documents AOS Release 8 for the OmniSwitch 6860 and OmniSwitch 6860E.
The functionality described in this guide is subject to change without notice.**

Alcatel, Lucent, Alcatel-Lucent and the Alcatel-Lucent logo are trademarks of Alcatel-Lucent. All other trademarks are the property of their respective owners. The information presented is subject to change without notice. Alcatel-Lucent assumes no responsibility for inaccuracies contained herein.

Copyright © 2014 Alcatel-Lucent. All Rights Reserved.



26801 West Agoura Road
Calabasas, CA 91301
(818) 880-3500 FAX (818) 880-3505

Service & Support Contact Information

North America: 800-995-2696
Latin America: 877-919-9526
EMEA : +800 00200100 (Toll Free) or +1(650)385-2193
Asia Pacific: +65 6240 8484
web: service.esd.alcatel-lucent.com
email: esd.support@alcatel-lucent.com

Contents

| | | |
|------------------|---|------|
| | About This Guide | xiii |
| | Supported Platforms | xiii |
| | Who Should Read this Manual? | xiii |
| | When Should I Read this Manual? | xiii |
| | What is in this Manual? | xiv |
| | What is Not in this Manual? | xiv |
| | How is the Information Organized? | xiv |
| | Documentation Roadmap | xv |
| | Related Documentation | xvii |
| | Technical Support | xvii |
| Chapter 1 | Logging Into the Switch | 1-1 |
| | In This Chapter | 1-1 |
| | Login Specifications | 1-2 |
| | Login Defaults | 1-2 |
| | Quick Steps for Logging Into the Switch | 1-3 |
| | Overview of Switch Login Components | 1-4 |
| | Management Interfaces | 1-4 |
| | Logging Into the CLI | 1-4 |
| | Using the WebView Management Tool | 1-5 |
| | Using SNMP to Manage the Switch | 1-5 |
| | User Accounts | 1-5 |
| | Accessing the micro USB Console Port | 1-6 |
| | Accessing the RS-232 Console Port | 1-7 |
| | Configuring the USB Bluetooth Adapter | 1-8 |
| | Identifying the Current Connection | 1-8 |
| | Setting the EMP Port's IP Address | 1-9 |
| | Modifying the EMP IP Address | 1-9 |
| | Modifying the NVRAM EMP IP Address | 1-9 |
| | Using Telnet | 1-10 |
| | Logging Into the Switch Via Telnet | 1-10 |
| | Starting a Telnet Session from the Switch | 1-10 |
| | Using Secure Shell | 1-11 |
| | Secure Shell Components | 1-11 |
| | Secure Shell Interface | 1-11 |

| | |
|---|------|
| Secure Shell File Transfer Protocol | 1-11 |
| Secure Shell Application Overview | 1-12 |
| Secure Shell Authentication | 1-13 |
| Protocol Identification | 1-13 |
| Algorithm and Key Exchange | 1-13 |
| Authentication Phase | 1-14 |
| Connection Phase | 1-14 |
| Using Secure Shell Public Key Authentication (PKA) | 1-14 |
| Revoking a Key | 1-15 |
| Starting a Secure Shell Session from the OmniSwitch | 1-15 |
| Modifying the Login Banner | 1-16 |
| Modifying the Text Display Before Login | 1-17 |
| Configuring Login Parameters | 1-18 |
| Configuring the Inactivity Timer | 1-18 |
| Enabling the DNS Resolver | 1-19 |
| Enabling the FIPS mode | 1-19 |
| FIPS Specifications | 1-20 |
| Quick Steps for Configuring FIPS mode | 1-20 |
| Verifying Login Settings | 1-22 |
| Chapter 2 | |
| Managing System Files | 2-1 |
| In This Chapter | 2-1 |
| File Management Specifications | 2-2 |
| Switch Administration Overview | 2-3 |
| File Transfer | 2-3 |
| Switch Directories | 2-4 |
| File and Directory Management | 2-5 |
| Directory Commands | 2-7 |
| Determining Your Location in the File Structure | 2-7 |
| Changing Directories | 2-8 |
| Making a New Directory | 2-8 |
| Copying an Existing Directory | 2-8 |
| Removing a Directory and its Contents | 2-9 |
| File Commands | 2-9 |
| Creating or Modifying Files | 2-9 |
| Copy an Existing File | 2-9 |
| Secure Copy an Existing File | 2-9 |
| Move an Existing File or Directory | 2-10 |
| Change File Attribute and Permissions | 2-10 |
| Delete an Existing File | 2-10 |
| Utility Commands | 2-11 |
| Displaying Free Memory Space | 2-11 |
| Performing a File System Check | 2-11 |
| Deleting the Entire File System | 2-12 |
| Loading Software onto the Switch | 2-13 |
| Using the Switch as a Server | 2-13 |

| | | |
|------------------|--|------------|
| | Using the Switch as an FTP Client | 2-14 |
| | Using Secure Shell FTP | 2-15 |
| | Closing a Secure Shell FTP Session | 2-15 |
| | Using TFTP to Transfer Files | 2-15 |
| | Installing Software Licenses | 2-16 |
| | Setting the System Clock | 2-17 |
| | Setting Date and Time | 2-17 |
| | Date | 2-17 |
| | Time Zone | 2-17 |
| | Time | 2-17 |
| | Daylight Savings Time Configuration | 2-18 |
| Chapter 3 | Managing CMM Directory Content | 3-1 |
| | In This Chapter | 3-1 |
| | CMM Specifications | 3-2 |
| | USB Flash Drive Specifications | 3-2 |
| | CMM Files | 3-3 |
| | Available Files | 3-3 |
| | CMM Software Directory Structure | 3-4 |
| | Where is the Switch Running From? | 3-4 |
| | Software Rollback Feature | 3-4 |
| | Software Rollback Configuration Scenarios | 3-5 |
| | Managing Switch Configurations | 3-8 |
| | Rebooting the Switch | 3-8 |
| | Saving the Running Configuration | 3-10 |
| | Rebooting from a Directory | 3-11 |
| | Copying the RUNNING DIRECTORY to the Certified Directory | 3-12 |
| | Show Currently Used Configuration | 3-13 |
| | Show Switch Files | 3-13 |
| | Using the USB Flash Drive | 3-14 |
| | Transferring Files Using a USB Flash Drive | 3-14 |
| | Automatically Copying Code Using a USB Flash Drive | 3-14 |
| | Disaster Recovery Using a USB Flash Drive | 3-15 |
| | Displaying CMM Conditions | 3-16 |
| Chapter 4 | Using the CLI | 4-1 |
| | CLI Specifications | 4-2 |
| | CLI Overview | 4-2 |
| | Online Configuration | 4-2 |
| | Offline Configuration Using Configuration Files | 4-2 |
| | Command Entry Rules and Syntax | 4-3 |
| | Text Conventions | 4-3 |
| | Using “Show” Commands | 4-4 |
| | Using the “No” Form | 4-4 |
| | Partial Keyword Completion | 4-4 |
| | Partial Keyword Abbreviation | 4-4 |

| | |
|--|------------|
| Command Help | 4-5 |
| Recalling the Previous Command Line | 4-5 |
| Inserting Characters | 4-6 |
| Command History | 4-6 |
| Logging CLI Commands and Entry Results | 4-7 |
| Enabling Command Logging | 4-7 |
| Disabling Command Logging | 4-7 |
| Viewing the Current Command Logging Status | 4-8 |
| Viewing Logged CLI Commands and Command Entry Results | 4-8 |
| Customizing the Screen Display | 4-9 |
| Changing the Screen Size | 4-9 |
| Changing the CLI Prompt | 4-9 |
| Verifying CLI Usage | 4-10 |
| Chapter 5 Working With Configuration Files | 5-1 |
| In This Chapter | 5-1 |
| Configuration File Specifications | 5-2 |
| Tutorial for Creating a Configuration File | 5-2 |
| Quick Steps for Applying Configuration Files | 5-4 |
| Setting a File for Immediate Application | 5-4 |
| Setting an Application Session for a Date and Time | 5-4 |
| Setting an Application Session for a Specified Time Period | 5-5 |
| Configuration Files Overview | 5-6 |
| Applying Configuration Files to the Switch | 5-6 |
| Verifying a Timed Session | 5-6 |
| Cancelling a Timed Session | 5-7 |
| Configuration File Error Reporting | 5-7 |
| Setting the Error File Limit | 5-7 |
| Syntax Checking | 5-7 |
| Text Editing on the Switch | 5-8 |
| Invoke the “Vi” Editor | 5-8 |
| Creating Snapshot Configuration Files | 5-9 |
| Snapshot Feature List | 5-9 |
| User-Defined Naming Options | 5-10 |
| Editing Snapshot Files | 5-10 |
| Verifying File Configuration | 5-12 |
| Chapter 6 Managing Switch User Accounts | 6-1 |
| In This Chapter | 6-1 |
| User Database Specifications | 6-2 |
| User Account Defaults | 6-2 |
| Overview of User Accounts | 6-4 |
| Startup Defaults | 6-4 |
| Quick Steps for Network Administrator User Accounts | 6-6 |
| Default User Settings | 6-7 |

| | |
|---|------|
| Account and Password Policy Settings | 6-7 |
| How User Settings Are Saved | 6-7 |
| Creating a User | 6-8 |
| Removing a User | 6-8 |
| User-Configured Password | 6-8 |
| Configuring Password Policy Settings | 6-10 |
| Setting a Minimum Password Size | 6-10 |
| Configuring the Username Password Exception | 6-10 |
| Configuring Password Character Requirements | 6-11 |
| Configuring Password Expiration | 6-11 |
| Default Password Expiration | 6-11 |
| Specific User Password Expiration | 6-12 |
| Configuring the Password History | 6-12 |
| Configuring the Minimum Age for a Password | 6-12 |
| Configuring Global User Lockout Settings | 6-13 |
| Configuring the User Lockout Window | 6-13 |
| Configuring the User Lockout Threshold Number | 6-13 |
| Configuring the User Lockout Duration Time | 6-14 |
| Manually Locking and Unlocking User Accounts | 6-14 |
| Configuring Privileges for a User | 6-15 |
| Setting Up SNMP Access for a User Account | 6-16 |
| SNMP Access Without Authentication/Encryption | 6-16 |
| SNMP Access With Authentication/Encryption | 6-17 |
| Removing SNMP Access From a User | 6-17 |
| Multiple User Sessions | 6-18 |
| Listing Other User Sessions | 6-18 |
| Listing Your Current Login Session | 6-19 |
| Terminating Another Session | 6-19 |
| Verifying the User Configuration | 6-20 |
| Chapter 7 | |
| Managing Switch Security | 7-1 |
| In This Chapter | 7-1 |
| Switch Security Defaults | 7-2 |
| Switch Security Overview | 7-3 |
| Authenticated Switch Access | 7-4 |
| AAA Servers—RADIUS or LDAP | 7-4 |
| Interaction With the User Database | 7-4 |
| Configuring Authenticated Switch Access | 7-6 |
| Quick Steps for Setting Up ASA | 7-7 |
| Setting Up Management Interfaces for ASA | 7-9 |
| Enabling Switch Access | 7-9 |
| Configuring the Default Setting | 7-10 |
| Configuring Accounting for ASA | 7-11 |
| Verifying the ASA Configuration | 7-12 |

| | | |
|------------------|---|------|
| Chapter 8 | Using WebView | 8-1 |
| | In This Chapter | 8-1 |
| | WebView Specifications | 8-2 |
| | WebView CLI Defaults | 8-2 |
| | Browser Setup | 8-2 |
| | WebView CLI Commands | 8-3 |
| | Enabling/Disabling WebView | 8-3 |
| | Changing the HTTP Port | 8-3 |
| | Enabling/Disabling SSL | 8-3 |
| | Changing the HTTPS Port | 8-3 |
| | Quick Steps for Setting Up WebView | 8-4 |
| | WebView Overview | 8-4 |
| | WebView Page Layout | 8-4 |
| | Banner | 8-5 |
| | Toolbar | 8-5 |
| | Feature Options | 8-5 |
| | View/Configuration Area | 8-5 |
| Chapter 9 | Using SNMP | 9-1 |
| | In This Chapter | 9-1 |
| | SNMP Specifications | 9-2 |
| | SNMP Defaults | 9-2 |
| | Quick Steps for Setting Up An SNMP Management Station | 9-3 |
| | Quick Steps for Setting Up Trap Filters | 9-4 |
| | Filtering by Trap Families | 9-4 |
| | Filtering by Individual Traps | 9-5 |
| | SNMP Overview | 9-6 |
| | SNMP Operations | 9-6 |
| | Using SNMP for Switch Management | 9-7 |
| | Setting Up an SNMP Management Station | 9-7 |
| | SNMP Versions | 9-7 |
| | SNMPv1 | 9-7 |
| | SNMPv2 | 9-8 |
| | SNMPv3 | 9-8 |
| | Using SNMP For Switch Security | 9-9 |
| | Community Strings (SNMPv1 and SNMPv2) | 9-9 |
| | Configuring Community Strings | 9-9 |
| | Encryption and Authentication (SNMPv3) | 9-10 |
| | Configuring Encryption and Authentication | 9-10 |
| | Setting SNMP Security | 9-11 |
| | Working with SNMP Traps | 9-12 |
| | Trap Filtering | 9-12 |
| | Filtering by Trap Families | 9-12 |
| | Filtering By Individual Trap | 9-12 |

| | |
|--|-------|
| Authentication Trap | 9-13 |
| Trap Management | 9-13 |
| Replaying Traps | 9-13 |
| Absorbing Traps | 9-13 |
| Sending Traps to WebView | 9-13 |
| SNMP MIB Information | 9-14 |
| MIB Tables | 9-14 |
| MIB Table Description | 9-14 |
| Verifying the SNMP Configuration | 9-15 |
| Chapter 10 | |
| Web Services, CLI Scripting, and OpenFlow | 10-1 |
| In This Chapter | 10-1 |
| Web Services Specifications | 10-2 |
| Web Services Overview | 10-2 |
| Representational State Transfer (REST) | 10-2 |
| REST Characteristics | 10-2 |
| REST Verbs | 10-3 |
| Web Service routing | 10-3 |
| Security | 10-3 |
| AOS REST Implementation | 10-3 |
| Output format | 10-4 |
| Caching | 10-4 |
| Web Services REST Examples | 10-5 |
| Query Structure | 10-5 |
| JSON or XML | 10-5 |
| Response Elements | 10-5 |
| Login Example | 10-6 |
| Logout Example | 10-7 |
| Create Table Entry Example - VLAN | 10-8 |
| Create Table Entry Example - IP Interface | 10-9 |
| Modify Table Entry Example - VLAN | 10-10 |
| Modify Table Entry Example - Interface Speed | 10-11 |
| Delete Table Entry Example | 10-12 |
| Query Table Info Example | 10-13 |
| CLI Example | 10-14 |
| Using Python | 10-15 |
| Library Use | 10-15 |
| PYTHON APIs - Quick Reference | 10-17 |
| CLI Scripting | 10-20 |
| Quoting and escaping | 10-20 |
| Variables and functions | 10-21 |
| Variables | 10-21 |
| Functions | 10-21 |
| CLI Tools | 10-23 |
| awk | 10-23 |
| OpenFlow Specifications | 10-25 |
| OpenFlow Agent Overview | 10-26 |

| | | |
|-------------------|---|-------------|
| | OpenFlow Logical | 10-26 |
| | OpenFlow Normal Mode | 10-26 |
| | OpenFlow Hybrid (API) Mode | 10-26 |
| | Quick Steps to Configure OpenFlow Agent | 10-27 |
| Chapter 11 | Configuring Network Time Protocol (NTP) | 11-1 |
| | In This Chapter | 11-1 |
| | NTP Specifications | 11-2 |
| | NTP Defaults Table | 11-2 |
| | NTP Quick Steps | 11-3 |
| | NTP Overview | 11-5 |
| | Stratum | 11-6 |
| | Using NTP in a Network | 11-6 |
| | Authentication | 11-8 |
| | Configuring NTP | 11-9 |
| | Configuring the OmniSwitch as a Client | 11-9 |
| | NTP Servers | 11-10 |
| | Using Authentication | 11-12 |
| | Verifying NTP Configuration | 11-13 |
| Chapter 12 | Managing Automatic Remote Configuration Download | 12-1 |
| | In This Chapter | 12-1 |
| | Automatic Remote Configuration Specifications | 12-2 |
| | Automatic Remote Configuration Defaults | 12-3 |
| | Quick Steps for Automatic Remote Configuration | 12-4 |
| | Overview | 12-5 |
| | Basic Operation | 12-5 |
| | Network Components | 12-6 |
| | Information Provided by DHCP Server | 12-6 |
| | Information Provided by Instruction File | 12-6 |
| | File Servers and Download Process | 12-7 |
| | LED Status | 12-7 |
| | Interaction With Other Features | 12-8 |
| | UDP/DHCP Relay | 12-8 |
| | 802.1Q | 12-8 |
| | LLDP | 12-8 |
| | Dynamic Link Aggregation (LACP) | 12-8 |
| | Automatic Remote Configuration Download Process | 12-9 |
| | Process Illustration | 12-10 |
| | Additional Process Notes | 12-11 |
| | Download Component Files | 12-12 |
| | Instruction File | 12-12 |
| | Instruction File Syntax | 12-13 |
| | Instruction File Usage Guidelines | 12-14 |

| | | |
|-------------------|--|-------------|
| | Firmware Upgrade Files | 12-14 |
| | Bootup Configuration File | 12-14 |
| | Debug Configuration File | 12-14 |
| | Script File | 12-15 |
| | Script File Usage Guidelines | 12-15 |
| | DHCP Client Auto-Configuration Process | 12-16 |
| | Nearest-Edge Mode Operation | 12-17 |
| | LACP Auto Detection and Automatic Link Aggregate Association | 12-19 |
| | Troubleshooting | 12-21 |
| | Error Resolution | 12-21 |
| | Server Connection Failure and File Download Errors | 12-21 |
| | Error Description Table | 12-22 |
| | Script File Errors | 12-22 |
| | Error Description Table | 12-23 |
| Chapter 13 | Configuring Virtual Chassis | 13-1 |
| | In This Chapter | 13-2 |
| | Virtual Chassis Specifications | 13-3 |
| | Virtual Chassis Default Values | 13-4 |
| | Quick Steps for Automatically Configuring A Virtual Chassis | 13-5 |
| | Virtual Chassis Overview | 13-6 |
| | Virtual Chassis Concepts and Components | 13-6 |
| | Virtual Chassis - Boot-Up | 13-7 |
| | Startup Error Mode | 13-8 |
| | Master/Slave Election | 13-8 |
| | Virtual Chassis - Redundancy | 13-8 |
| | Virtual Chassis Split Protection (VCSP) | 13-8 |
| | Managing A Virtual Chassis | 13-9 |
| | Chassis ID Assignment | 13-9 |
| | Slave Chassis-id Assignment | 13-9 |
| | Virtual Chassis Split Protection (VCSP) | 13-12 |
| | VCSP Key Components and Terms | 13-12 |
| | Basic Operation | 13-13 |
| | Protection States | 13-13 |
| | VC Split Recovery | 13-13 |
| | Virtual Chassis - ISSU | 13-15 |
| | Virtual Chassis - Upgrading with ISSU | 13-15 |
| | Interaction with Other Features | 13-17 |
| | VCSP | 13-17 |
| | Configuring Virtual Chassis | 13-18 |
| | Virtual Chassis Configuration Guidelines | 13-18 |
| | General | 13-18 |
| | Chassis Identifier | 13-18 |
| | Virtual Chassis Group Identifier | 13-19 |
| | Virtual Fabric Link (VFL) | 13-19 |

| | | |
|-------------------|---|---------|
| | Control VLAN | 13-19 |
| | Configuring the Chassis Identifier | 13-20 |
| | Duplicate chassis identifier | 13-20 |
| | Configuring the Virtual Chassis Group Identifier | 13-20 |
| | Configuring the Hello Interval | 13-20 |
| | Configuring the Control VLAN | 13-21 |
| | Configuring Virtual Chassis EMP IP Addresses | 13-21 |
| | Configuring the Chassis EMP IP Address | 13-21 |
| | Configuring the Virtual Chassis EMP IP Address | 13-21 |
| | Controlled Shutdown of a Virtual Chassis Participant Switch | 13-22 |
| | Displaying Virtual Chassis Configuration and Status | 13-23 |
| Appendix A | Software License and Copyright Statements | A-1 |
| | Alcatel-Lucent License Agreement | A-1 |
| | ALCATEL-LUCENT SOFTWARE LICENSE AGREEMENT | A-1 |
| | Third Party Licenses and Notices | A-4 |
| Appendix B | SNMP Trap Information | B-1 |
| | SNMP Traps Table | B-2 |
| | Index | Index-1 |

About This Guide

This *OmniSwitch AOS Release 8 Network Configuration 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 6860 and OmniSwitch 6860E 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 *Hardware 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 AOS Release 8 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 6860/6860E Hardware Users Guide*
Release Notes

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 6860/6860E Hardware Users Guide*
OmniSwitch AOS Release 8 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 *Hardware Guide*. This guide provides 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 8 Network Configuration Guide*
OmniSwitch AOS Release 8 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. The *OmniSwitch AOS Release 8 Network Configuration Guide* contains overview information, procedures, and examples on how standard networking technologies are configured on the OmniSwitch.

The *OmniSwitch AOS Release 8 Advanced Routing Configuration Guide* includes configuration information for networks using advanced routing technologies (OSPF and BGP) and multicast routing protocols (DVMRP and PIM-SM).

Anytime

The *OmniSwitch AOS Release 8 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 6860/6860E Hardware Users Guides*
Complete technical specifications and procedures for all OmniSwitch chassis, power supplies, fans, and Network Interface (NI) modules.
- *OmniSwitch AOS Release 8 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 8 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 8 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 AOS Release 8 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 8 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 AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- [“Quick Steps for Logging Into the Switch” on page 1-3](#)
- [“Accessing the micro USB Console Port” on page 1-6](#)
- [“Accessing the RS-232 Console Port” on page 1-7](#)
- [“Configuring the USB Bluetooth Adapter” on page 1-8](#)
- [“Setting the EMP Port’s IP Address” on page 1-9](#)
- [“Using Telnet” on page 1-10](#)
- [“Using Secure Shell” on page 1-11](#)
- [“Modifying the Login Banner” on page 1-16](#)
- [“Configuring Login Parameters” on page 1-18](#)
- [“Enabling the DNS Resolver” on page 1-19](#)
- [“Enabling the FIPS mode” on page 1-19](#)

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” |

| For more information about... | See... |
|---|--|
| Creating user accounts directly on the switch | Chapter 6, “Managing Switch User Accounts” |
| Using the CLI | Chapter 4, “Using the CLI” |
| 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 6860, 6860E |
| EMP port access | OmniSwitch 6860E models only |
| Login Methods | Telnet, SSH, HTTP, SNMP |
| Number of concurrent Telnet sessions | 6 |
| 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 will guide you through connecting to the switch using the micro USB console connection.

- 1 Download and install the USB to UART device driver on your laptop/device. See [“Accessing the micro USB Console Port” on page 1-6](#).
- 2 Connect the OmniSwitch to the USB port of your device using the included micro USB to USB cable.
- 3 The OmniSwitch will be recognized as a new device and assigned a COM port.
- 4 Use your terminal emulation program to assign the OmniSwitch to the appropriate COM port.
- 5 At the login prompt, enter the default **admin** as the username and **switch** as the password or any valid username and password. The switch’s welcome banner will display, followed by the CLI prompt.

```
Welcome to the Alcatel-Lucent OS6860 8.1.1.R01, April 05, 2014.
```

```
Copyright(c), 1994-2014 Alcatel-Lucent. All Rights reserved.
```

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

```
->
```

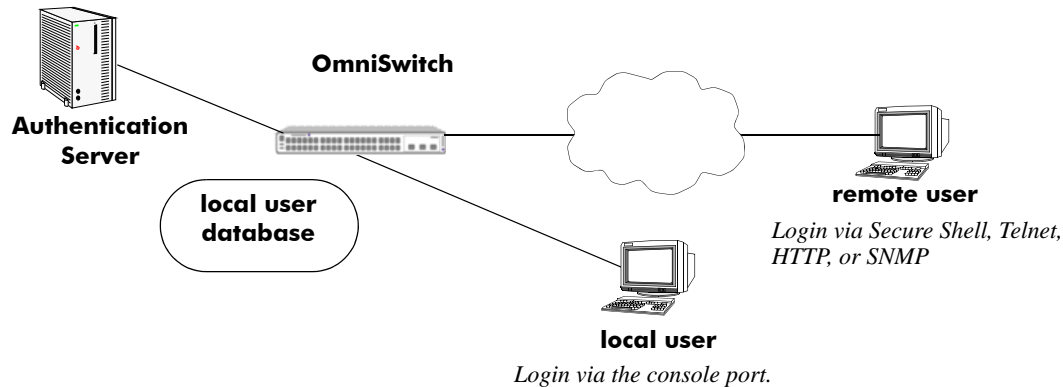
You are now logged into the CLI. For information about changing the welcome banner, see [“Modifying the Login Banner” on page 1-16](#).

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 **system fips admin-state** 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 [“Accessing the micro USB Console Port” on page 1-6](#) or [“Accessing the RS-232 Console Port” on page 1-7](#).
- **Bluetooth port**—A direct connection to the switch using Bluetooth. The console port is always enabled for the default user account, see [“Configuring the USB Bluetooth Adapter” on page 1-8](#).
- **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 1-9](#)
- **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 1-10](#).
- **Secure Shell**—Any standard Secure Shell client may be used for remote login to the switch. See [“Using Secure Shell” on page 1-11](#).

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 8 Network Configuration Guide*.

Accessing the micro USB Console Port

The following procedure is used for accessing the switch using the micro USB console connection.

- 1 Download and install the USB to UART device driver from the following location:

<http://www.silabs.com/products/mcu/pages/usbtouartbridgevcpcdrivers.aspx>

- 2 Connect the OmniSwitch to the USB port of your device using the included micro USB to USB cable.
- 3 The OmniSwitch will be recognized as a new USB device and assigned a COM port.
- 4 Use your terminal emulation program to assign the OmniSwitch to the appropriate COM port.

Note: Each switch will be seen as a new USB device and assigned a different COM port. Use your terminal emulation program to switch between COM ports as required.

- 5 At the login prompt, enter the default **admin** as the username and **switch** as the password or any valid username and password. The switch's welcome banner will display, followed by the CLI prompt.

```
Welcome to the Alcatel-Lucent OS6860 8.1.1.R01, April 05, 2014.
```

```
Copyright(c), 1994-2014 Alcatel-Lucent. All Rights reserved.
```

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

```
->
```

The console port default settings are **9600 baud, 8 data bits, no parity, 1 stop bit, no flow control**. If you wish to modify the default serial connection settings 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.

Accessing the RS-232 Console Port

The following procedure is used for accessing the switch using the RS-232 console connection.

1 Connect the OmniSwitch to the serial port of your device using the micro USB to RJ-45 adapter.

Note: This adapter is not included with the switch but can be ordered separately.

2 Use your terminal emulation program to assign the OmniSwitch to the appropriate COM port.

3 At the login prompt, enter the default **admin** as the username and **switch** as the password or any valid username and password. The switch's welcome banner will display, followed by the CLI prompt.

```
Welcome to the Alcatel-Lucent OS6860 8.1.1.R01, April 05, 2014.
```

```
Copyright(c), 1994-2014 Alcatel-Lucent. All Rights reserved.
```

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

The console port default settings are **9600 baud, 8 data bits, no parity, 1 stop bit, no flow control**. If you wish to modify the default serial connection settings 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.

Configuring the USB Bluetooth Adapter

The following procedure is used for accessing the OmniSwitch using a USB Bluetooth adapter.

- 1 Enable Bluetooth connectivity on the OmniSwitch using the **bluetooth** command.
- 2 Insert the USB Bluetooth adapter into the USB port on the OmniSwitch.
- 3 The OmniSwitch will begin advertising and can now be discovered.
- 4 Once the OmniSwitch is discovered it will be assigned a COM port.

Note: Each switch will be seen as a new bluetooth device and assigned a different COM port. Use your terminal emulation program to switch between COM ports as required.

- 5 Use your terminal emulation program to assign the OmniSwitch to the appropriate COM port.
- 6 At the login prompt, enter the default **admin** as the username and **switch** as the password or any valid username and password. The switch's welcome banner will display, followed by the CLI prompt.

```
Welcome to the Alcatel-Lucent OS6860 8.1.1.R01, April 05, 2014.
```

```
Copyright(c), 1994-2014 Alcatel-Lucent. All Rights reserved.
```

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

Note: Refer to the release notes for a list of validated Bluetooth adapters.

Identifying the Current Connection

When configuring multiple switches using bluetooth it may be difficult to determine which switch has the active bluetooth connection. Issuing the **show me** command will cause the chassis ID LED of the active connection to blink for 10 seconds.

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 of the switch stored in the **vcboot.cfg** file (EMP-CHAS1).
- The EMP IP address of the switch stored in NVRAM. (EMP-CMMA-CHAS1)

Only the EMP IP address stored in the **vcboot.cfg** file is required for remote access to the switch. The NVRAM IP address is designed for chassis-based products with multiple CMMs and is not required for the OS6860. If multiple EMP IP addresses are configured:

- All the EMP IP addresses must be in the same subnet.
- Each of the EMP IP addresses must be unique.
- The NVRAM IP address will be retained even if the switch configuration is erased.

Modifying the EMP IP Address

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

```
-> ip interface emp address 10.255.1.1 mask 255.255.0.0
```

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

Modifying the NVRAM EMP IP Address

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

```
-> modify boot parameters

Boot > boot empipaddr 10.255.1.50
Boot > boot empmasklength 16
Boot > show

EMP IP Address: 10.255.1.50/16
(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.
- For additional information on EMP IP addresses on multiple switches configured as a Virtual Chassis refer to the [“Configuring Virtual Chassis EMP IP Addresses” on page 13-21](#).

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 **system fips admin-state** 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.

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 6860
Software Version 8.1.1.R01, April 05, 2014.

Copyright(c), 1994-2014 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

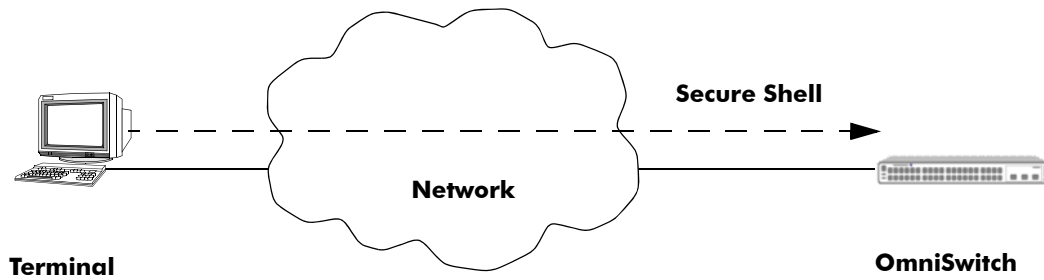
Secure Shell FTP is the standard file transfer protocol used with Secure Shell. Secure Shell FTP is an interactive file transfer program (similar to the industry standard FTP) which performs all file transfer operations over a Secure Shell connection.

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 1-15](#) for detailed information.

Secure Shell Application Overview

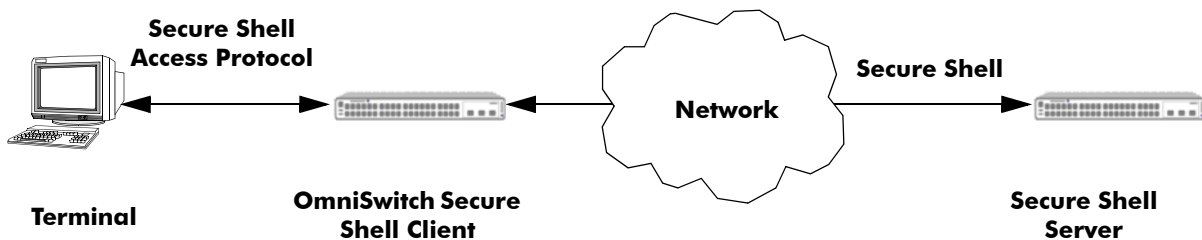
Secure Shell is an access protocol used to establish secured access to your OmniSwitch. The Secure Shell protocol can be used to manage an OmniSwitch directly or it can provide a secure mechanism for managing network servers through the OmniSwitch.

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+2ZwEipJI5HY11qbYKTA0em0kwK
HNa+naIkWsTSwNj81HaAkaL21LMhcHnRytBfTeyySLgNHxy6VFX1ipMN3pdtQbJn0cfRI
evyxroMs7S+nMvht1lhrRzNaC3iW90IskS9zNjKUD2Becj5+Bt1JHmlqu3Is9H67kySd
HeF1XTMVWHD030n9msA1vB7Bqolw26qzV3S97vbhrApQtYJAn0bIilVIAEasIYIbqrkTQ
/kmD04uMpCDgZKta7bP+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 6860 8.1.1.R01, April 26, 2014.  
Copyright (c) 1994-2014 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 6860
Software Version 8.1.1.R01, April 05, 2014.

Copyright(c), 1994-2014 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 6860
Software Version 8.1.1.R01, April 05, 2014.

Copyright(c), 1994-2014 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 IPv4 domain name servers and 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.

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 AOS Release 8 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
```

Enabling the FIPS mode

Federal Information Processing Standards (FIPS) is a mode of operation that satisfies security requirements for cryptographic modules. It is a requirement as per the National Institute of Standards and Technology (NIST), FIPS 140-2 standard that strong cryptographic algorithms has to be supported to achieve FIPS compliance. When FIPS mode is enabled on OmniSwitch, FIPS 140-2 compliant encryption is used by the OmniSwitch devices in the various management interfaces such as SFTP, HTTP, SSh and SSL.

These strong cryptographic algorithms ensure secure communication with the device to provide interoperability, high quality, cryptographically-based security for IP networks through the use of appropriate security protocols, cryptographic algorithms, and keys and prevent any form of hijacking/hacking or attack on the device through the secure mode of communication.

Note The FIPS mode is configurable through all the three User Interfaces: WebView, SNMP and CLI.

FIPS mode functionalities:

- FIPS operates in OpenSSL mode allowing only highly secure and strong cryptographic algorithms.
- OpenSSH and Web Server which use the OpenSSL as the underlying layer for secure communications also works in the FIPS mode.
- SNMPv3 supports secure SHA+AES. MD5 or DES are not allowed.

- The FIPS mode is enabled/disabled only with a reboot of the switch.

The SNMPv3 module as well as all switch management protocols such as SFTP, HTTP, SSH, and SSL use the FIPS 140-2 compliant encryption algorithms.

FIPS Specifications

| | |
|---------------------|--|
| Client | To access an OmniSwitch in FIPS mode, a FIPS supported client is required. For Example, Absolute Telnet. |
| Platforms Supported | OmniSwitch 6860, 6860E |
| Access types | SSH, SFTP, HTTP, SNMPV3 |

Quick Steps for Configuring FIPS mode

Prior to enabling the FIPS mode of communication, complete the following pre-requisites.

- The SSH/SFTP/SSL/SNMPv3 clients should support the secure FIPS standard cryptographic algorithms to communicate with an OmniSwitch device on FIPS mode.
- SNMPv3 communications in the FIPS mode supports SHA+AES. Session establishment with MD5 or DES should be rejected.
- User-specific certificates/ keys have to be generated using FIPS compliant cryptographic algorithms. There are no checks in the OpenSSL module to verify the FIPS compliance of the certificate/keys in the flash.
- When takeover happens, management sessions with the old Primary will be disconnected. User will have to reconnect to the new Primary.

The following procedure is used to configure the FIPS mode on the switch:

- 1 Enable the FIPS mode on an OmniSwitch using the following command.

```
-> system fips admin-state enable
```

```
WARNING: FIPS Admin State only becomes Operational after write memory and reload
```

- 2 Reboot the system, an reconfirmation message is displayed. Type “Y” to confirm reload.

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

```
-> Confirm Activate (Y/N) : y
```

- 3 Use the **show system fips** to view the configured and running status of the FIPS mode on the Switch.

```
-> show system fips
```

```
Admin State: Enabled
```

```
Oper State: Enabled
```

- 4 Disable insecure management interfaces such as Telnet/ FTP manually after FIPS mode is enabled to achieve a complete secure device.

- 5 Configure a user-id and password.

```
-> user snmpadmin password trustsha+aes sha+aes
```

This user-id and password can be used to access an OmniSwitch in secure mode when FIPS is enabled on the switch.

6 Access the OmniSwitch from the SSH/SFTP/SSL/SNMPv3 clients with encryption AES using the user credentials defined.

NoteA FIPS supported client such as Absolute Telnet can be used to access the OmniSwitch.

7 Use the **show user** command to view the SNMP level configured for the user.

```
-> show user = snmpadmin
```

```
User name = snmpadmin,  
Password expiration      = 12/22/2014 11:01 (30 days from now),  
Password allow to be modified date    = 03/25/2014 10:59 (3 days from now),  
Account lockout         = Yes (Automatically unlocked after 19 minute(s)from now),  
Password bad attempts   = 3,  
Read Only for domains   = None,  
Read/Write for domains  = Admin System Physical Layer2 Services policy Security ,  
Read/Write for families = ip rip ospf bgp vrrp ip-routing ipx ipmr ipms ,  
Snmp allowed            = YES,  
Snmp authentication     = SHA,  
Snmp encryption        = AES  
Console-Only           = Disabled
```

A secure session of the user “snmpadmin” is established between the client and the OmniSwitch in FIPS enabled mode.

8 FIPS mode can be disabled using the **system fips admin-state disable** command. When the FIPS mode is disabled, all other existing cryptographic algorithms will be supported.

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. |
| system fips admin-state | Enable or disable the FIPS mode on the switch. |
| show system fips | Show the Configured and Running status of the FIPS mode on the Switch. |
| user | Configures or modifies user entries in the local user database. Use the no form of the command to remove the user from the local database. |
| show user | Displays information about all users or a particular user configured in the local user database on the switch. |

For more information about these commands, refer to the *OmniSwitch AOS Release 8 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 AOS Release 8 CLI Reference Guide*.

In This Chapter

Configuration procedures described in this chapter include:

- [“Switch Administration Overview” on page 2-3](#)
- [“Loading Software onto the Switch” on page 2-13](#)
- [“Installing Software Licenses” on page 2-16](#)

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 6860, 6860E |
| 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 editor. |

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.

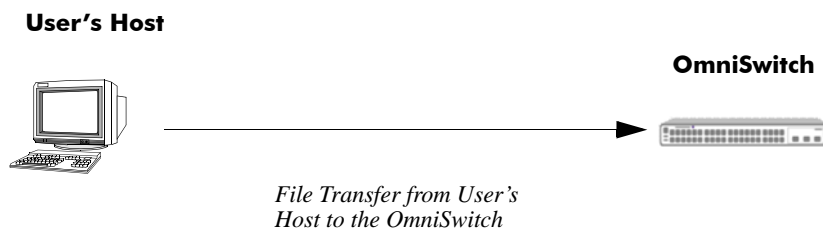
The OmniSwitch Series uses flash memory store files, including executable files (used to operate switch features and applications), configuration files, and log files.

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.

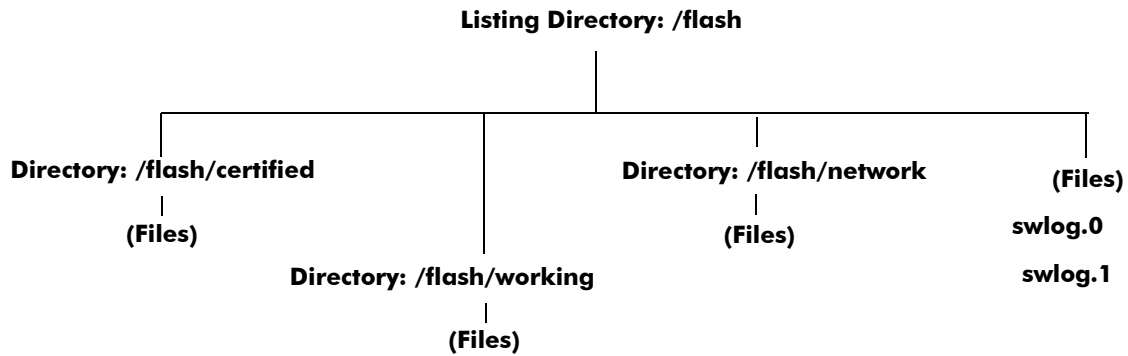


File Transfer to OmniSwitch

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

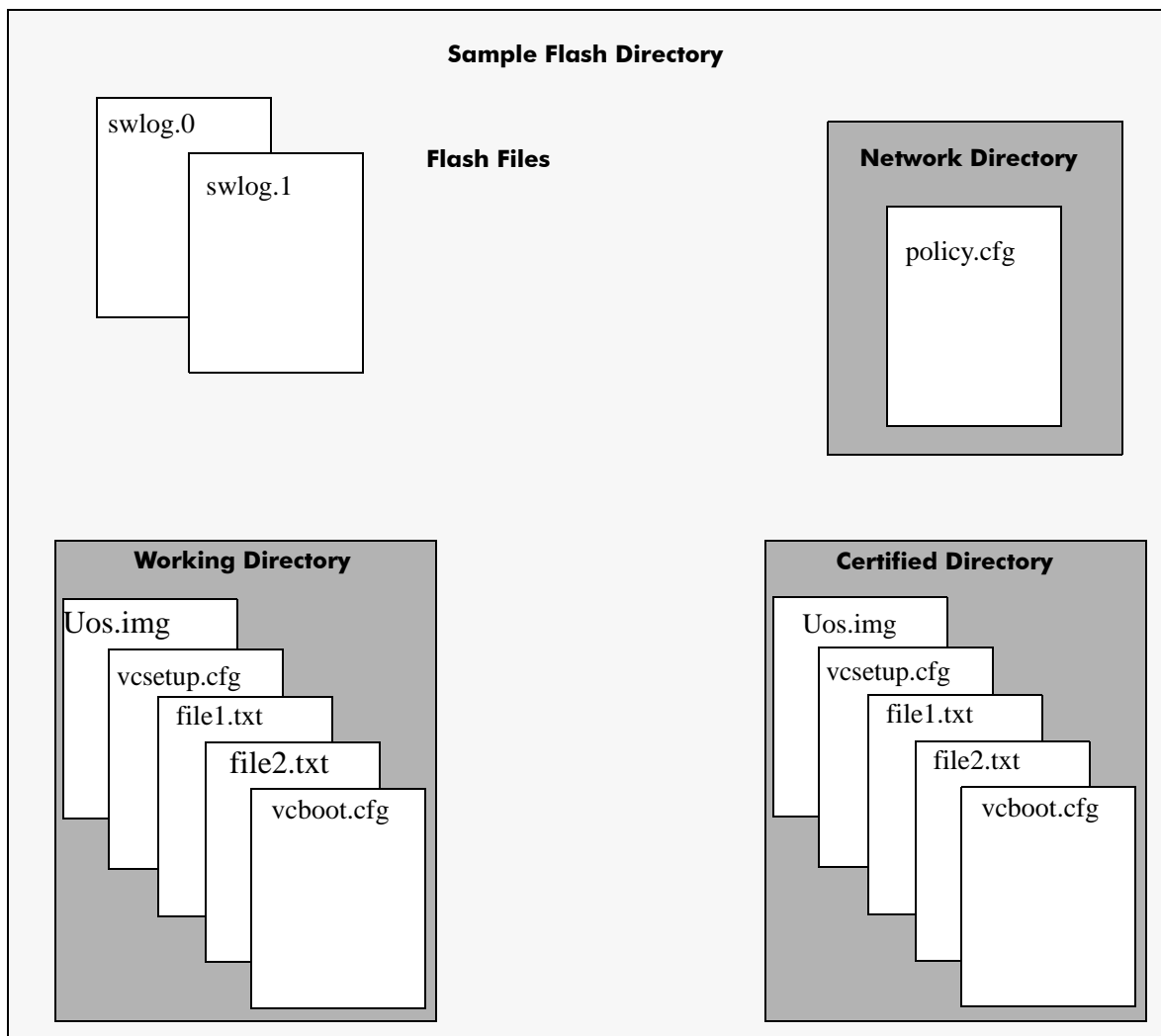
File and Directory Management

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.



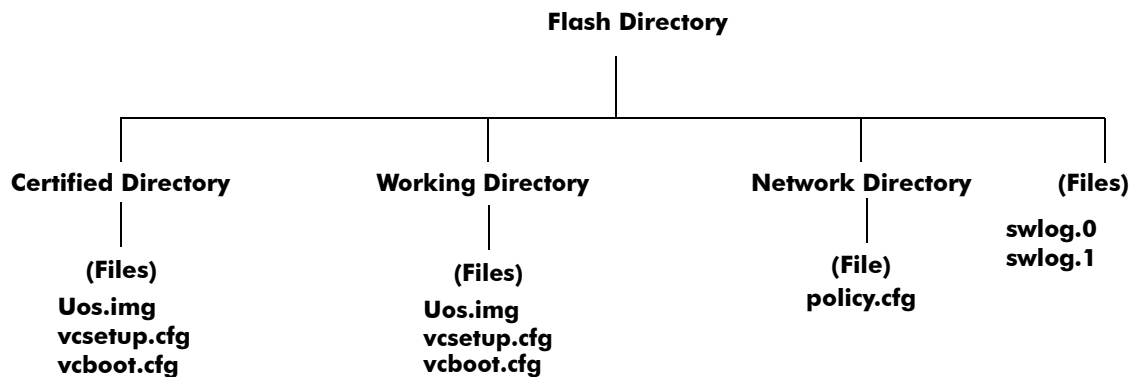
To list all the files and directories in your current directory, use the **ls** command. Here is a sample display of the flash directory.

```
-> 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
```

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.

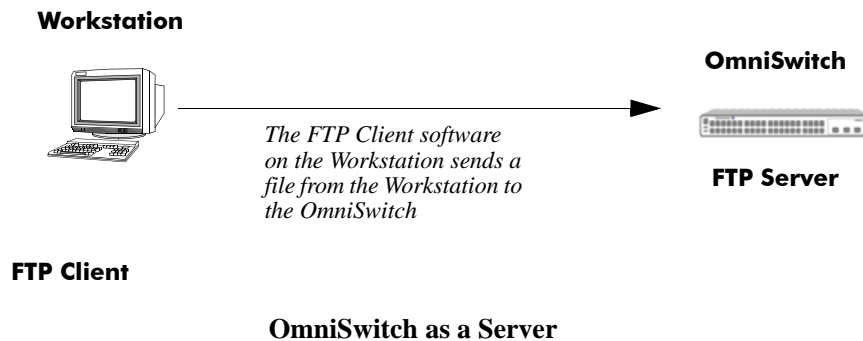
Loading Software onto the Switch

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.



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 192.168.1.1
Connecting to [192.168.1.1]...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 198.168.1.100.

```
-> sftp 192.168.1.100
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 192.168.1.100
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 192.168.1.100 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 192.168.1.100
```

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

- Establishes a TFTP session with the TFTP server 192.168.1.100.
- 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 6860/6860E | | | |
|-----------------------|-------------|----------------------|-------|
| Advanced License | | | |
| OSPF v2/v3 | VRF | Policy-Based Routing | RIPng |
| BGP | PIM-SM/DM | DVMRP | IPSec |
| MP-BGP | IS-IS v4/v6 | 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 2-18](#) 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 2-18](#) 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. 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 AOS Release 8 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” on page 3-8](#).
- Managing, upgrading and restoring files using a USB flash drive described in [“Using the USB Flash Drive” on page 3-14](#).

CMM Specifications

| | |
|-----------------------------------|------------------------|
| Platforms Supported | OmniSwitch 6860, 6860E |
| Size of Flash Memory | 2 GB |
| Maximum Length of File Names | 255 Characters |
| Maximum Length of Directory Names | 255 Characters |
| Maximum Length of System Name | 32 Characters |
| Default Boot Directory | Certified |

USB Flash Drive Specifications

| | |
|----------------------------|--|
| Platforms Supported | OmniSwitch 6860, 6860E |
| USB Flash Drive Support | Alcatel-Lucent Certified USB Flash Drive |
| Automatic Software Upgrade | Supported |
| Disaster Recovery | Urescue.img file required |

Note. The format of the Alcatel-Lucent certified USB Flash Drive must be FAT. 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 **vcboot.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 **vcboot.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 6860/6860E

| File Name | Base or Optional Software | Description |
|-------------|---------------------------|--|
| Uos.img | Base Software | CMM Operating System |
| Urescue.img | Optional Software | Disaster Recovery file (used on USB flash drive) |
| vcboot.cfg | | Switch Configuration File |
| vcsetup.cfg | | Virtual Chassis Setup File |
| boot.md5 | | An automatically created checksum file used to ensure the image files in the running directory cannot be certified until the switch has booted from the running directory with those image files. The boot.md5 file is created in the current running directory from the image files immediately after the switch boots. If new image files are copied into that directory and the “copy running certified” command is attempted, the command will be rejected since the switch has not yet booted with the new image files. This prevents possibly corrupted image files from becoming certified. |

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 reboot from the CERTIFIED 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 3-13.

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 be placed in the *working* 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.

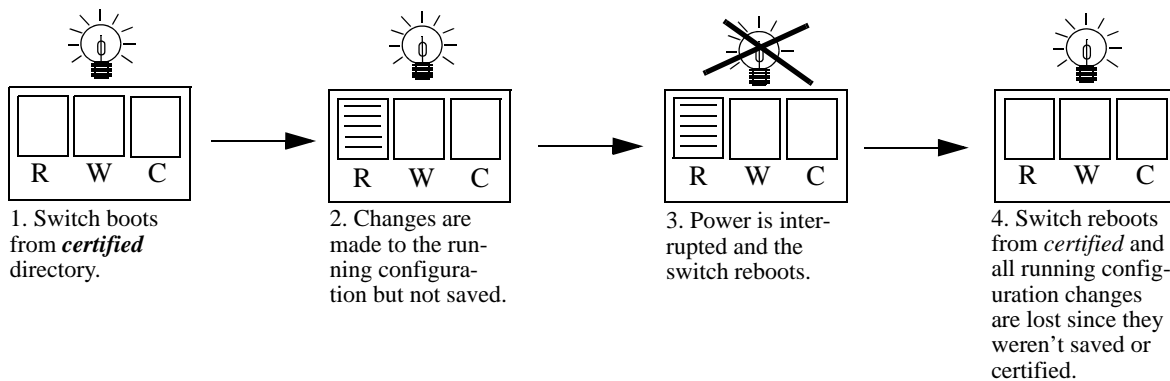
In the examples below, **R** represents the RUNNING CONFIGURATION, **W** represents the *working* directory, and **C** represents the *certified* directory.

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:



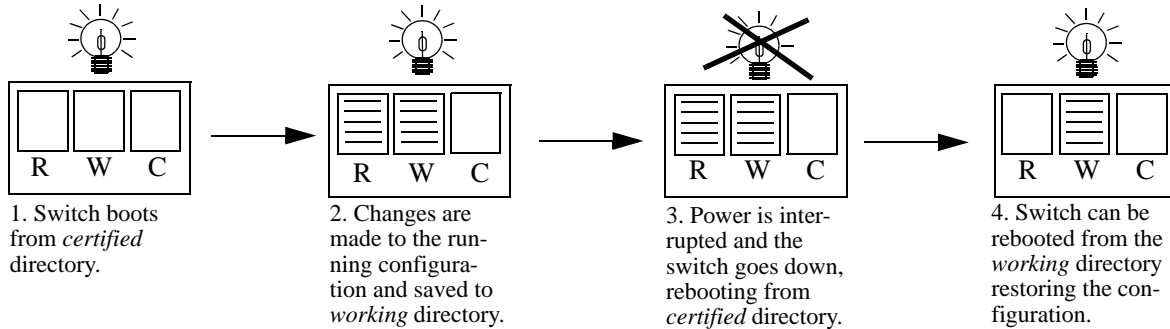
Running Configuration is Overwritten by the Certified Directory on Reboot

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.

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

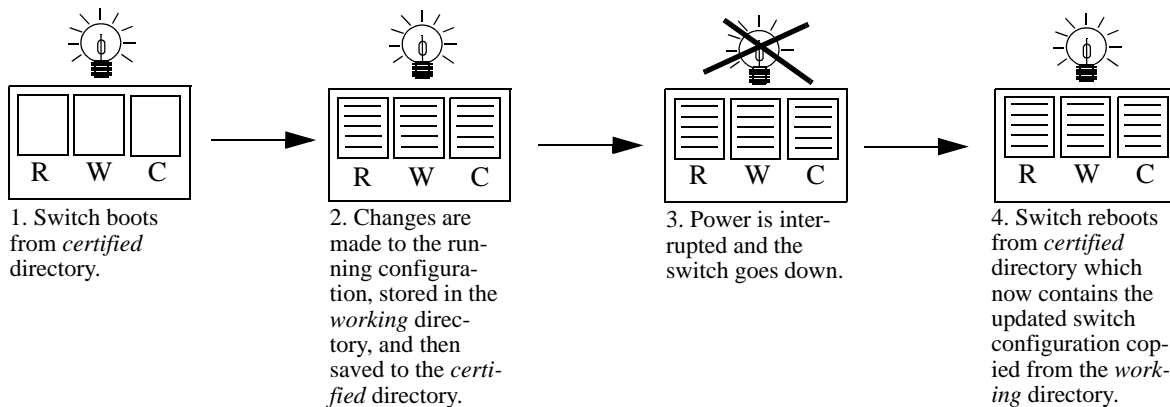
This is illustrated in the diagram below:



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.



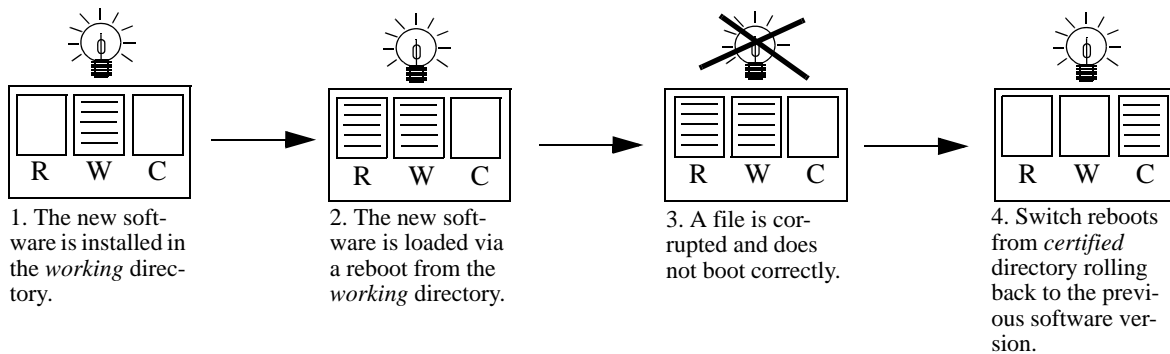
Running Configuration is Saved to Working Directory, then to the Certified Directory

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:



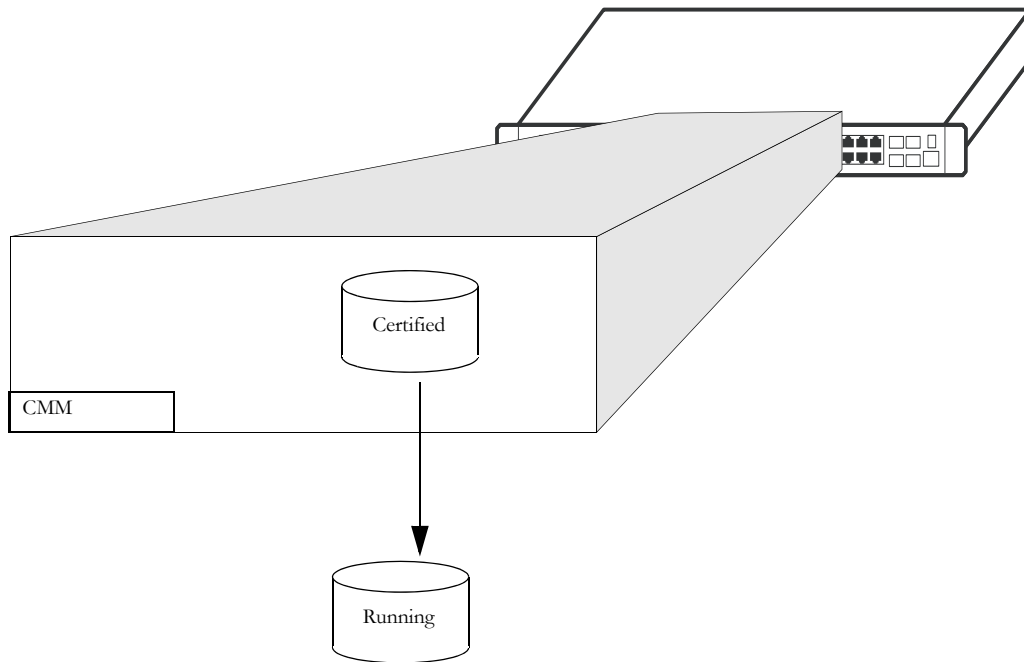
Switch Rolls Back to Previous Software Version

Managing Switch Configurations

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

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 *vbboot.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 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
```

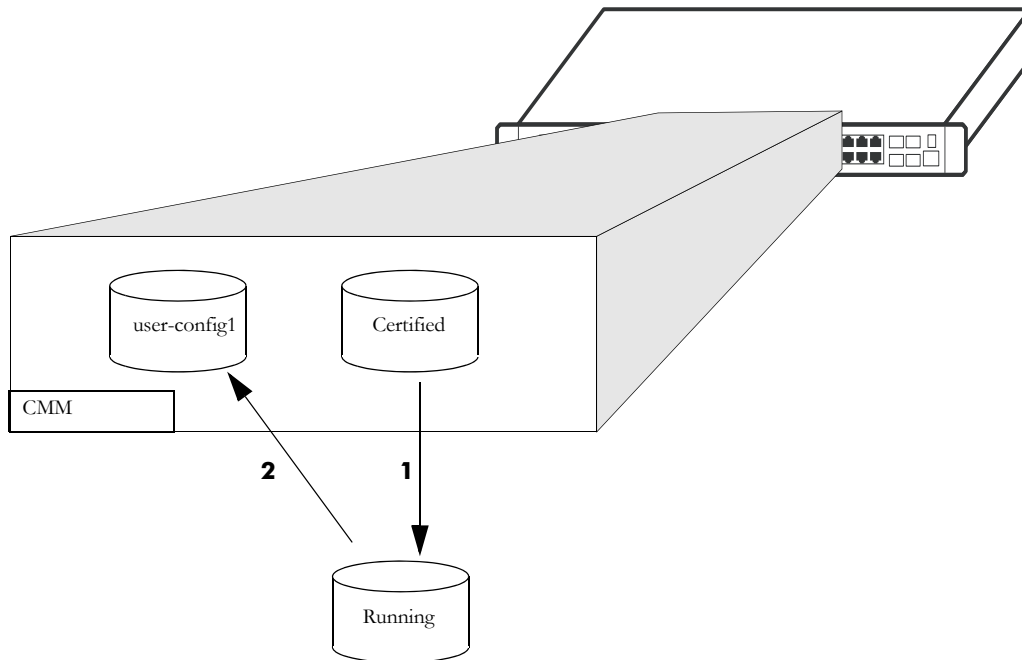
Checking the Status of a Scheduled Reboot

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
```

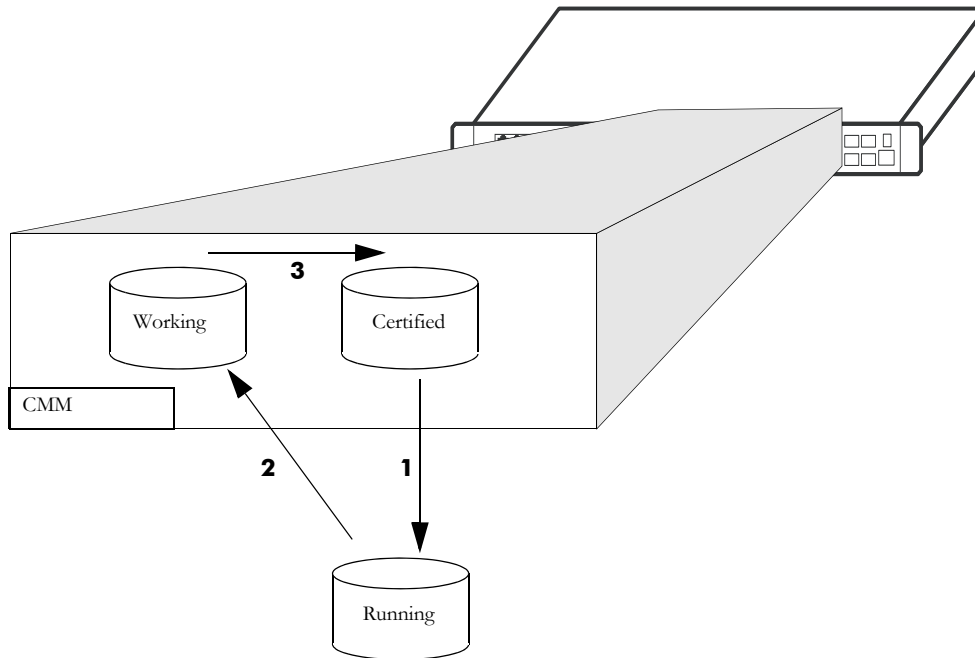
Cancelling a Rollback Timeout

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 *vcboot.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.
-> modify running-directory working
- 2** The RUNNING DIRECTORY is changed from *certified* to a different directory such as *working*.
-> write memory
- 3** The changes are saved to the *working* directory in the *vcboot.cfg* file.
-> copy running certified

Show Currently Used Configuration

Depending on how 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           : MASTER-PRIMARY,
  CMM Mode              : VIRTUAL-CHASSIS MONO CMM,
  Current CMM Slot      : CHASSIS-1 A,
  Running configuration : WORKING,
  Certify/Restore Status : CERTIFY NEEDED
SYNCHRONIZATION STATUS
  Running Configuration : NOT 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
-----+-----+-----+-----
Uos.img      8.1.1.99.R01  2486643 Alcatel-Lucent OS
```

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

Using the USB Flash Drive

An Alcatel-Lucent certified USB flash drive can be connected to the switch 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 switch. For the automatic upgrades and disaster recovery the USB flash drive must be configured with the proper directory structure 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 6860 | 6860/working | 6860/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/vcboot.cfg /uflash/vcboot.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 root directory 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 *6860/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 *vcboot.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 switch. This can be used if the image files on the switch become corrupted, deleted, or the switch is unable to boot 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.

- 1 Create the directory structure *6860/certified* on the USB flash drive with the backup system and configuration files.
- 2 Copy the **Urescue.img** file to the root directory on the USB flash drive.
- 3 Connect the USB flash drive to the switch 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 switch will reboot and be operational again.

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 AOS Release 8 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 AOS Release 8 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 Overview” on page 4-2](#)
- [“Command Entry Rules and Syntax” on page 4-3](#)
- [“Recalling the Previous Command Line” on page 4-5](#)
- [“Logging CLI Commands and Entry Results” on page 4-7](#)

CLI Specifications

The following table lists specifications for the Command Line Interface.

| | |
|-------------------------|---|
| Platforms Supported | OmniSwitch 6860, 6860E |
| Configuration Methods | <ul style="list-style-type: none"> • Online configuration via real-time sessions using CLI commands. • Offline configuration using text file containing CLI commands. |
| Command Capture Feature | Snapshot feature captures switch configurations in a text file. |
| User Service Features | <ul style="list-style-type: none"> • Command Line Editing • Command Prefix Recognition • CLI Prompt Option • Command Help • Keyword Completion • Command Abbreviation • 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 AOS Release 8 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.
- The CLI is bash-based and non-alphanumeric characters have special meaning. Bash rules should be followed for user-defined text strings. To use spaces or non-alphanumeric characters within a user-defined text string, typically you must enclose the entry in single 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 AOS Release 8 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 or non-alphanumeric characters. 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 AOS Release 8 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.

Partial Keyword Abbreviation

The CLI has a partial keyword abbreviation 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*. For example, ‘**show vlan**’ can be abbreviated to:


```
-> sh vl
```

If the keyword cannot be uniquely identified an error will be displayed. For example:

```
-> sh v
ERROR: Invalid entry" "v"
```

The letter 'v' does not uniquely identify a keyword and could stand for multiple keywords such as 'vlan', 'violation' or 'verbose'. The '?' can be used to list the possible keywords.

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 FCK 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 AOS Release 8 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 configurations are dependent up the terminal emulator mode.

Changing the Screen Size

You may specify the size of the display shown on your terminal screen by using the **tty** or **stty** commands. These commands are useful when you have a small display screen or you want to limit the number of lines scrolled to the screen at one time or improve mis-aligned output. For example:

```
-> tty 10 150  
  
or  
  
-> stty cols 255
```

The first number entered after **tty** defines the number of lines on the screen and the second number defines the number of columns on the screen. The **stty cols** command changes the number of columns. You may view the current setting for your screen by using the **show 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 AOS Release 8 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 the switch usability.

In This Chapter

Configuration procedures described in this chapter include:

- [“Tutorial for Creating a Configuration File” on page 5-2](#)
- [“Applying Configuration Files to the Switch” on page 5-6](#)
- [“Configuration File Error Reporting” on page 5-7](#)
- [“Text Editing on the Switch” on page 5-8](#)
- [“Creating Snapshot Configuration Files” on page 5-9](#)

Configuration File Specifications

The following table lists specifications applicable to Configuration Files.

| | |
|--|--|
| Platforms Supported | OmniSwitch 6860, 6860E |
| Creation Methods for Configuration Files | <ul style="list-style-type: none"> • 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 AOS Release 8 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.

```
-> show configuration status  
  
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
```

- 3 (Optional). When the keyword **verbose** is specified in the command line, all syntax contained in the configuration file is printed to the console.

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

- 4 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
```

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
```

- 3 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
```

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
```

- 3 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
```

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 AOS Release 8 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, user the CLI help to get a list of all available features:

```
-> configuration snapshot ?
```

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 OS6860  
! 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 lxl
! 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 <i>identical</i> or <i>different</i> . 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 AOS Release 8 CLI Reference Guide*.

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

- [“Creating a User” on page 6-8.](#)
- [“Configuring Password Policy Settings” on page 6-10.](#)
- [“Configuring Privileges for a User” on page 6-15.](#)
- [“Setting Up SNMP Access for a User Account” on page 6-16.](#)
- [“Multiple User Sessions” on page 6-18](#)

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 8 Network Configuration Guide*.

User Database Specifications

| | |
|--|------------------------|
| Platforms Supported | OmniSwitch 6860, 6860E |
| Maximum number of alphanumeric characters in a username | 63 |
| Maximum number of alphanumeric characters in a user password | 30 |

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-contains-username | disabled |
| Minimum number of uppercase characters allowed in a password. | user password-policy min-upper-case | 0 (disabled) |
| Minimum number of lowercase characters allowed in a password. | user password-policy min-lower-case | 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-non-alpha | 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 8 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 6-11](#).

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 6-15 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 6-16 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 6-8](#).

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 6-15](#).

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 AOS Release 8 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 6-10](#) and [“Configuring Global User Lockout Settings” on page 6-13](#) 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 **vcboot.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 lowercase 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.
- Passwords with non-alphanumeric characters must be enclosed in single quotes.
- 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 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 02/19/2014 13:30
```

This command sets the password expiration to February 19, 2014, 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 6-13) 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 6-14).

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 6-13). 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 6-14](#)) 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 6-13](#)).

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 6-14](#)) 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 AOS Release 8 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 AOS Release 8 CLI Reference Guide*.

An overview of switch security is given in this chapter. In addition, configuration procedures described in this chapter include:

- [“Configuring Authenticated Switch Access” on page 7-6](#)
- [“Setting Up Management Interfaces for ASA” on page 7-9](#)
- [“Configuring Accounting for ASA” on page 7-11](#)

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 *OmniSwitch AOS Release 8 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

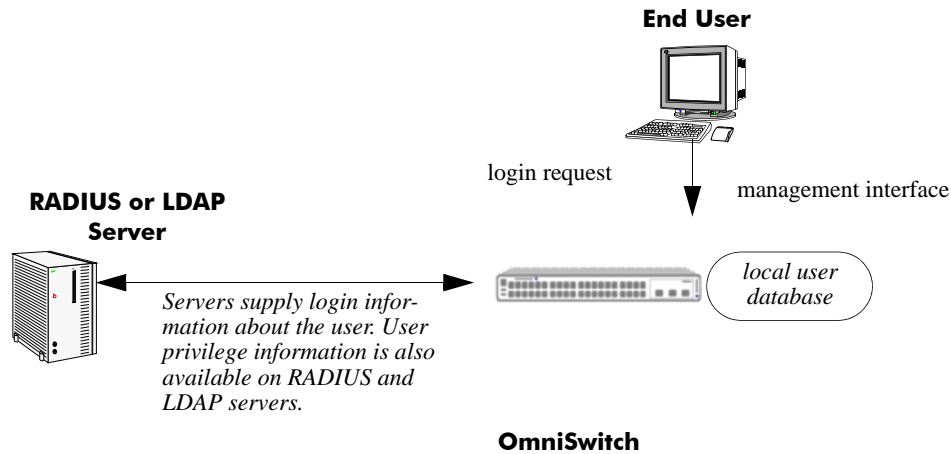
Access to managing the switch is always available for the **admin** user through the console port, even if management access to the console port is disabled for other users.

| 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 *OmniSwitch AOS Release 8 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 *OmniSwitch AOS Release 8 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.

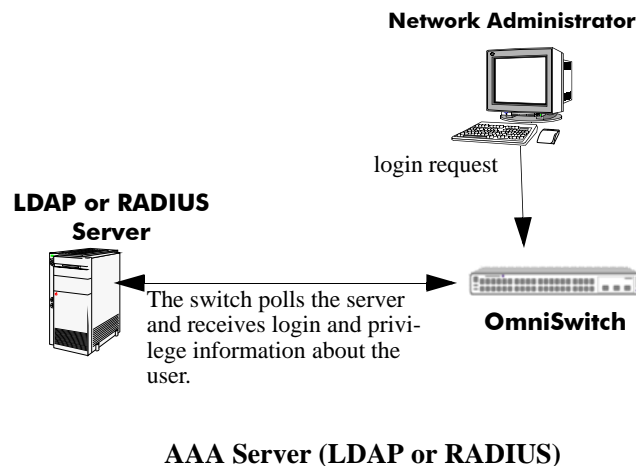
This section describes how to configure management interfaces for authenticated access as well as how to specify external servers that the switch can poll for login information. The type of server may be an authentication-only mechanism or an authentication, authorization, and accounting (AAA) mechanism.

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:



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 *OmniSwitch AOS Release 8 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 *OmniSwitch AOS Release 8 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 ldap-server aaa tacacs+-server | Setting up the switch to communicate with external RADIUS, LDAP or TACACS 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 | <i>Optional.</i> 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 *OmniSwitch AOS Release 8 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 *OmniSwitch AOS Release 8 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 AOS Release 8 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 *OmniSwitch AOS Release 8 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 *OmniSwitch AOS Release 8 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 *OmniSwitch AOS Release 8 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 AOS Release 8 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 Specifications

| | |
|---------------------|--|
| Platforms Supported | OmniSwitch 6860, 6860E |
| Supported Browsers | Internet Explorer for Windows Firefox for Windows, Linux, and Solaris SunOS |

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.

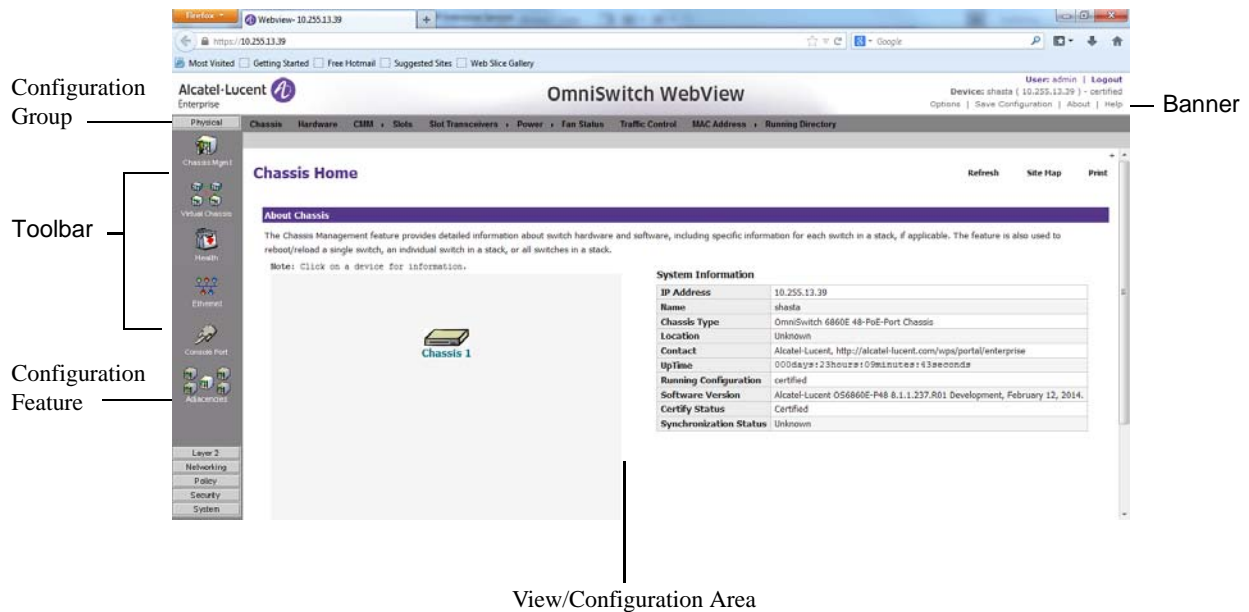
WebView Overview

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 Chassis 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 AOS Release 8 CLI Reference Guide*.

Configuration procedures described in this chapter include:

- [“Setting Up An SNMP Management Station”](#) on page 9-3
- [“Setting Up Trap Filters”](#) on page 9-4
- [“Using SNMP For Switch Security”](#) on page 9-9
- [“Working with SNMP Traps”](#) on page 9-12

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 6860, 6860E |
| 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 |
| 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, AES |
| 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 “SNMP Trap Information” on page -1 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 |
| 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 AOS Release 8 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 9-9](#). For more information on the CLI commands and the displays in these examples, refer to the *OmniSwitch AOS Release 8 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 9-3](#)

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 9-9](#). 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 9-4](#). For a list of trap names, command families, and their descriptions refer to the [“Using SNMP For Switch Security” on page 9-9](#).

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 9-5](#). For a list of trap names, ID numbers, and their descriptions refer to the table [“Using SNMP For Switch Security” on page 9-9](#).

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 AOS Release 8 CLI Reference Guide*.

10 Web Services, CLI Scripting, and OpenFlow

The Web Services feature provides the ability to customize and extend the management interface on AOS devices. It supports the use of CLI scripting in AOS as well as a REST based 'web' interface that interacts with AOS management variables (MIB) and CLI commands. It provides two methods for configuration through either the direct handling of MIB variables or the use of CLI commands and supports both XML and JSON response formats.

OpenFlow is a communications interface defined between the control and forwarding layers that is used in a Software Defined Network (SDN). OpenFlow separates the control plane and the data plane in the . Traditionally, switches and routers have made decisions on where packets should travel based on rules local to the device.

In This Chapter

This chapter contains the following information:

- [“Web Services Specifications” on page 10-2](#)
- [“Web Services Overview” on page 10-2](#)
- [“Web Services REST Examples” on page 10-5](#)
- [“Using Python” on page 10-15](#)
- [“CLI Scripting” on page 10-20](#)
- [“OpenFlow Specifications” on page 10-25](#)
- [“OpenFlow Agent Overview” on page 10-26](#)
- [“Quick Steps to Configure OpenFlow Agent” on page 10-27](#)

Web Services Specifications

The following table lists specifications for Web Services.

| | |
|---------------------------------------|--|
| Platforms Supported | OmniSwitch 6860, 6860E |
| Configuration Methods | <ul style="list-style-type: none"> • HTTP/HTTPS • Python API |
| Response Formats | <ul style="list-style-type: none"> • Extensible Markup language (XML) • JavaScript Object Notation (JSON) |
| Maximum Web Services Sessions | 4 |
| Alcatel-Lucent Example Python Library | <p>consumer.py (Python version 2.X compatible)</p> <p>Note: This file is available on the Service & Support Website. It is being provided as an example application to help with Web Services familiarization but is not an officially supported part of the Web Services solution.</p> |

Web Services Overview

The Web Services interface provides two levels of granularity, either through direct handling of MIB variables or using the embedded CLI commands to configure the switch. The Web Services feature provides a RESTful interface to OmniSwitch configuration.

Representational State Transfer (REST)

REST is a set of guidelines for software architecture of distributed systems. It is an architectural style with the following characteristics:

REST Characteristics

- **Client-Server architecture:** all interactions are based on a set of Consumers performing pull-based interactions with a set of Producers.
- **Stateless:** each request from Consumers to Producers must be self-sufficient and not presume any pre-agreed upon knowledge on the Producer side. Each request must contain all information necessary for the Producer to understand and reply to it. If a new resource or API is identified, the Producer needs to return a unique URL to the Consumer who will then re-use that URL when communicating with the Producer. This is known as **Interconnected Resource Representation:** this succession of URLs is how a Consumer can move from one state to another without the Producer needing to maintain any state information.
- **Cacheable:** when similar requests are issued repeatedly to a Producer, existing HTTP caching mechanisms must be capable to cache results the way HTTP caches usually do. Caching can be handled using the usual mechanisms: unique URL generation and cache lifecycle headers. This reliance on caches, proxies, etc. follows the natural layer model found in Web models.
- **Names Resources:** all resources are named using a Uniform Resource Identifier (URI). Their location is defined using a complete URL. No URL is to be manually recreated client-side based on previous assumptions. All URLs are assumed to be canonical.

- **Uniform Interface:** all resources can be thought of as nouns: as hinted before, both state representation and functionality are expected to be represented using nouns; and accessed using a minimal set of verbs: GET, POST, PUT, and DELETE.
- **Media Types:** These are to be used to identify the type of resources being dealt with.

REST Verbs

As described earlier, only a small set of verbs are be used. They are:

- **GET:** To retrieve information. It is a rough equivalent to SNMP/MIP GET but also, at a higher level, a SHOW command. This is exclusively for read-only, side-effect free commands.
- **PUT:** To create new information. For instance, a new VLAN. This is a write operation.
- **POST:** The same action used when submitting web forms is used, in a Web Service context, to update existing information.
- **DELETE:** To delete information. This verb is used to delete resources.

Unsupported verbs will cause the Producer to return an error diagnostic such as '405 Method Not Allowed'

Web Service routing

The producer (server-side) is implemented by piggybacking on top of the existing Webview architecture. WebView continues to provide web pages as usual. However, when a certain URL is requested ("Web Service Endpoint"), information is interpreted and delivered using alternative formats such as JSON or simple XML, rather than HTML pages or HTML forms.

Security

Security is maintained through the use of backend sessions and frontend cookies which is the same as current HTTP security for thin clients.

- **Authentication** - Adheres to a web-service model, through its own REST domain and use of the GET verb.
- **Authorization** - Follows the usual authorization mechanism already in use in WebView, where WebView checks with Partition Manager what permission families a user belongs to, thus specifying which MIB tables are accessible to that user.
- **Encryption** - Follows the same model as WebView: if unencrypted access ("HTTP") is allowed, then the Web Service is allowed over the same transport. Similarly, if listening HTTP/HTTPS ports are changed, the Web Service will be available through those ports.

AOS REST Implementation

All requests are performed through a URL being in accordance with the principles of REST. The following elements are used to build the REST URL:

Protocol - The protocol can be 'http' which defaults to port 80, or 'https' which defaults to port 443. HTTPS is encrypted and HTTP is clear-text.

Server address[:port] - Server address: the IP address typically used to access the switch's WebView interface. If the listening port was changed, the port number should be appended after ':' The combi-

nation of Protocol + Server address[:port] constitutes the Web Service's endpoint.

Domain - This is the first element the AOS REST web service will look at. It indicates in what domain the resource being accessed is located as listed below:

- MIB - Used to denote accessing MIB variables.
- CLI - Used to ask the web service to run CLI commands.
- INFO - Used to return information on a MIB variable.

URN - A Unified Resource Name represents the resource to be accessed.. For instance, when reading information from the 'mib' domain, URNs are MIB variables names; in most instances, tables. The URN is accessed using the following verbs: GET, PUT, POST, DELETE.

Variables - A list of variables that are dependant on the domain being accessed. When reading from the 'mib' domain, this is a list of variables to be retrieved from a MIB table.

Output format

The output format can be encoded using either XML or JSON. The Accept request-header can be used to specify a given media type and leveraged to specify what the output type will be:

- application/vnd.alcatellucentaos+json
- application/vnd.alcatellucentaos+xml

Caching

Due to the volatile nature of the content being returned, the producer will instruct any system sitting between the producer and the consumer (included) not to cache its output. The following headers are sent by the producer:

- Cache-Control: no-cache, no-store
- Pragma: no-cache
- Vary: Content-Type

The first two headers indicate that caching should not take place. The last header is intended for proxy servers, informing them that the Content-Type header is a variable not to cache. Should a proxy server decide not to respect the latter header it's possible to have unexpected behaviors such as retrieving JSON-encoded data after specifically requesting XML-encoded data.

Web Services REST Examples

All requests are performed through a URL being in accordance with the principles of REST. The following elements are used to build the REST URL

Query Structure

- Endpoint: <http|https>://<device ip>
- Unified Syntax: <endpoint>/<domain>/<URN>
<var 1> .. <var n>

JSON or XML

The response format can be returned in either JSON or XML.

```
GET https://192.168.1.1/auth/?username=admin&password=switch
Accept: application/vnd.alcatellucentaos+json
```

Response Elements

| | |
|---------------|---|
| domain | Shows how the Producer interpreted the domain parameter; in most instances, it will be the same domain passed by the Consumer plus some internal information |
| diag | This integer will be a HTTP standard diagnostic code: <ul style="list-style-type: none"> • A 2xx value if the command was successful; in most cases '200' will be used. • A 3xx value if a resources was moved (not implemented). • A 4xx value if the request contained an error; e.g. '400' in case of failed authentication. • A 5xx value if the server encountered an internal error such as a resource error. |
| error | May be a string, containing a clear text error message. It may also be an array of such strings in case the Producer found multiple problems with a request. |
| output | In some instances, the subsystem being queried may wish to return a "blob of text" and this variable will contain it. |
| data | If a GET request is issued this variable should contain the values being queried in a structured form. |

Login Example

This REST example logs a user into the switch.

| | |
|------------------|---|
| Domain | auth |
| URN | - |
| Verb | GET |
| Variables | username, password |
| REST URL | GET https://192.168.1.1/auth/?&username=admin&password=switch |

Example Success Response

| JSON | XML |
|---|---|
| <pre>{ "result":{ "domain":"auth (login)", "diag":200, "error":""," "output":""," "data":[] }</pre> | <pre><?xml version="1.0" encoding="UTF-8" ?> <nodes> <result> <domain>auth (login)</domain> <diag>200</diag> <error></error> <output></output> <data></data> </result> </nodes></pre> |

Example Error Response

| JSON | XML |
|---|--|
| <pre>{ "result":{ "domain":"auth (login)", "diag":400, "error":"Authentication failure : Invalid login name or password", "output":""," "data":[] }</pre> | <pre><?xml version="1.0" encoding="UTF-8" ?> <nodes> <result> <domain>auth (login)</domain> <diag>400</diag> <error>Authentication failure : Invalid login name or password</ error> <output></output></pre> |

Logout Example

This REST example logs a user out of the switch.

| | |
|------------------|--------------------------------|
| Domain | auth |
| URN | - |
| Verb | GET |
| Variables | - |
| REST URL | GET https://192.168.1.1/auth/? |

Example Success Response

| JSON | XML |
|--|--|
| <pre>{ "result": { "domain": "auth (logout)", "diag": 200, "error": "", "output": "", "data": [] } }</pre> | <pre><?xml version="1.0" encoding="UTF-8" ?> <nodes> <result> <domain>auth (logout)</domain> <diag>200</diag> <error></error> <output></output> <data></data> </result> </nodes></pre> |

Create Table Entry Example - VLAN

The following REST example creates a new VLAN using MIB objects.

| | |
|-----------------|---|
| Domain | mib |
| URN | vlanTable |
| Verb | PUT |
| REST URL | PUT https://192.168.1.1/mib/vlanTable? mibObject0=vlanNumber:2&mibObject1=vlanDescription:VLAN-2 |

Example Success Response

| JSON | XML |
|---|--|
| <pre>{ "result": { "domain": "mib:vlanTable", "diag": 200, "output": "", "error": ["Set operation finished successfully!"], "data": [] } }</pre> | <pre><?xml version="1.0" encoding="UTF-8" ?> <nodes> <result> <domain>mib:vlanTable</domain> <diag>200</diag> <output></output> <error> <node name="0">Set operation finished successfully!</node> </error> <data></data> </result> </nodes></pre> |

Example Error Response

| JSON | XML |
|---|--|
| <pre>{ "result": { "domain": "mib:vlanTable", "diag": 400, "output": "", "error": ["Submission failed : VLAN Id should be between 1 and 4096 (inclusive)"], "data": [] } }</pre> | <pre><?xml version="1.0" encoding="UTF-8" ?> <nodes> <result> <domain>mib:vlanTable</domain> <diag>400</diag> <output></output> <error> <node name="0">Submission failed : VLAN Id should be between 1 and 4096 (inclusive)</node> </error> <data></data> </result> </nodes></pre> |

Create Table Entry Example - IP Interface

The following REST example creates an IP interface using MIB objects.

| | |
|-----------------|--|
| Domain | mib |
| URN | alaIpItfConfigTable and alaIpInterface |
| Verb | PUT |
| REST URL | PUT https://192.168.1.1/mib/alaIpItfConfigTable? mibObject1=alaIpItfConfigName:my_new_interface2&mibObject0=alaIpItfConfig IfIndex:0 POST Request: [https://192.168.1.1/mib/alaIpInterfaceTable?] mibObject1=alaIpInterfaceAddress:2.1.1.1&mibObject0=ifIndex:13600002&mibO bject3=alaIpInterfaceVlanID:1&mibObject2=alaIpInterfaceMask:255.255.255.0 |

Example Success Response

| JSON | XML |
|---|--|
| <pre>{ "result":{ "domain":"mib:vlanTable", "diag":200, "output":"", "error":["Set operation finished successfully!"], "data":[] }</pre> | <pre><?xml version="1.0" encoding="UTF-8" ?> <nodes> <result> <domain>mib:vlanTable</domain> <diag>200</diag> <output></output> <error> <node name="0">Set operation finished successfully!</node> </error> <data></data> </result> </nodes></pre> |

Example Error Response

| JSON | XML |
|--|--|
| <pre>{ "result":{ "domain":"mib:vlanTable", "diag":400, "output":"", "error":["Submission failed : VLAN Id should be between 1 and 4096 (inclusive)"], "data":[] }</pre> | <pre><?xml version="1.0" encoding="UTF-8" ?> <nodes> <result> <domain>mib:vlanTable</domain> <diag>400</diag> <output></output> <error> <node name="0">Submission failed : VLAN Id should be between 1 and 4096 (inclusive)</node> </error> <data></data> </result> </nodes></pre> |

Modify Table Entry Example - VLAN

The following REST example modifies the VLAN description for an existing VLAN using MIB objects.

| | |
|------------------|--|
| Domain | mib |
| URN | vlanTable |
| Verb | POST |
| Variables | mibObject0, mibObject1 |
| REST URL | POST https://192.168.1.1/mib/vlanTable? mibObject1=vlanNumber:2&mibObject0=vlanDescription:vlan-Two |

Example Success Response

| JSON | XML |
|--|--|
| <pre>{ "result":{ "domain":"mib:vlanTable", "diag":200, "output":""," "error":["Set operation finished successfully!"], "data":[]}}</pre> | <pre><?xml version="1.0" encoding="UTF-8" ?> <nodes> <result> <domain>mib:vlanTable</domain> <diag>200</diag> <output></output> <error> <node name="0">Set operation finished successfully!</node> </error> <data></data> </result> </nodes></pre> |

Modify Table Entry Example - Interface Speed

The following REST example modifies the interface speed for a port using MIB objects.

| | |
|------------------|---|
| Domain | mib |
| URN | esmConfigTable |
| Verb | POST |
| Variables | mibObject0, mibObject1 |
| REST URL | POST Request: https://192.168.1.1/mib/esmConfTable?mibObject0=esmPortCfgSpeed:1000&mibObject1=ifIndex:1001&mibObject2=esmPortCfgDuplexMode1 |

Example Success Response

| JSON | XML |
|---|---|
| <pre>{ "result": { "domain": "mib:esmConfTable", "diag": 200, "output": "", "error": ["Set operation finished successfully!"], "data": [] } }</pre> | <pre><?xml version="1.0" encoding="UTF-8"?> <nodes> <result><domain>mib:esmConfTable</domain> <diag>200</diag> <output></output> <error><node name="0">Set operation finished successfully!</node> </error>\n<data></data> </result> </nodes></pre> |

Delete Table Entry Example

The following REST example deletes an existing VLAN using MIB objects.

| | |
|-----------------|--|
| Domain | mib |
| URN | vlanTable |
| Verb | DELETE |
| REST URL | DELETE https://192.168.1.1/mib/vlanTable? mibObject1=vlanNumber:2 |

Example Success Response

| JSON | XML |
|---|--|
| <pre>{ "result": { "domain": "mib:vlanTable", "diag": 200, "output": "", "error": ["Set operation finished successfully!"], "data": [] } }</pre> | <pre><?xml version="1.0" encoding="UTF-8" ?> <nodes> <result> <domain>mib:vlanTable</domain> <diag>200</diag> <output></output> <error> <node name="0">Set operation finished successfully!</node> </error> <data></data> </result> </nodes></pre> |

Example Error Response

| JSON | XML |
|---|--|
| <pre>{ "result": { "domain": "mib:vlanTable", "diag": 400, "output": "", "error": ["Submission failed : VLAN 5 does not exist"], "data": [] } }</pre> | <pre><?xml version="1.0" encoding="UTF-8" ?> <nodes> <result><domain>mib:vlanTable</domain> <diag>400</diag> <output></output> <error><node name="0">Submission failed : VLAN 5 does not exist</ node> </error> <data></data> </result> </nodes></pre> |

Query Table Info Example

The following REST example queries the VLAN table for an existing VLAN using MIB objects.

| | |
|-----------------|---|
| Domain | info |
| URN | vlanTable |
| Verb | GET |
| REST URL | GET https://192.168.1.1/info/vlanTable? |

Example Success Response

| JSON | XML |
|--|--|
| <pre>{ "result":{ "domain":"info", "diag":200, "output":""," "error":""," "data":{ "table":"vlanTable", "type":"Table", "rowstatus":"vlanStatus", "firstobject":"vlanStatus" } } }</pre> | <pre><?xml version="1.0" encoding="UTF-8" ?> <nodes> <result> <domain>info</domain> <diag>200</diag> <output></output> <error></error> <data> <table>vlanTable</table> <type>Table</type> <rowstatus>vlanStatus</rowstatus> <firstobject>vlanStatus</firstobject> </data> </result> </nodes></pre> |

CLI Example

The following REST example return the output of the 'show vlan' command using the CLI.

| | |
|-----------------|--|
| Domain | cli |
| URN | aos |
| Verb | GET |
| REST URL | GET https://192.168.1.1/cli/aos?&cmd=show+vlan+5 |

Example Success Response

| JSON | XML |
|--|--|
| <pre>{ "result": { "domain": "cli", "cmd": "show vlan 1", "diag": 200, "output": "Name VLAN-1, \nType Static Vlan, \nAdministrative State : enabled, \nOperational State : enabled, \nIP Router Port : enabled, \nIPMTU : 1500\n", "error": "", "data": [] } }</pre> | <pre><?xml version="1.0" encoding="UTF-8" ?> <nodes> <result><domain>cli</domain> <cmd>show vlan 5</cmd> <diag>200</diag> <output>Name : VLAN-5, Type : Static Vlan, Administrative State : enabled, Operational State : enabled, IP Router Port : enabled, IP MTU : 1500 </output> <error></error> <data></data> </result> </nodes></pre> |

Example Error Response

| JSON | XML |
|---|--|
| <pre>{ "result": { "domain": "show vlan 5", "diag": 400, "output": "", "error": "VLAN 342 does not exist\n", "data": [] } }</pre> | <pre><?xml version="1.0" encoding="UTF-8" ?> <nodes> <result><domain>mib:vlanTable</domain> <diag>400</diag> <output></output> <error><node name="0">Submission failed : VLAN 5 does not exist</ node> </error> <data></data> </result> </nodes></pre> |

Using Python

Python is an easy to learn, powerful, general-purpose scripting language. It combines easily readable code with an object-oriented programming approach for fast and easy development on many platforms. Additional information on Python as well as installation instructions can be found from the Python website: <http://www.python.org>.

A Python library has been created which can be used by any Python Consumer communicating with the AOS Web Service Provider. The library is available in source form and provides a tool allowing developers to learn how to write code that communicates with the OmniSwitch Web Service Provider. In addition, this library can also be used as a standalone query tool using the command line.

Library Use

Invoking the library from third-party code is as simple as importing the relevant classes:

```
from consumer import AOSAPI, AOSConnection
```

The library itself relies on the dependency injection pattern, allowing the implementer to replace only bits of the library with their own code, should they need to do so. The two example components imported above allow a connection to be established to an AOS device.

Connection Example

A typical connection to an AOS device should look like this:

```
def do_something():
    try:
        api = AOSAPI(AOSConnection(
            username = 'admin',
            password = 'switch',
            hostaddress = '192.168.1.1',
            secure = False,
            obeyproxy = False,
            prettylinks = True,
            useport = 80,
            aosheaders = None,
            debug = True))
        api.login()

        # Additional code goes here

        api.logout()
    except HTTPError, e:
        api.logout()
        print "Error: " + e.msg
```

Query Example

Augmenting the code above to perform a query is straightforward. Simply call **api.query()** and check its **success()** property as in the example below:

```
results = api.query('mib', 'chasEntTemperatureTable', {
    'mibObject0': 'chasEntTempCurrent',
    'mibObject1': 'chasEntTempThreshold',
```

```
        'mibObject2':'chasEntTempDangerThreshold'})['result']
if api.success():
    return results['data']['rows']
else:
    raise HTTPError("Bad Diag: %d" % api.diag())
```


PYTHON APIs - Quick Reference

AOSAPI (AOSConnection connection)

Connection is an AOSConnection object being injected into AOSAPI. The client implementer can write their own connection class and use it instead.

Methods

| | |
|----------------------------------|--|
| login() | Invoke this method to log in to the Web Service. A cookie will be created. |
| logout() | Invoke this method to log out from the Web Service. If a cookie exists, it be destroyed. |
| query(domain, urn, args) | <p>Invoke this method to perform a "show" query or run a CLI command.</p> <p>domain - the semantic domain being accessed. when accessing mibs, it can be 'mib' if performing a 'show' command; it can be 'info' to retrieve information on a mib table (helpful when developing new queries); when running a CLI command, domain must be 'cli';</p> <p>urn - represents the "address" of the entity being accessed: when accessing mibs, it will typically be a mib table name; when running a CLI command, it will represent the CLI type being used; in version 1, only one type is available: 'AOS'</p> <p>args - is a dictionary of key->value pairs where each respective key's name is 'mibObjectx' and the trailing x is a value ranging from 0 to (max number of arguments - 1): when accessing mibs, the values will typically be the name of the table columns being accessed; when running a CLI command, the dictionary will contain only one element, named 'cmd'; its value will be the command's plain text representation followed by an equal sign (=), followed by the value being used for filtering results.</p> |
| put(domain, urn, args) | <p>Invoke this method to create a new object. It is not a valid command when using the 'cli' domain.</p> <p>domain - is the same as described in the 'query()' section, except 'cli' is not supported.</p> <p>urn - is the same as described in the 'query()' section, except no cli-related value is supported.</p> <p>args - is a dictionary of key->value pairs as described in the 'query()' section, with a major difference: the values will be composed of a column name, followed by a column, followed by the value being set;</p> |
| post(domain, urn, args) | Invoke this method to update an existing object. Arguments are the same as described in the 'put()' section with one semantic difference: values specified for columns that belong to a table index will be used to, first, find the row matching this index, then update the value of the non-index columns specified in this query. |
| delete(domain, urn, args) | Invoke this method to delete an object. Arguments are the same as described in the 'post()' section; however, non-index columns will be ignored. |

| | |
|------------------|---|
| success() | This method will return true if the previous operation succeeded. It is a convenience method that will evolve to support all success codes returned by future versions of the AOS API. |
| diag() | This method can be used to retrieve a specific error code delivered by the Web Service Producer. It is not recommended to use it to determine success or failure; the preferred approach is to invoke 'success()' first and, if it returns False, call 'diag()' to retrieve the error code. |

AOSConnection(string username, string password, string server, boolean secure, boolean obeyproxy, boolean prettylinks, int port, AOSHeaders headers, boolean debug)

| | |
|--------------------|---|
| username | AAA username; same as when using WebView. |
| password | AAA user password |
| server | The address of the device to connect to. |
| secure | When True, SSL connections will be used. default value: True |
| obeyproxy | When True, system proxy settings will be followed. default value: True |
| prettylinks | When True, use semantically correct links as opposed to '?a=b&c=d...' default value: True |
| port | The port where the Web Service Producer is expected to be available; typically 80 or 443; however, -1 can be used to specify the use of the default port for secure/unsecure HTTP. default value: -1 |
| headers | An object used to inject additional headers in the request if necessary. default value: None |
| debug | When True, low-level GET, POST, PUT and DELETE commands will be displayed in the current terminal. default value: False |

AOSHeaders(Dict config)

config

A a dictionary that contains the current configuration:

if config['json'] is True, then a mime-type of vnd.alcatellucentaos+json will be requested; if it is False, then vnd.alcatellucentaos+xml will be requested;

config['api'] will be used to specify a given version of the API. Since the implementer can specify their own header object, they are free to create their own object (child of Dict or, preferably, child of AOSHeaders) which will provide its own additional headers in key->value form.

CLI Scripting

The AOS CLI relies on Bash scripting, it can be leveraged for creating CLI scripts without the need for an external tool. This Bash-based CLI allows users to perform high-level scripting work if necessary as given in the example below. This example illustrates simple example that creates multiple, non-contiguous, through the use of loops and variables. For instance:

```
#!/bin/bash
for vlanid in 1 2 3 4 10 15; do
vlan $vlanid
done
```

Since the existing CLI infrastructure is being leveraged, the CLI's own security model is followed (Bash already authorizes commands based on partition management).

Quoting and escaping

Quotes (') and double quotes (") are used to enclose literal strings. Single quotes provide the most literal guard whereas double quotes will expand "\$" variables. Due to this behavior, entering the text below will display "Hello" on a first row of the terminal, followed by "World" on the next row.

```
echo 'Hello,<Return>
World'<Return>
```

Because literal mode single quotes were used pressing <Return> simply added that key's code to its literal string. Literal mode was exited with the closing single quote, which is why the second <Return> submitted the command to Bash.

Backslash (\) is a continuation character. This means that the current line is continued on the next line. The example below will display "Hello World" on a single row:

```
echo Hello,\<Return>
World<Return>
```

HEREDOC (<<) is a form of I/O redirection that will feed a whole block to executables. HEREDOC takes a parameter and that parameter will be used by Bash to find the end of this pseudo I/O stream.

For instance, entering as root:

```
wall <<EOB<Return>
Hello,<Return>
World<Return>
EOB<Return>
```

will display the following on every logged in user's terminal:

```
Broadcast message from root (<Date>):
Hello,
World
```

The example above indicated to Bash a block of text was begun and that it would end when **EOB** was encountered at the beginning of a line.

Variables and functions

Variables

The asterisk character ('*') and the question mark have very specific meanings in Bash. The asterisk character can be used to replace an arbitrary number of characters of a command with a file name. This file needs to be referenced in a way that lets Bash find it. For instance, the following will list all the files found in the current directory that begin with the letter 'a' and end with the letter 'c'.

```
-> ls a*c
```

Similarly, the question mark will be replaced by a single character. Therefore, the following will list all files, in the current directory, that are three characters long, begin with the letter 'a' and end with the letter 'c'. Three characters long because '?' can only be replaced by a single character.

```
-> ls a?c
```

The dollar sign prefix is used to name variables. Assigning a value to a variable is done without the dollar sign prefix as shown below.

```
-> A="hello there"  
-> echo $A  
hello there
```

Variables can be used in CLI commands. For instance:

```
-> MYIF=192.168.1.1  
-> ip interface $MYIF  
-> show ip interface $MYIF
```

Functions

A function is a piece of code that can be reused after creating it. It can take parameters and return a diagnostic value. As a simple example is there's a need to repetitively create VLANs with similar parameters a function can be used to avoid having to specify these parameters every time.

To create a function, type its name followed by a pair of parenthesis and an opening curly brace. To complete the function definition, enter a closing curly brace. The body of the function will go between both curly braces, the function can then be run by entering its name as in the example below:

```
function myvlans()  
{  
  
}
```

To handle parameters within the function, positional parameters are used. For instance the following will create VLAN 5:

```
function myvlans()  
{  
    vlan $1  
}  
  
-> myvlans 5
```

Additional functionality can be added. As an example the function can be enhanced to handle cases when the user forgets to pass a parameter.

```
function myvlans()
{
    if [ $# -lt 1 ]; then
        echo "Please provide a paramater"
    else
        vlan $1
    fi
}

-> myvlans
```

This will display an error message because `$#`, which represents the number of arguments that were passed to the function, is less than ("`-lt`") one.

`Shift` can be used to cycle through a parameter list so that multiple parameters can be used with a function. The example below creates each VLAN using the "`vlan`" command. Every parameter will end up being seen as "parameter 1" due to the "`shift`" command. `Shift` moves all the positional parameters down by one every time it is invoked as in the example below:

```
function myvlans()
{
    while [ "$1" != "" ]; do
        vlan $1
        shift
    done
}

-> myvlans 5 6 7
```

Now, the script will "`shift`" the parameters, cycling through them:

```
$1="5", $2="6", $3="7"
> shift
$1="6", $2="7"
> shift
$1="7"
```

Additional functionality can be added to check that a VLAN was successfully created before moving on to the next one. This can be done using the previous command's return code which is stored in `$?` , for instance:

```
function myvlans()
{
    while [ "$1" != "" ]; do
        vlan $1
        if [ $? -ne 0 ]; do
            echo "Error!"
            return 1
        done
        shift
    done
}

-> myvlans 5 6 7
```

If "`vlan $1`" returned a value other than "`0`" which traditionally denotes success, the script returns immediately.

The `$_` represents the most recently used parameter. For instance, the following would result in VLAN 5 being created and then deleted:

```
vlan 5
no vlan $_
```

Adding user interaction

To enhance a function even further user interaction can be added. As an example, to have the function prompt the user for information the **read** command can be used to read user input as in the example below:

```
function myvlans()
{
    echo -n "Enter VLAN id: "
    read vlanid
    if [ "$vlanid" -eq "" ]; do
        echo "No VLAN ID entered..."
        return 1
    fi
    vlan $vlanid
}
```

CLI Tools

Shell-based scripting is only one aspect of the programmability of the AOS CLI. Specialized tools such as **grep** can also be invoked to refine the behavior of CLI commands. Additionally, **awk** offers a powerful syntax for advanced users.

The following is a list of some of the more common tools available in AOS:

- Page/search in current output/file: **more**, **less**
- Search/Filter files, output on strings, regular expressions: **egrep**, **fgrep**, **grep**
- Filter file/output: **cat**, **head**, **tail**
- Input parser (Can be used in conjunction with other commands such as 'find' or 'cat'): **sed**
- Count words/line/characters in file/current output: **wc**
- Evaluate arbitrary expressions (Bash built-in evaluation engine): **expr**
- Search for files: "find (based on name/wildcard, file type, access date, etc.). Combined with **xargs** or using built-in **-exec** can be used in conjunction with **grep**, etc.
- Compare files: **cmp**, **diff**

awk

As mentioned earlier, **awk** is scripting language in its own right. Here is a sample **awk** script that can be used to filter output based on current grouping. The **show ip routes** command produces the following output:

```
+ = Equal cost multipath routes? Total 25886 routes
```

| Dest Address | Gateway Addr | Age | Protocol |
|--------------|--------------|----------|----------|
| 1.1.1.1/32 | +10.1.12.1 | 02:19:54 | OSPF |
| | +10.2.12.1 | 02:19:54 | OSPF |
| | +10.3.12.1 | 02:19:54 | OSPF |
| | +10.4.12.1 | 02:19:54 | OSPF |
| 1.1.1.2/32 | 10.1.22.100 | 02:19:54 | OSPF |
| 1.1.1.3/32 | +10.11.23.3 | 02:19:42 | OSPF |
| | +10.12.23.3 | 02:19:54 | OSPF |
| | +10.13.23.3 | 02:19:54 | OSPF |
| | +10.14.23.3 | 02:19:42 | OSPF |
| 1.1.1.4/32 | 10.1.24.4 | 02:19:54 | OSPF |

If we use the **grep** command we can extract just the first line as in the following example:

```
-> show ip routes | grep "1.1.1.3/32"
1.1.1.3/32      +10.11.23.3      02:19:42      OSPF
```

Using **awk** the command output can be filtered more precisely. The following is a script that would perform this task:

```
awk -v pattern="$1" 'BEGIN {
    # This will be our flag:
    # are we currently reading desired block of info?
    INBLOCK = 0
}
{
    # Is first field not empty?
    # (when it is, number of fields (NF) is just 3)
    if (NF == 4) {
        # Check whether our string is found in column 3
        if ((p = index($0, pattern)) == 3) {
            INBLOCK = 1
        }
        else {
            INBLOCK = 0
        }
    }
    # If in block, display line
    if (INBLOCK == 1) {
        print $0
    }
}'
```

This script can then be easily turned into a standalone shell script by storing it in **/flash** as **filter.sh** and sourcing it using the **."** prefix syntax. The script can then be used to filter the output as shown below:

```
-> show ip routes | . /flash/filter.sh 1.1.1.3/32

1.1.1.3/32      +10.11.23.3      02:19:42      OSPF
                  +10.12.23.3      02:19:54      OSPF
                  +10.13.23.3      02:19:54      OSPF
                  +10.14.23.3      02:19:42      OSPF
```


OpenFlow Specifications

| | |
|---|------------------------|
| Platforms Supported | OmniSwitch 6860, 6860E |
| Modes Supported | Normal Hybrid (API) |
| Versions Supported | 1.0 1.3.1 |
| Maximum number of logical switches | 3 |
| Maximum number of controllers per logical switch | 3 |
| Support for Virtual Chassis | Not Supported |
| Maximum number of logical switches in Hybrid mode | 1 |

OpenFlow Agent Overview

OpenFlow is a communications interface defined between the control and forwarding layers that is used in a Software Defined Network (SDN). OpenFlow essentially separates the control plane and the data plane in the . Traditionally, es and routers have made decisions on where packets should travel based on rules local to the device. With OpenFlow, only the data plane exists on the itself, and all control decisions are communicated to the from a central Controller. If the device receives a packet for which it has no flow information, it sends the packet to the Controller for inspection, and the Controller determines where that packet should be sent based on QoS-type rules configured by the user (drop the packets to create a fire-wall, pass the packets to a specific port to perform load balancing, prioritize packets, etc).

The Omni can operate in AOS or OpenFlow mode, including a modified OpenFlow mode known as Hybrid mode. AOS will designate the ports managed/controlled by AOS or by OpenFlow on a per-port basis. By default, ports are managed/controlled by AOS.

The following are the key components available on an Omni for OpenFlow support.

OpenFlow Logical

An OpenFlow logical consists of a portion of the 's resources that are managed by an OpenFlow Controller (or set of Controllers) via the OpenFlow Agent. Logical es can be configured on an Omni, with each logical supporting separate controllers. A logical has a VLAN, physical ports, and/or link aggregate ports assigned to it. All packets received on these ports are forwarded directly to the Openflow agent. Spanning tree and source learning do not operate on OpenFlow assigned ports.

OpenFlow Normal Mode

In Normal mode, the logical operates as per the OpenFlow standards.

OpenFlow Hybrid (API) Mode

In Hybrid mode, logical acts as an interface through which the Controller may insert flows. These flows are treated as QoS policy entries and offer the same functionality. A Hybrid logical operates on all ports, link aggregates, and VLANs not assigned to other OpenFlow logical es.

Quick Steps to Configure OpenFlow Agent

Follow the steps in this section for a quick tutorial on how to configure an OpenFlow Agent on the Omni A logical in Hybrid mode does not have a VLAN or interface configured.

1 Create the logical and configure the mode

```
-> openflow logical- v1 mode normal version 1.3.1 vlan 5
-> openflow logical- v2 mode api
```

2 Assign a controller to the logical

```
-> openflow logical- v1 controller 1.1.1.1
-> openflow logical- v2 controller 2.2.2.2
```

3 Assign interfaces to the logical

```
-> openflow logical- v1 interfaces port 1/1/3
```

4 Verify the configuration

```
-> show openflow logical-
```

| Logical | State | Admin Mode | Versions | VLAN | Ctrlrs | Intf | Flows |
|---------|-------|------------|-----------|------|--------|------|-------|
| v1 | Ena | Norm | 1.3.1 | 5 | 1 | 1 | 5 |
| v2 | Ena | API | 1.0 1.3.1 | N/A | 1 | 56 | 0 |

```
-> show openflow logical- controllers
```

| Logical | Controller | Role | State | Admin State | Oper |
|---------|--------------|-------|-------|-------------|------|
| v1 | 1.1.1.1:6633 | Equal | Ena | Connect | |
| v2 | 2.2.2.2:6633 | Equal | Ena | Backoff | |

```
-> show openflow logical- interfaces
```

| Logical | Interface | Mode |
|---------|-----------|------|
| v1 | 1/1/3 | Norm |
| v2 | 1/1/1 | API |
| v2 | 1/1/2 | API |
| v2 | 1/1/4 | API |
| v2 | 1/1/5 | API |
| v2 | 1/1/6 | API |
| v2 | 1/1/7 | API |

```
(output truncated)
```


11 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 AOS Release 8 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 11-9](#).
- Selecting an NTP server for the NTP client and modifying settings for communicating with the server. See [“NTP Servers” on page 11-10](#).
- Enabling authentication in NTP negotiations. See [“Using Authentication” on page 11-12](#).

NTP Specifications

| | |
|--|----------------------------|
| Platforms Supported | OmniSwitch 6860, 6860E |
| RFCs supported | 1305–Network Time Protocol |
| NTP Key File Location | /flash/network |
| Maximum number of NTP servers per client | 12 |

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 status** 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 2013 17:44:54 (UTC)
Last NTP update:      THU SEP 15 2013 17:30:54
Client mode:          enabled
Broadcast client mode: disabled
Broadcast delay (microseconds): 4000
```


NTP Overview

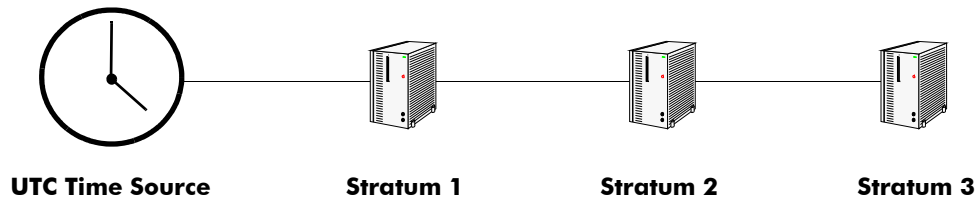
Network Time Protocol (NTP) is used to synchronize the time of a computer client or server to another server or reference time source, such as a radio or satellite receiver. It provides client time accuracies within a millisecond on LANs, and up to a few tens of milliseconds on WANs relative to a primary 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 Positioning 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.

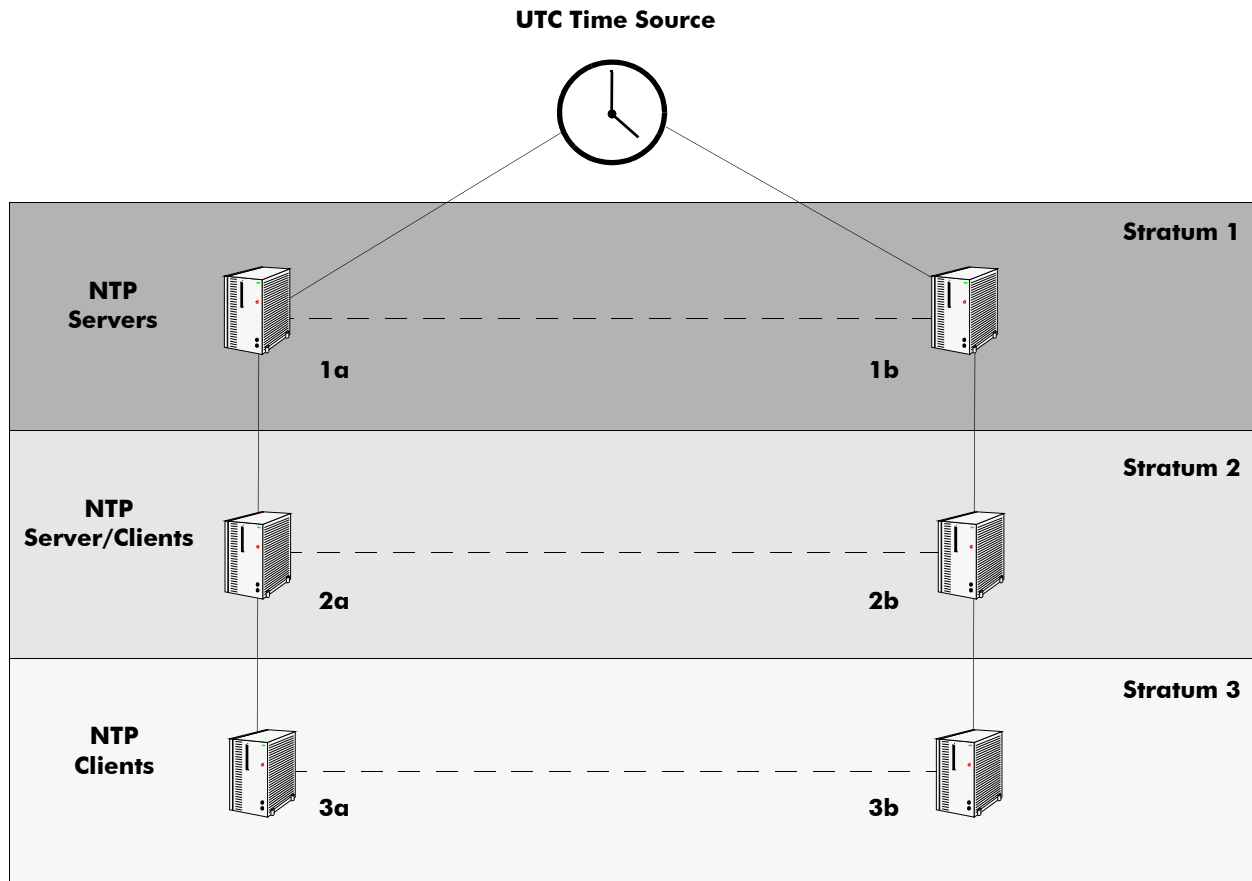
Note. It is not required that NTP be connected to an officially recognized time source (for example, a radio clock). NTP can use any time source to synchronize time in the network.

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 gradation 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 11-8.](#))

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 AOS Release 8 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 11-3.

12 Managing Automatic Remote Configuration Download

The Automatic Remote Configuration feature enables:

- the automatic upgrade of firmware and/or configuration of an OmniSwitch without user intervention.
- the automated configuration of the switch on bootup, when the switch is connected to the network for the first time.
- the automatic download and installation of the critical configuration bootup and image files.

In This Chapter

This chapter describes the Automatic Remote Configuration on OmniSwitch. The sections in this chapter are:

- [“Automatic Remote Configuration Specifications” on page 12-2](#)
- [“Automatic Remote Configuration Defaults” on page 12-3](#)
- [“Quick Steps for Automatic Remote Configuration” on page 12-4](#)
- [“Overview” on page 12-5](#)
- [“Interaction With Other Features” on page 12-8](#)
- [“Automatic Remote Configuration Download Process” on page 12-9](#)
- [“Download Component Files” on page 12-12](#)
- [“DHCP Client Auto-Configuration Process” on page 12-16](#)
- [“Nearest-Edge Mode Operation” on page 12-17](#)
- [“LACP Auto Detection and Automatic Link Aggregate Association” on page 12-19](#)
- [“Troubleshooting” on page 12-21](#)

Automatic Remote Configuration Specifications

| | |
|--|--|
| Platforms Supported | OmniSwitch 6860, 6860E |
| DHCP Specifications | DHCP Server required DHCP Client on VLAN 1 or VLAN 127 (DHCP client on VLAN 127 works on uplink ports only), LLDP Management VLAN |
| File Servers | TFTP FTP/SFTP |
| Clients supported | TFTP FTP/SFTP |
| Instruction file | Maximum length of: <ul style="list-style-type: none">• Pathname: 255 characters• Filename: 63 characters |
| Maximum length of username for FTP/SFTP file server. | 15 characters |
| Unsupported Features: | <ul style="list-style-type: none">• ISSU and IPv6 are not supported.• Upgrade of uboot, miniboot, or FPGA files is not supported. |

Automatic Remote Configuration Defaults

| Description | Default |
|---|--|
| Management VLAN Untagged Management VLAN | VLAN 1 |
| DHCP broadcast VLAN 802.1q tagged VLAN | VLAN 127 |
| Default Auto Link Aggregate Creation | Between VLAN 1 and VLAN 127 |
| Instruction file | Location: TFTP Server File name: *.alu (* represents any instruction filename) Download location: /flash directory Downloaded as a temporary file. |
| Configuration file | File name: Any name Location: FTP/SFTP/TFTP Server Download location: /flash/working directory |
| Debug configuration file | File name: AlcatelDebug.cfg Location: FTP/SFTP/TFTP Server Download location: /flash/working directory |
| Script file | File name: Any name Location: FTP/SFTP/TFTP Server Download location: /flash/working directory |
| Firmware version | OS_*_*_R01 (*_* represents version number) |
| Firmware or image files | File name extension: *.img (* represents image filename) Location: FTP/SFTP/TFTP Server Download location: /flash/working directory |
| File download server | Primary FTP/SFTP/TFTP Server |
| Backup server for file download | Secondary FTP/SFTP/TFTP Server |
| Password for FTP/SFTP Server | Same as username |

Quick Steps for Automatic Remote Configuration

- 1 Configure the DHCP server in the network to provide IP address, gateway, and TFTP server addresses to the OmniSwitch DHCP client.
- 2 Store the instruction file on the TFTP server.
- 3 Store the configuration, image, and script files on the primary and/or secondary FTP/SFTP servers.
- 4 When the OmniSwitch is integrated in to the network as a new device with no **vcboot.cfg** file in the *working* directory, the automatic remote configuration process is initiated.
- 5 A DHCP client is automatically configured on the OmniSwitch. The OmniSwitch obtains IP address information, TFTP server address, instruction file name, and location from the DHCP server through the DHCP client.
- 6 The OmniSwitch downloads the instruction file from the TFTP server. The instruction file contains the file names and file locations of the configuration, image, and script files.
- 7 The OmniSwitch downloads the image files from the FTP/SFTP server if necessary.
- 8 The OmniSwitch downloads the configuration file from the FTP/SFTP server, if available, and saves it as the **vcboot.cfg** file in the **/flash/working/** directory. If no script file is downloaded, the switch reboots applying the downloaded configuration file and the automatic configuration process is complete.
- 9 The OmniSwitch downloads the script file, if available, from the FTP/SFTP server and runs the commands in the script file.

Note.

- If the script file is not specified in the instruction file, or if it is not properly downloaded, then the Remote Configuration Manager software automatically initiates a **reload from working no rollback-timeout** command after firmware or bootup configuration files are downloaded.
 - If a **write memory** command is used in the script file, then it overwrites the **vcboot.cfg** file. Hence, if the script file is downloaded along with the bootup configuration file, then the script file must not contain the **write memory** command.
 - If a **vcboot.cfg** is already present in the **working** directory of the switch, Automatic Remote Configuration Download does not occur.
-

Overview

The Automatic Remote Configuration feature provides the advantage of automatic download and installation of critical configuration and image files at initial bootup or when firmware upgrade is required for the OmniSwitch.

Automatic Remote Configuration download occurs when:

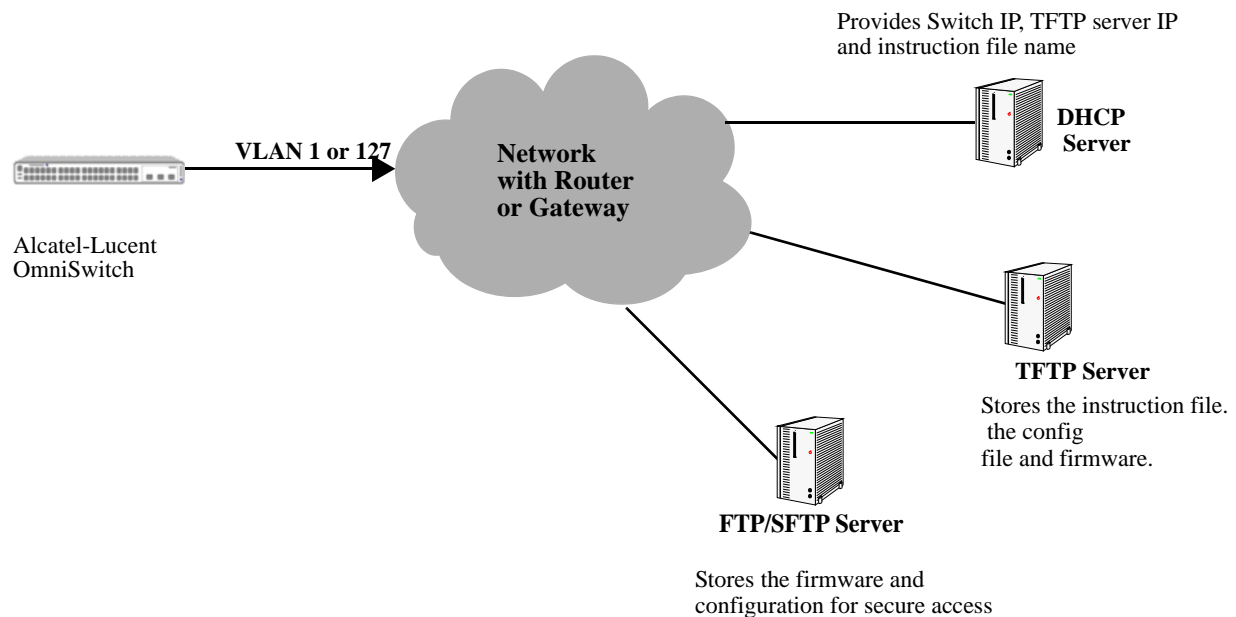
- There is no bootup configuration file (**vcboot.cfg**) in the *working* directory of the switch.
- During a takeover or reboot on the new Primary unit or CMM.
- The initialization process of the switch is complete and the network interfaces or ports are ready.
- There is connectivity with a DHCP server through the default VLAN 1, the Nearest-Edge mode management VLAN, or through a tagged VLAN 127.
- There is connectivity with TFTP file server.

The following sections provide more information about the automatic configuration and download process.

Basic Operation

Automatic remote configuration process is initialized on the OmniSwitch if the **vcboot.cfg** file is not found in the *working* directory of the switch.

The following illustration shows the basic setup required for Automatic Remote Configuration Download operation.



Basic Network Components for Automatic Remote Configuration Download

Network Components

The network components required for the Automatic Remote Configuration download process are:

- DHCP server (mandatory)
- TFTP file server (mandatory)
- Primary FTP/SFTP server (mandatory)
- Secondary FTP/SFTP server (optional)
- Management Switch (only required for Nearest-Edge Mode)

Information Provided by DHCP Server

When the network interfaces or ports on the switch are ready, a DHCP client is automatically configured. For details on the DHCP client auto-configuration, see [“DHCP Client Auto-Configuration Process” on page 12-16](#). The following information is acquired from the DHCP server, after a connection is established:

- IP address of the Network Gateway or Router.
- TFTP file server address.
- Instruction file name and location.
- Dynamic IP address for the OmniSwitch (valid only for initial bootup process).

Information Provided by Instruction File

The TFTP server address information is received from the DHCP server. The OmniSwitch downloads the instruction file from the TFTP server. The instruction file provides the following information:

- Firmware version and file location.
- Configuration file name and location.
- Debug configuration file name and location.
- Script file name and location.
- Primary FTP/SFTP file server address / type / username.
- Secondary FTP/SFTP file server address / type / username.

For more details on all the component files downloaded during the automatic remote configuration download process, see - [“Download Component Files” on page 12-12](#).

File Servers and Download Process

The download process from the file servers is as follows:

- 1 The username required to connect to the FTP/SFTP enabled servers is provided in the instruction file. The password required to connect to the servers is same as the username.
- 2 The required files mentioned in the instruction file are downloaded from the primary FTP/SFTP file server.
- 3 If the configuration, debug and script file names are specified in the instruction file, then they are downloaded to the **/flash/working** directory of the switch.
- 4 The Remote Configuration Manager now compares the current firmware version on the switch to the one mentioned in the instruction file. If the firmware version is different, then firmware upgrade is performed.
- 5 The new firmware or image files are downloaded to the *working* directory of the switch.

Note. If the primary server is down or if there is any failure in downloading the files from the primary file server, then a connection is established with the secondary file server. The secondary file server is used for file download.

- 6 All the required files are downloaded.

Note. If a specific filename (for firmware and **configuration/debug/script** files) is not found, an error is logged. The download process continues with the next available file. File transfer is tried three times and if file transfer still fails, an error is logged, and download process is stopped. In such instances, the *working* folder of the switch will contain an incomplete set of image files, configuration, debug, or script files. For details on troubleshooting under such instances, see - [“Nearest-Edge Mode Operation” on page 12-17](#)

- 7 Now, the DHCP client configured on the related VLAN is removed.
- 8 The script file is downloaded and the commands in the script file are run. All the commands in the script file are implemented on the switch in the order specified.

For other detailed steps that are part of the automatic remote configuration download process, see [“Automatic Remote Configuration Download Process” on page 12-9](#)

LED Status

The LED status during different stages of the Automatic Remote Configuration download process is as follows:

- DHCP phase: OK1 LED is flashing green
- DHCP lease obtained: OK1 LED is solid green
- DHCP phase stopped by console login: OK1 LED is solid green.

Interaction With Other Features

This section contains important information about how other OmniSwitch features interact with Automatic Remote Configuration. Refer to the specific sections if required, to get detailed information about the feature interaction process.

UDP/DHCP Relay

Interaction with UDP/DHCP Relay is required for the following processes, to support Automatic Remote Configuration:

- All the DHCP responses from the DHCP server are processed. The IP address, mask, and gateway details are processed
- To acquire **Option (66) and Option(67)** information - the TFTP Server name and Boot file name are retrieved.

For details on DHCP interaction see the section [“DHCP Client Auto-Configuration Process” on page 12-16](#)

802.1Q

802.1Q tagging is applied on VLAN 127 or the Management VLAN.

LLDP

In Nearest-Edge mode operation LLDP packets carry and provide the advertised VLAN ID to the Access OmniSwitches running in Auto Remote Configuration mode.

Dynamic Link Aggregation (LACP)

Interaction with LACP is required for the following processes, to support Automatic Remote Configuration:

- To detect LACP PDU from the peer device on uplink ports
- To enable the auto link aggregate creation after receiving LACP message
- The link aggregate is associated as a tagged member of VLAN 127 and VLAN 1.

On completion of the Automatic Download and configuration process, the automatic link aggregate is disabled and all port associations are deleted.

Automatic Remote Configuration Download Process

The automatic remote configuration process is initialized when an OmniSwitch is integrated in to the network as a new device or when a firmware and configuration upgrade is required.

If the automatic configuration download process is not performed completely on the switch, manual intervention is required. For details on troubleshooting techniques under such instances, see [“Nearest-Edge Mode Operation” on page 12-17](#)

The detailed process of Automatic Remote Configuration Download performed on the OmniSwitch is as follows:

- 1 When the switch is integrated in to the network as a new device with no **vcboot.cfg** file, then Automatic Remote Configuration is performed on the switch.
- 2 The Remote Configuration Manager on OmniSwitch configures a link aggregate automatically when a LACP PDU is detected on the uplink ports on the switch during Automatic Remote Configuration. For details, see the following section [“LACP Auto Detection and Automatic Link Aggregate Association” on page 12-19](#).
- 3 A DHCP client is automatically configured on VLAN 1, Management VLAN, and VLAN 127 at switch boot up. OmniSwitch then uses different methods of DHCP client configuration until connection to a DHCP Server is obtained. For details, see the following section [“DHCP Client Auto-Configuration Process” on page 12-16](#)
- 4 The DHCP client obtains the switch IP address information from the DHCP server.
- 5 The DHCP client obtains the TFTP server IP address from the DHCP server using Option (66).
- 6 The DHCP client obtains the instruction file name and location from the DHCP server using Option (67).
- 7 SSH access is automatically enabled to allow remote access in case the automatic configuration process fails.
- 8 The instruction file with the **.alu** extension is downloaded from the TFTP server to the **/flash/working** directory of the OmniSwitch.
- 9 If available, the configuration, script, and images files are downloaded from the FTP or SFTP servers. The password used to connect to the FTP/SFTP servers is same as the username.
- 10 If available, the switch compares the firmware version available on the switch with the firmware version in the instruction file. If the firmware versions are different, then the new firmware is downloaded in to the **/flash/working** directory.
- 11 If available, the downloaded configuration file is saved as the **vcboot.cfg** file in the **/flash/working** directory and the switch is rebooted completing the auto configuration process (a reboot occurs only if no script file is downloaded).
- 12 If available, commands in the script file are run and the DHCP client configuration is automatically removed.

Process Illustration

The following flowchart represents the automatic remote configuration download process in detail.

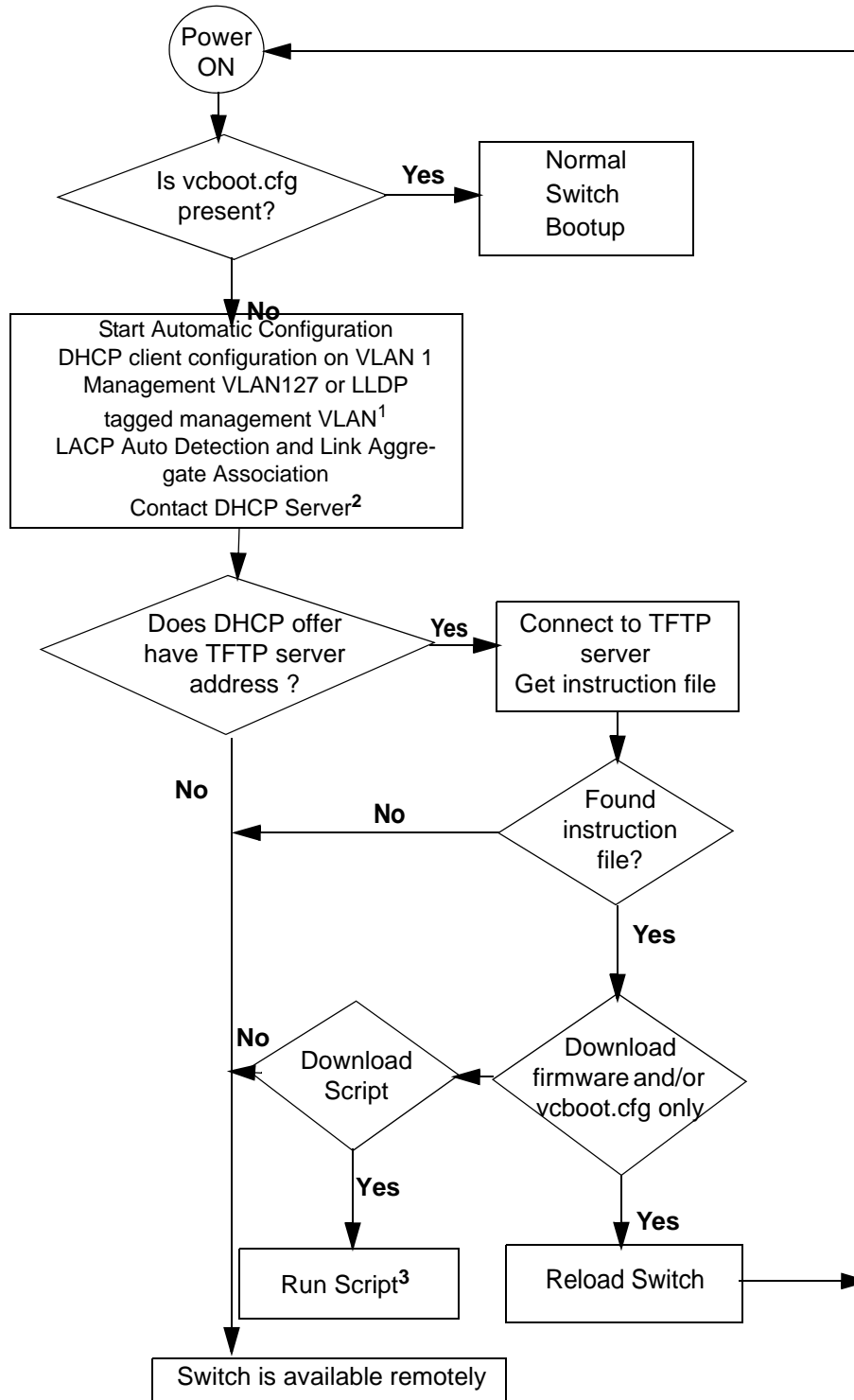


Illustration of Automatic Remote Configuration Process

Additional Process Notes

1 Once the switch obtains an IP interface from the DHCP server, remote access through SSH is automatically configured to allow remote access in case of any download errors during the Auto Configuration process.

Note. It is not recommended to have the **write memory** command in the script file if a configuration file is downloaded. This causes the **vcboot.cfg** file to be overwritten with the commands in the script file.

2 After the successful download of the script file, the DHCP IP interface is automatically deleted. However, SSH access remains enabled. Use the **no aaa authentication ssh** command to disable SSH connectivity if desired.

Download Component Files

This section provides the details of the files downloaded and how they are utilized during the automatic configuration process. The main component files are:

- **Instruction file**—The instruction file is the initial file required for the automatic remote configuration process to occur. The instruction file is stored in the TFTP server with the **.alu** extension. For further details, see [“Instruction File” on page 12-12](#)
- **Firmware upgrade files**—The firmware files or image files differ for different OmniSwitch platforms. These image files contain executable code, which provides support for the system, Ethernet ports, and network functions. For further details, see [“Firmware Upgrade Files” on page 12-14](#)
- **Bootup configuration file**—The file contains bootup configuration information for the switch. The bootup configuration file stores the network configuration parameters. For further details, see [“Bootup Configuration File” on page 12-14](#)
- **Debug Configuration file**—The debug configuration file stores the default debug configuration information. For further details, see [“Debug Configuration File” on page 12-14](#)
- **Script file**—The script file consists of commands to be performed on the switch so that appropriate actions can be taken on the downloaded files. For further details, see [“Script File” on page 12-15](#)

Instruction File

The instruction file is the initial file required for automatic remote configuration process to occur. The instruction file is stored in the TFTP server with the **.alu** extension.

The instruction file contains user information such as switch ID, file version, firmware version, image file names and location, configuration file (**vcboot.cfg**) name and location, script file name and location, FTP/SFTP server IP address, username and password to connect to the FTP/SFTP server.

The TFTP server IP address and instruction filename details are received from the DHCP server by the DHCP client on the OmniSwitch.

The instruction file is downloaded from the TFTP server and stored in the **/flash/working** directory of the switch.

Note.

- If an error or failure occurs during the file transfer, the transfer process is retried up to three times. If file transfer and download are not successful, the automatic remote configuration process is halted and the switch is made available remotely using SSH.
 - All contents of the instruction file are stored in the switch log (**swlog.log**) file as evidence of the last Automatic Remote Configuration download.
-

Instruction File Syntax

The instruction file is a text file containing the following information:

| | |
|---|---|
| Header | Contains user information such as switch ID, file version, and so on. Header text is a type of comment. |
| Comments | Comments provide additional information for better user readability. These lines are ignored during the remote configuration download process. |
| Firmware version and file location | Image files required for firmware upgrade. |
| Configuration file name and location | The file containing the configuration for the switch, this file is saved as the vcboot.cfg file in the /flash/working directory. |
| Debug file name and location | The AlcatelDebug.cfg containing additional debug configuration commands |
| Script file name and location | The script file containing commands to be implemented on the switch. |
| Primary file server address/protocol/username | The primary file server from which the required files are downloaded. The specified protocol and username is used for the download. |
| Secondary file server address/protocol/username | The secondary file server from which the required files are downloaded if the connection to primary file server fails. The specified protocol and username are used for the download. |

Example

The instruction file has the Keyword:Value format as shown below:

```
! Alcatel-Lucent OmniSwitch OS6860 - Instruction file version 1.2.1
! Firmware version
Firmware version:OS_8_1_1_R01
Firmware location:/home/ftpboot/firmware
! Configuration file
Config filename:boot_OS6860.cfg
Config location:/home/ftpboot/config
! Debug file
Debug filename:AlcatelDebug.cfg
Debug location:/home/ftpboot/debug
! Script File
Script filename:OS6860_script.txt
Script location:/home/ftpboot/scripts
! Primary file Server
Primary server:10.200.100.112
Primary protocol:FTP
Primary user:admin
! Secondary file Server
Secondary server:10.200.110.111
Secondary protocol:SFTP
Secondary user:admin
```

Instruction File Usage Guidelines

- The instruction file is case sensitive and can contain only the keywords provided in the instruction file output example.
- The keywords can be placed in any order.
- If the Keyword:Value format is incorrect, the information on that line is discarded.
- Firmware version must be provided in the format as specified in the example.
- Pathnames provided must contain the complete path to the file location.
- If any file is not required, the value is provided as “None”. For example, if a debug configuration file is not required to be downloaded, the instruction file syntax is as follows:

```
Debug filename:None
Debug location:None
```
- The header line is the first line of the instruction file and begins with “!” character.
- Header line contents are logged to the switch log along with the other contents of the instruction file.
- The header and comment lines begin with “!” character.

Firmware Upgrade Files

Firmware files are also known as image files. These files have the **.img** extension.

Firmware files may be different based on the OmniSwitch platform. The relevant firmware files are downloaded from the location mentioned in the instruction file. The filenames of the firmware files must exactly match the files which are to be downloaded. The filenames are in the ***.img** format. Modified filenames are not recognized.

Details about the different firmware files and file names can be found in [“Managing System Files” on page 2-1](#).

Firmware files are downloaded only when the firmware version in the instruction file is higher than the firmware version present on the switch.

Bootup Configuration File

The bootup configuration (**vcboot.cfg**) file is not present during the initial bootup process when a new switch is integrated in to the network. The **vcboot.cfg** file is automatically generated and stored in the **/flash/working** directory when a **write memory** command is issued.

During the automatic remote configuration process, the bootup configuration file is downloaded from the FTP/SFTP server and stored as **vcboot.cfg** in the **/flash/working** directory of the switch.

If no script file is downloaded, the switch boots up normally according to the configurations specified in the **vcboot.cfg** file when the remote configuration download process is completed.

Debug Configuration File

The debug configuration file is used for setting specific OmniSwitch settings and must only be used as directed by Service and Support. During the automatic remote configuration process, the debug configuration file is downloaded with the filename **AlcatelDebug.cfg**.

Script File

The script file is downloaded and stored with the same name in the **/flash/working** directory. The script file contains the commands to be implemented on the switch after running the configuration file.

If a configuration file is not available, the script file can be used to configure the switch dynamically without a **vcboot.cfg** file.

Script File Example

```
vlan 100 enable name "VLAN 100"  
vlan 100 members port 1/1/1 untagged  
write memory
```

Script File Usage Guidelines

- After the script file is downloaded the switch does not automatically reboot.
- If a **write memory** command is used in the script file, then it overwrites the **vcboot.cfg** file. Hence, the script file must not contain the **write memory** command if it is downloaded along with the configuration file.
- If any script file command fails, it is logged in to a file ***.err** (* is the script file name) in the **/flash** directory and the remaining commands are implemented.
- If the script file name mentioned in the instruction file is incorrect, then an error is logged in the switch log or **swlog.log** file.

DHCP Client Auto-Configuration Process

The automatic remote configuration download feature supports the following client configuration methods to obtain an initial dynamic IP address from the DHCP server:

- Static DHCP client on untagged VLAN 1
- Dynamic DHCP client on tagged VLAN 127
- Dynamic DHCP client on LLDP tagged Management VLAN
- Auto Link Aggregate Detection

OmniSwitch creates a DHCP Client interface on:

- the default untagged VLAN 1 and then on tagged VLAN 127 alternating between each, or
- the Management VLAN being advertised in the LLDP PDUs sent by the Management Switch configured in Nearest-Edge Mode.

If OmniSwitch receives LLDP PDUs with VLAN and port information from a Management switch in nearest edge mode, then the DHCP client interface is moved to user defined LLDP management VLAN on the network. See the [“” on page 12-16](#) for additional information.

The detailed process of DHCP client auto-configuration on an OmniSwitch is as follows:

- 1** At boot-up, the initial DHCP client starts with untagged VLAN 1. The DHCP client waits for 30 seconds for a DHCP lease.
- 2** If the lease is not obtained even after 30 seconds, the DHCP client is stopped on the untagged VLAN 1 and DHCP client is started on tagged VLAN 127. The DHCP client on tagged VLAN 127 waits for 30 seconds for a DHCP lease.
- 3** If the DHCP client does not get the lease in 30 seconds, DHCP client moves back to untagged VLAN 1 and this process continues until it gets the DHCP lease on any one of the two VLANs.
- 4** If during this process the switch receives an LLDP PDU advertising the management VLAN, the DHCP process will stop on VLANs 1 and 127 and begin on the management VLAN.

Note.

DHCP client is automatically stopped once a user logs in the switch through console port before getting the DHCP lease. Once the DHCP client gets the lease, the Remote Config process does not stop even if the user logs on to the switch through console port.

Nearest-Edge Mode Operation

In order for the network to propagate Nearest-Edge mode LLDP PDUs a Management Switch must be configured to send the LLDP PDUs with the Management VLAN information. Additionally, the peer switches are automatically configured to process the Nearest-Edge Mode LLDP PDU frames by the Automatic Configuration Download feature.

An OmniSwitch running the Automatic Remote Configuration feature is automatically enabled to process LLDP PDUs with the unique Nearest-Edge destination MAC address. In Nearest-Edge mode the Management OmniSwitch uses a unique MAC address when sending LLDP PDUs. The network OmniSwitch also looks for these unique packets to determine a Management VLAN. It then creates a DHCP client interface on that tagged VLAN.

LLDP Transmission from Management Switch

- The Management Switch is configured to use the Nearest-Edge Mode MAC address and is connected to the network using an untagged interface.
- LLDP is configured on the untagged port of the Management Switch so that the LLDP PDUs are sent with the Management VLAN information.
- The LLDP interval must not be set higher than 30 seconds (default).
- The Management Switch sends LLDP PDUs on the untagged interface with the MAC address of 01:20:DA:02:01:73.

LLDP Propagation through Network

These LLDP PDUs are propagated throughout the network as normal L2 multicast frames, eventually reaching the Access Switch.

LLDP Reception by Access Switch

The Automatic Configuration Download feature enables the processing of the Nearest-edge LLDP PDUs by default.

Nearest-Edge Mode Configuration Example

LLDP Nearest Edge Configuration

Automatic Remote Configuration feature requires learning Management VLAN ID from a centralized management switch. This VLAN ID information is distributed through LLDP message.

On the management switch, LLDP sends Port VLAN ID TLV to the special MAC address [01:20: DA: 02:01:73]. The CLI command for this functionality is as follows:

```
-> lldp nearest-edge mode {enable | disable}
```

This functionality also depends on the nearest bridge agent LLDPDU transmit mode. So LLDP sends a Port VLAN ID TLV when both of the below commands are executed:

```
-> lldp nearest-bridge chassis lldpdu tx-and-rx  
-> lldp nearest-edge mode enable
```

The LLDPDUs are sent on the untagged interface with the Nearest-edge MAC address and propagated throughout the network eventually reaching the Access Switch.

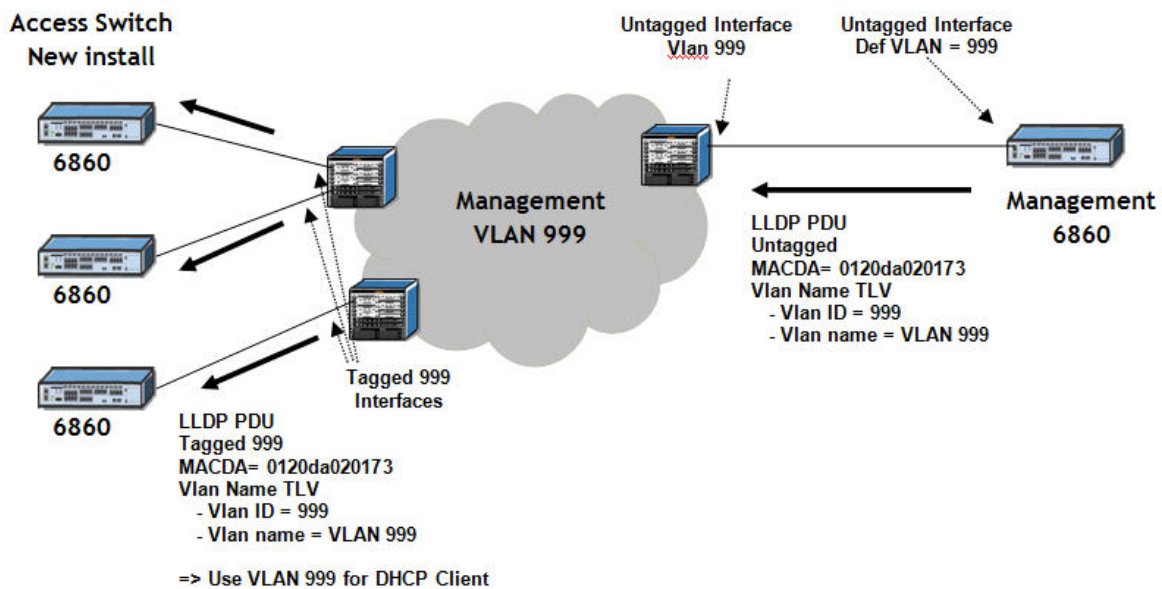
The Management Switch is connected to the network using an untagged interface and is configured to use the Nearest-edge Mode MAC address. LLDP is configured on the untagged port of the Management Switch so that the LLDP PDUs are sent with the Management VLAN information. The LLDP PDUs are sent on the untagged interface with the Nearest-edge MAC address and propagated throughout the network eventually reaching the Access Switch.

For example:

```
-> vlan 999 name "VLAN 999"
-> vlan 999 members port 1/1/1 untagged
```

Access Switch

When used in conjunction with the Automatic Remote Configuration feature no configuration is necessary on the Access OmniSwitches. Newly connected switches without a *vcboot.cfg* file receive the Nearest-Edge LLDP PDUs, discover the Management VLAN, tag the port with that VLAN ID, and create a DHCP client interface on the Management VLAN. This auto-configuration allows the DHCP client interface on the OmniSwitch to receive an IP address in the proper IP subnet.

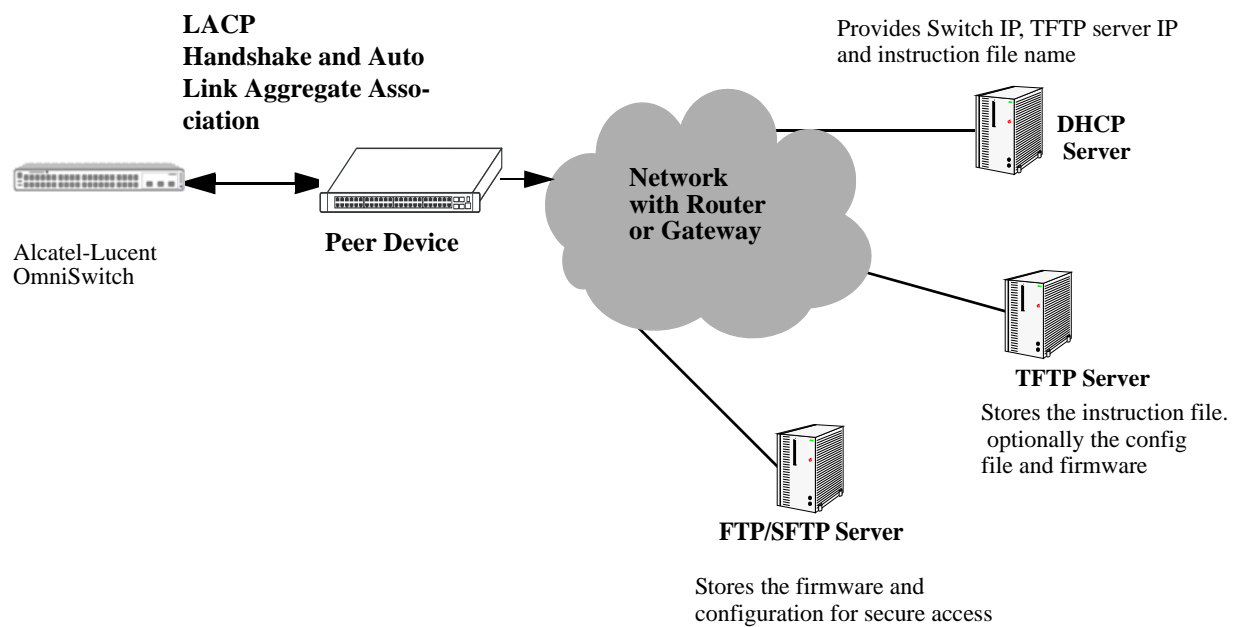


Example Nearest-Edge Configuration

LACP Auto Detection and Automatic Link Aggregate Association

DHCP Server Association and DHCP Client creation works on fixed ports. When an OmniSwitch is newly introduced to a network, an assigned peer network device detects this device as new. If the peer device has a link aggregate configuration on the detecting port, then it sends LACP PDU to the newly connected OmniSwitch. In such instances, LACP PDUs must be acknowledged by OmniSwitch. The Remote Configuration Manager on OmniSwitch detects any LACP PDUs on uplink ports and configures a link aggregate automatically during Automatic Remote Configuration.

The following diagram illustrates the different network components required for Auto Remote Configuration and LACP Auto Detection and Link Aggregate Association process



Network Components for LACP Auto Detection and Link Aggregate Association

LACP auto detection is enabled by default and operates only on the uplink ports on OmniSwitch during the Automatic Remote Configuration stage.

Note. This feature is supported only for uplink ports so it is required that the connectivity to the remote device is only through uplink ports.

- 1** When an OmniSwitch detects LACP PDUs from a remote peer connected through an uplink port, it configures that port as a LACP port and starts LACP handshake with the peer device.
- 2** The newly formed link aggregation is made a member of VLAN 127 and VLAN 1.
- 3** Once the remote configuration download is complete on this LACP port, the switch configuration file can automatically configure the required ports for the link aggregate.
- 4** After the process is completed, this automatic link aggregate and related associations are deleted.

Note. The LACP auto detection mode is not supported when the switch boots up in normal mode (non-remote configuration load mode). The LACP configuration at the peer device must not be changed once the automatic link aggregate is created using the parameters in the LACP PDU sent from the peer device.

Troubleshooting

Due to errors during download, the automatic configuration process can halt, or the file download process can be incomplete. The errors that occur during the automatic remote configuration download process are displayed on the switch command prompt and also stored in switch log or the **swlog.log** file.

The following section provides information on some of the common errors that can occur during the configuration download process and troubleshooting techniques to resolve these errors.

Error Resolution

If there are any issues downloading the required files for the auto configuration process the switch can be reached using the DHCP client IP address and the SSH protocol for manual intervention or configuration.

Server Connection Failure and File Download Errors

Manual download of component files is required when there is a failure in connecting to the servers or when all the component files are not downloaded during the automatic remote configuration download process.

Server connection failures can occur when:

- DHCP server is not reachable.
- TFTP server is not reachable.
- Primary and secondary servers are not reachable.

File download errors can occur when:

- Files are corrupted.
- File locations or names listed in the instruction file are incorrect.

Error Description Table

The following table provides information on the common server connection failures and file download errors that can occur during Automatic Remote Configuration:

| Error Type | Error | Description |
|---|---|--|
| User Login Auto-Config Abort | User logged in via console, Automatic Remote configuration is aborted. | DHCP client is automatically stopped only if a user logs in to the switch through console port before getting the DHCP lease. |
| TFTP Response Timeout | Instruction File not Downloaded and the Max try 3 For TFTP reached. | Instruction file not downloaded due to TFTP not reachable. |
| Primary/Secondary Server Connection | Download of file: <File name and pathname> from Primary Server Failed | File download failure from primary server. |
| | Starting download of file: <File name and pathname> from Secondary Server | |
| | Download Failed - <File name and pathname> using both Pri & Sec IP | File download failure from both primary and secondary server. |
| File Download and File Location Errors | Transfer error <File name and pathname> | File transfer failure. |
| | Download failed for configuration file <File name and pathname> | Configuration file download failure. |
| | Not all image files are downloaded | Some of the image files are not downloaded. |
| | Unable to download the firmware version | File location errors occur when the corresponding files are not available in the locations as mentioned in the instruction file. |
| | Unable to download boot config file | |
| | Unable to download AlcatelDebug.cfg | |
| | Unable to download script file | |

Script File Errors

The different types of script file errors and the troubleshooting techniques for such errors are as follows:

- If any script file command fails, it is logged in to a file ***.err** (* is the script file name) in the **/flash** directory and the remaining commands are implemented. In such an instance, check the ***.err** file. The script file commands can be manually implemented and debugged in the order specified in the script file.
- If the script file name mentioned in the instruction file is incorrect, then an error is logged in the switch log or **swlog.log** file. In such an instance, check the **swlog.log** file. The script file can be downloaded manually from the FTP/SFTP servers and implemented onto the OmniSwitch.

Error Description Table

The following error description table provides information about some of the common script file errors that occur during Automatic Remote Configuration:

| Error Type | Error | Description |
|------------------------------------|--|---|
| Script File Download | Download of Script file from Primary Server Failed | Script file cannot be downloaded from the primary server. |
| | Starting download of Script file: <File name and pathname> from Secondary Server Download failed - <File name and pathname> using Pri and Sec IP | Script file cannot be downloaded from both primary and secondary server. |
| Script File Command Failure | Unable to remove Instruction file <File name and pathname> | Instruction file cannot be removed from flash due to error in running the script file commands. |
| | Error in executing the downloaded script file | The downloaded script file cannot be run. |

13 Configuring Virtual Chassis

A Virtual Chassis (VC) is a group of chassis managed through a single management IP address. It provides both node level and link level redundancy for layer 2 and layer 3 services and protocols acting as a single device. Up to eight chassis can be combined into a single virtual chassis. A VC can easily expand switching capacity simply by adding additional chassis to the VC. For example, a deployment can start with a VC composed of two chassis with the option of adding up to six additional chassis to that VC as network demands increase over time. VCs also provide enhanced resiliency and redundancy features. If a chassis in a VC goes down or is taken offline, the other chassis in the VC will continue to operate without disruption. In addition operating software and configuration parameters are synchronized on all chassis in the VC.

The following are key points regarding a virtual chassis configuration:

- The OmniSwitch 6860/6860E operates only in Virtual Chassis mode, a single chassis may still be referred to as a standalone chassis or a virtual chassis of one chassis.
- Virtual Chassis CLI requires a chassis identifier to be used and displayed for all commands such as interfaces or ports even if it is a standalone chassis and not integrated with other chassis into a virtual chassis.
- A Virtual Chassis provides a single management IP address for a group of chassis that are acting as a single bridge or router.
- The chassis participating in a Virtual Chassis are connected through dedicated VFL ports.

For more information on the components of a Virtual Chassis, see [“Virtual Chassis Overview” on page 13-6](#)

In This Chapter

This chapter describes the basic components of a Virtual Chassis and how to configure it through the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of the commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

The following information and configuration procedures are included in this chapter:

- [“Virtual Chassis Specifications” on page 13-3](#)
- [“Virtual Chassis Default Values” on page 13-4](#)
- [“Quick Steps for Automatically Configuring A Virtual Chassis” on page 13-5](#)
- [“Virtual Chassis Overview” on page 13-6](#)
- [“Managing A Virtual Chassis” on page 13-9](#)
- [“Virtual Chassis Split Protection \(VCSP\)” on page 13-12](#)
- [“Virtual Chassis - ISSU” on page 13-15](#)
- [“Interaction with Other Features” on page 13-17](#)
- [“Displaying Virtual Chassis Configuration and Status” on page 13-23](#)

Virtual Chassis Specifications

The table below lists specifications for dynamic aggregation groups and ports:

| | |
|--|------------------------|
| Platforms Supported | OmniSwitch 6860, 6860E |
| Maximum number of physical chassis in a Virtual Chassis. | 8 |
| Valid chassis identifier | 1 through 8 |
| Valid chassis group identifier | 0-255 |
| Valid chassis priority | 0-255 |
| Maximum number of Virtual Fabric Links per chassis | 2 |
| Valid Virtual Fabric Link identifier | 0 or 1 |
| VFL Supported Port Types | Dedicated VFL ports |
| Valid control VLAN | 2-4094 |
| EMP Address functionality | OmniSwitch 6860E only |
| Valid Virtual Chassis protocol hello interval | 1-65535 |
| Licenses Required | N/A |

Virtual Chassis Default Values

The table below lists default values for Virtual Chassis.

| Parameter Description | Command | Default Value/Comments |
|--------------------------|--|------------------------|
| Chassis Identifier | <code>virtual-chassis configured-chassis-id</code> | 0 |
| Chassis group identifier | <code>virtual-chassis chassis-group</code> | 0 |
| Chassis priority | <code>virtual-chassis configured-chassis-priority</code> | 100 |
| Hello-interval | <code>virtual-chassis hello-interval</code> | 15 seconds |
| Control VLAN | <code>virtual-chassis configured-control-vlan</code> | 4094 |

Quick Steps for Automatically Configuring A Virtual Chassis

Follow the steps below for a quick tutorial on configuring multiple chassis to automatically operate as a Virtual Chassis. Additional information on how to configure a Virtual Chassis is provided in the section “[Configuring Virtual Chassis](#)” on page 13-18.

Note: The vcsetup.cfg and vcboot.cfg files must NOT be present on the chassis. This is the normal configuration for newly shipped chassis.

Automatic Virtual Chassis Creation

- 1 Connect each chassis to the next in a ring configuration using the dedicated VFL ports.
- 2 Power on each chassis.
- 3 The VFL ports will be automatically detected and configured.
- 4 The chassis with the lowest MAC address will become the Master and be configured with chassis-id 1.
- 5 All other chassis will become Slave chassis and be assigned a unique chassis-id.

Viewing the Virtual Chassis Configuration

- 1 Use the [show virtual-chassis topology](#) command to check the topology of the Virtual Chassis.

```
-> show virtual-chassis topology
Local Chassis: 1
Oper
Chas  Role           Status           Config  Oper
Chas  ID              Pri             Group   MAC-Address
-----+-----+-----+-----+-----+-----+-----+-----
1     Master           Running          1       100  0       e8:e7:32:a4:8d:7f
2     Slave            Running          2       100  0       e8:e7:32:a4:89:2b
3     Slave            Running          3       100  0       e8:e7:32:a4:8c:cd
```

- 2 Use the [show virtual-chassis consistency](#) command to check the consistency of the virtual chassis.

```
-> show virtual-chassis consistency
Legend: * - denotes mandatory consistency which will affect chassis status
        licenses-info - A: Advanced; B: Data Center; D: DP-Inspect

Config  Oper  Oper  Config
Chas*  Chas  Chas  Chas  Hello  Oper  Config
Chas*  ID    Status  Type*  Group*  Interv  Vlan*  Vlan  License*
-----+-----+-----+-----+-----+-----+-----+-----+-----
1     1     OK      OS6860  0      15     4094  4094
2     2     OK      OS6860  0      15     4094  4094
3     3     OK      OS6860  0      15     4094  4094
```

- 3 Use the [show virtual-chassis vf-link](#) command to check the status of the of the virtual-link (VFL).

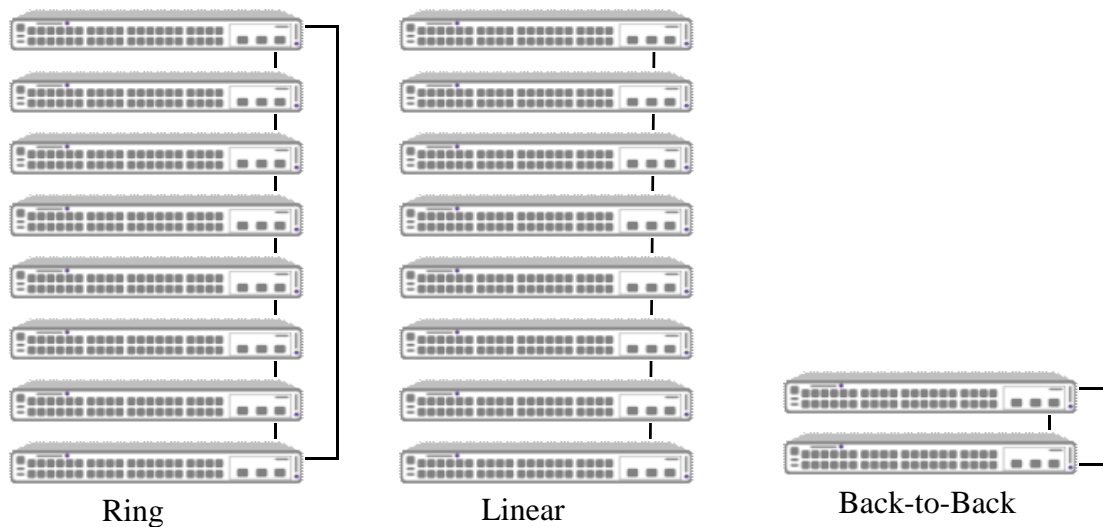
```
-> show virtual-chassis vf-link member-port
Chassis/VFLink ID  Chassis/Slot/Port  Oper  Is Primary
-----+-----+-----+-----+-----+-----
1/0                1/1/53             Up     Yes
1/1                1/1/54             Up     Yes
2/0                2/1/53             Up     Yes
2/1                2/1/54             Up     Yes
3/0                3/1/53             Up     Yes
3/1                3/1/54             Up     Yes
```

Virtual Chassis Overview

Virtual Chassis is a group of chassis managed through a single management IP address. It provides both node level and link level redundancy for both layer 2 and layer 3 protocols and services. This section describes the main topics regarding Virtual Chassis such as benefits, components, mode of operation, configuration conversion, start up and redundancy.

Some of the key benefits provided by a Virtual Chassis are:

- A single, simplified configuration to maintain
- Optimized bandwidth usage between the access layer and core
- Provides predictable and consistent convergence with redundant links to the two chassis
- Allows for exclusion of spanning-tree and other redundancy protocols like VRRP between the access layer and the core
- A Virtual Chassis appears as single router or bridge with support for all protocols
- A Virtual Chassis can be upgraded using ISSU to minimize network impact



Virtual Chassis Topologies

Virtual Chassis Concepts and Components

Virtual Chassis is an OmniSwitch feature that requires specific building blocks to provide full functionality. The following sections highlight the various components of a Virtual Chassis architecture.

Virtual Chassis - The entity consisting of multiple physical chassis connected using the virtual-fabric links.

Master Chassis - The Master chassis in a virtual chassis topology acting as the entry point for management and control operations. All configuration changes will be made on this chassis and communicated to the Slave chassis. The VC LED will be green.

Slave Chassis - Any chassis which is not the Master chassis is considered a Slave chassis. A Slave chassis is not directly configured, it communicates with the Master chassis through the virtual-fabric links to determine its configuration. The VC LED will be amber.

Virtual Chassis EMP Address - The Virtual Chassis management IP address (EMP-VC). This is a configurable IP address that is automatically assigned to the current primary chassis management module (CMM) of the master chassis. This parameter is stored in the *vcboot.cfg* configuration file.

Chassis EMP Address - The local chassis management IP address (EMP-CHAS1). This is a configurable IP address that is automatically assigned to the primary chassis management module (CMM) of the local chassis regardless of its master or slave role. This parameter is stored in the chassis specific *vcsetup.cfg* configuration file.

Virtual Fabric Link (VFL) - Dedicated ports that connect the chassis participating in the Virtual Chassis. As one of the basic building blocks of a Virtual Chassis configuration, the VFL facilitates the flow of traffic and the transfer of control data between the Master and Slave chassis.

Control VLAN - A special type of VLAN reserved for the inter-chassis communication exchange between the switches participating in a Virtual Chassis. Only VFL ports are assigned to this VLAN, and no other ports are allowed to join the Control VLAN.

IS-IS VC - Proprietary protocol for managing a Virtual Chassis topology. This protocol has no interaction with IS-IS routing or IS-IS SPB protocols. Responsible for information exchange with peers over the VFL, determining adjacencies, loop-detection and the shortest path between members of the VC.

VCSP - Virtual Chassis Split Protection. A proprietary protocol used by VC to detect and protect against network disruption when a VC splits.

vcsetup.cfg - A file containing information pertaining to the current physical switches, helping incorporate it into a virtual chassis. This file contains information such as Chassis ID, Group ID, Chassis priority, control VLAN, and chassis EMP IP addresses.

vcboot.cfg - A file containing information pertaining to the virtual chassis as a whole including L2 and L3 configuration, management configuration, user ports configuration, etc.

Virtual Chassis - Boot-Up

The Master chassis contains the *vcboot.cfg* file that contains the configuration for the entire virtual chassis. All the switches (i.e. the one that will eventually become the Master and the ones that will become Slaves) contain a *vcsetup.cfg* file that allows them to establish an connection over a VFL to all the other neighboring chassis.

- 1 Upon boot-up, a switch will read its local *vcsetup.cfg* file and attempt to connect to the other neighbor switches.
- 2 Upon connection, the switches will exchange the parameters configured in their local *vcsetup.cfg* files.
- 3 As a result of this exchange, they will discover the topology, elect a Master based on criteria described in the next section, start periodic health checks over the VFL and synchronize their configuration as defined within the *vcboot.cfg* configuration file.
- 4 All Slaves, if they do not have a local copy of *vcboot.cfg*, or their local copy does not match the copy found on the Master, will download their complete *vcboot.cfg* from the Master chassis and reboot using this copy of *vcboot.cfg* as its configuration file.

Startup Error Mode

If a switch is unable to successfully come up in virtual chassis mode, it enters a special fallback mode called start up error mode. A switch start up error mode will keep all of its front-panel user ports, including the virtual-fabric links member ports disabled. This mode can be identified on the switch by using the [show virtual-chassis topology](#) command. The chassis role will display **Inconsistent**, whereas the chassis status will show either one of the following values:

- **Invalid-Chassis-Id:** The chassis is not operational in virtual chassis mode because no valid chassis identifier has been found in the configuration. Typically this means that the *vcsetup.cfg* file is corrupted, empty or contains an invalid (for example out of range) chassis identifier.
- **Invalid-License:** The chassis is not operational in virtual chassis mode because a needed license has not been found.

Master/Slave Election

Once all switches complete their initialization their VFLs become operational, they start the virtual chassis protocol. This protocol performs three basic functions including: topology discovery, master election and keep-alive/hello monitoring. The election of the Master chassis is based on the following criteria, which are listed from the higher to the lower priority.

1. Current Master Chassis - The current master chassis will remain the master chassis if it is not rebooted.
2. Highest chassis priority value
3. Longest chassis uptime
4. Smallest Chassis ID value
5. Smallest chassis MAC address

Virtual Chassis - Redundancy

- If the Master chassis goes down a Slave chassis will takeover the Master role.
- If a Slave chassis goes down the Master chassis will retain its Master role.
- If a VFL goes down in a ring topology, the Master chassis will retain its Master role and the VC will continue to function normally.
- The a VFL goes down in a linear topology, a Slave chassis will transition to assume the Master role. At this point the virtual chassis topology has been split and there will be two Masters in the network.

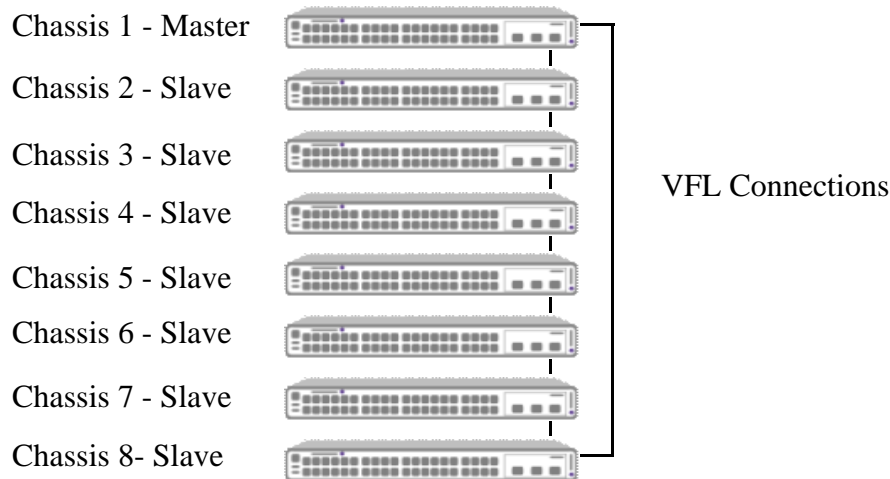
Virtual Chassis Split Protection (VCSP)

Virtual chassis split protection is implemented using the proprietary VCSP protocol. The goal of the VCSP mechanism is to provide information in a virtual chassis environment which can be used to determine whether a VFL has failed and resulted in a split VC. A split VC can occur when one or multiple VFL connections are broken but each of the switches remains operational. This scenario must be detected so that only one of the switches remains the Master and continues using the same IP and MAC address in the network.

See the [“Virtual Chassis Split Protection \(VCSP\)” on page 13-12](#) for information on configuring VCSP.

Managing A Virtual Chassis

The OmniSwitch 6860 has dedicated ports that connect the chassis participating in a Virtual Chassis. As one of the basic building blocks of a Virtual Chassis configuration, the VFL facilitates the flow of traffic and the transfer of control data between the Master and Slave chassis.



Basic Virtual Chassis Configuration

Chassis ID Assignment

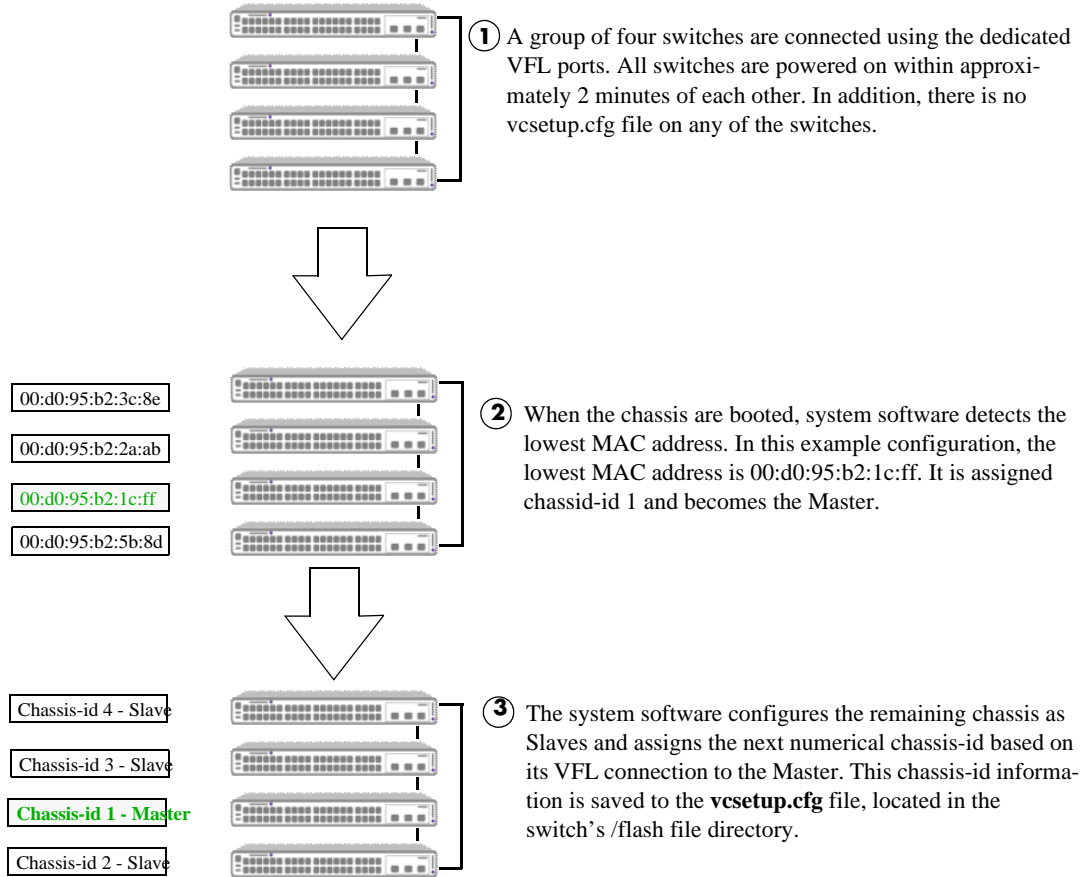
A new virtual chassis can easily be created by connecting multiple switches together using the dedicated VFL ports and then powering on the switches. Chassis-ID assignment occurs when there is no `vcsetup.cfg` file on a switch. This is how switches are shipped from the factory making new, “out of the box” switches, simple to incorporate into a virtual chassis. When no `vcsetup.cfg` exists, one will be automatically created and a chassis-id will be written to it as part of automatic chassis-id assignment.

For a newly created virtual chassis the Master role will be given to the switch with the lowest chassis MAC address. This is due to all other election criteria being equal, see “[Master/Slave Election](#)” on [page 13-8](#). However, for this to occur, all chassis in the virtual chassis *should be booted within approximately 2 minutes of each other*. In addition, the chassis *must have no preconfigured virtual chassis information*. Because of these two conditions, the MAC address method for selecting the Master will be used with new “out of the box” switches, or any switch that does not contain a `vcsetup.cfg` file.

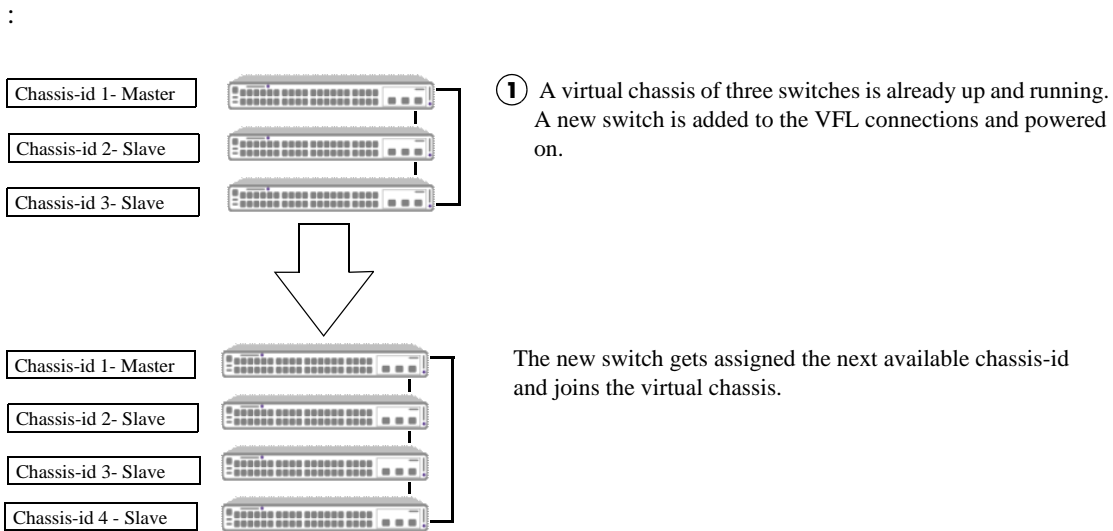
Slave Chassis-id Assignment

After the Master has been established, the remaining switches in the virtual chassis will become Slaves. The chassis-id of each slave chassis will be determined by each chassis' physical location in the VC. The chassis-id will be incremented with the direction from the Master chassis toward the chassis connected to lowest numbered VFL port (VFL 0) of the master chassis.

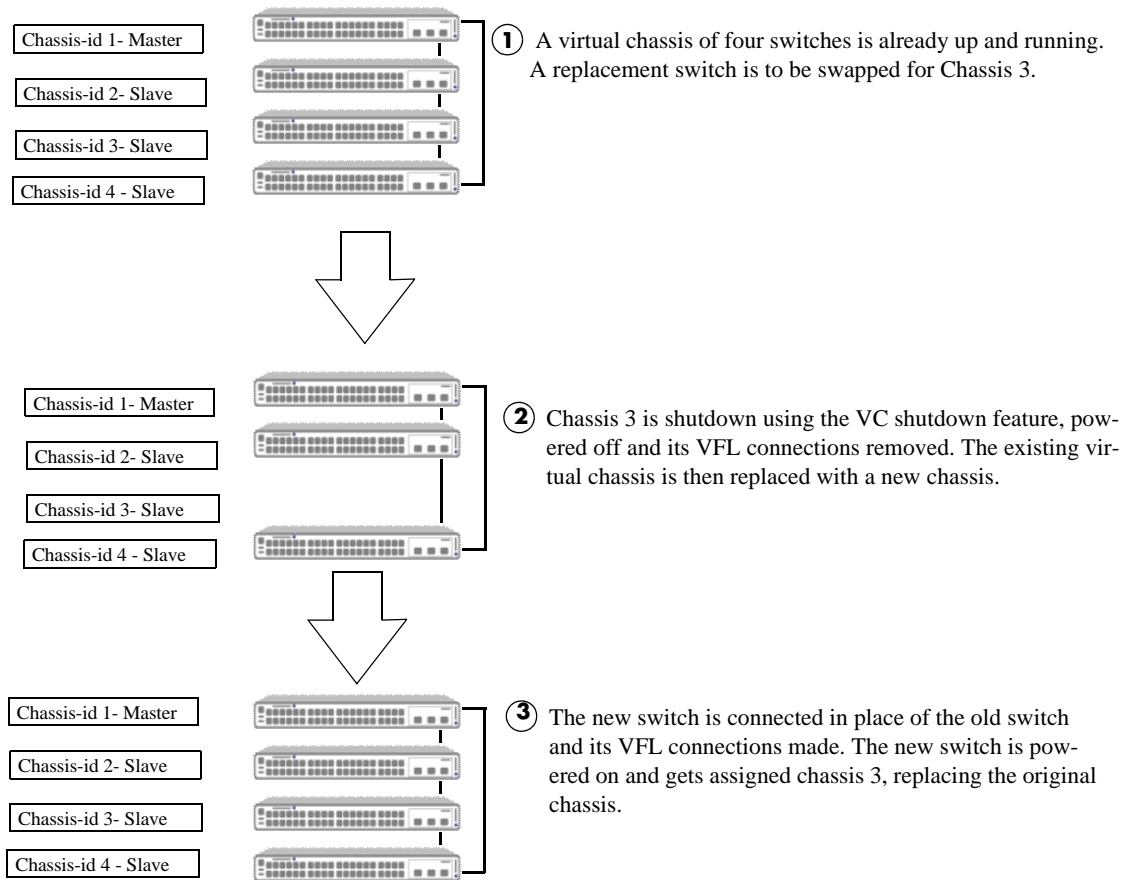
For more information on using the lowest MAC address to determine the Master switch in a VC, refer to the diagram below:



Automatic Master/Slave and Chassis-id Assignment



Adding A New Switch to Existing Virtual Chassis



Replacing A Switch

Virtual Chassis Split Protection (VCSP)

In the case of a VC split due to a VFL link failure or the failure of one of the VC elements, both of the resulting VCs could end up having the same system MAC and IP addresses. Since there is no communication between these individual VCs due to the VFL link failure they end up communicating with the rest of the network devices using the same MAC and IP addresses. This split scenario is disruptive to the network as the conflicting MAC and IP addresses can lead to layer 2 loops and L3 traffic disruption.

VCSP provides the following benefits:

- Avoid network disruptions by preventing duplicate MAC and IP addresses on the network when MAC retention is enabled and a VC split occurs.
- The sub-VC that forms out of the split is able to detect that a split has occurred.
- Once the VC split condition has been determined, the sub-VC will put its front-panel ports into an operationally down state preventing traffic forwarding and avoiding loops and possible traffic disruption. The VCSP link aggregate ports will remain up.
- A trap can be sent by the active-VC indicating the split state. The trap indicates that the split has occurred and which elements are in the operationally down sub-VC.
- A mechanism is available to recover the non-operational sub-VC.
- A method of detecting a VC split in a remote VC topology where the VC may consist of elements located in different physical locations such as a remote site, or multiple floors of a building.

VCSP Key Components and Terms

- **VCSP PDU** - A proprietary packet forwarded between VC elements to help determine that state of the VC.
- **VCSP Helper** - A neighboring OmniSwitch, not an element of the VC, responsible for forwarding VCSP PDUs between the VC elements. The VCSP feature and the VCSP Helper functionality cannot be enabled on the same switch. The VCSP helper and the VC cannot have the same Group ID.
- **VCSP Link Aggregate** - A dedicated link aggregate configured between all elements of a VC and a helper switch to be used for forwarding VCSP PDUs.
- **Active-VC** - An element, or multiple elements, that results when a VC split occurs. The active-VC will keep its front panel ports enabled and continue to forward traffic on the network.
- **Sub-VC** - An element, or multiple elements, that results when a VC split occurs. The sub-VC will disable its front panel ports to prevent traffic disruption caused by duplicate MAC/IP addresses with the active VC.
- **Protection State** - A state an element will transition to after determining a VC split has occurred. Its ports will be operationally disabled to prevent duplicate MAC and IP addresses and network disruption.
- **Guard Timer** - A configurable timer determining how long a unit will wait before beginning to send VCSP PDUs after a VC recovery.

Basic Operation

When VCSP is enabled, a proprietary protocol runs on a configured link aggregate to carry the VC information necessary for VC split detection. Each of the VC elements share a link aggregate with an OmniSwitch that can act as a helper to assist in the VC split detection.

The lowest member port of the link aggregate hosted on the master element is responsible for sending the VCSP PDUs on the member link. When the packet arrives on the remote helper device then the helper device will forward the packet out on all member ports of the link aggregate so that the packet reaches the remote VC elements.

Use the the [virtual-chassis split-protection admin-state](#) and [virtual-chassis split-protection linkaggid](#) commands to enable VCSP and create the VCSP link aggregate on the VC.

Use the the [virtual-chassis split-protection helper admin-state](#) and [virtual-chassis split-protection helper linkagg](#) commands to enable the VCSP helper and create the VCSP helper link aggregate on the helper switch.

Protection States

Under normal VC circumstances the VCSP PDUs are sent once every 3 seconds. When the VC detects a scenario which leads to a change in its size the protocol sends the VCSP PDUs at a rate of 1 per 50 milliseconds for 3 to 10 seconds. This helps to quickly identify a VC split.

On reception of a VCSP frame, the receiving elements match the master of their current VC against the MAC address of the VCSP sender. If there is a mismatch between the two, then that indicates the presence of an active VC in the network which is disconnected from the current VC. In this condition the VC element will monitor 3-5 such consecutive frames, after which it will transition to the **protection state**.

When an element transitions to the protection state, the following occurs:

- Each of the VC elements will independently transition into the protection state.
- In the protection state each of the VC elements will disable all the user ports except the ones belonging to the VCSP protection link aggregate.
- The VC element will store the protection state transition information in a non-volatile location. This information will be used whenever an element in the protection state re-boots and needs to check its current operational state.
- An element in the protection state will send a VCSP PDU back on the link aggregate carrying the VCSP state as PROTECTION. This is the only frame that is sent by the protection sub-VC. The purpose of this message is to inform the active-VC to generate an SNMP trap regarding the VC split state of the VC element.

VC Split Recovery

Once a sub-VC goes into the protection state then all the front panel ports are put into an operationally down state. There are two ways to recover the VC, manually or automatically.

Manual Recovery

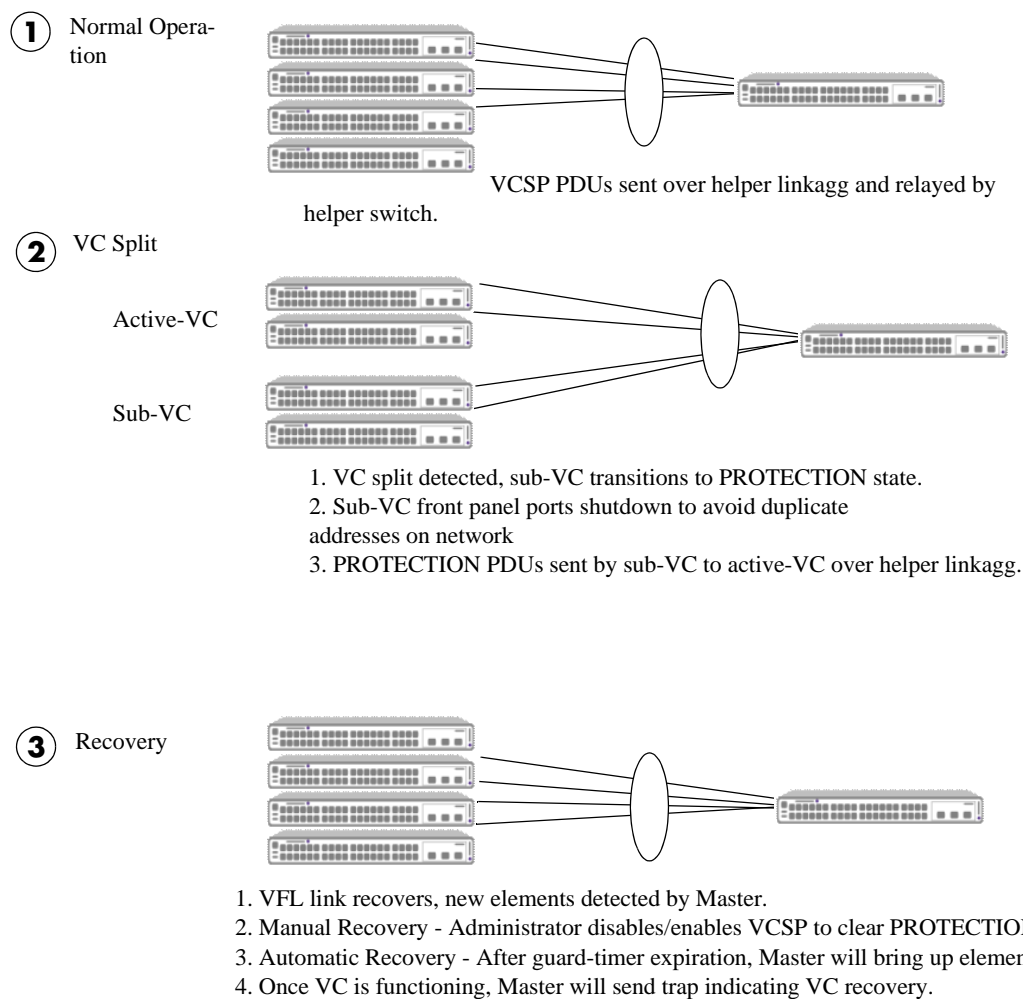
If the guard timer is 0, automatic recovery is disabled and the sub-VC stays in protection mode until the unit is reloaded by the administrator. After the re-boot the administrator has to manually recover the switch by first disabling VCSP and then re-enabling VCSP. This clears the protection state variables stored on the switch.

Use the **virtual-chassis split-protection guard-timer** and the **virtual-chassis split-protection admin-state** commands to configure the guard timer and enable/disable VCSP.

Automatic Recovery

If the VFL link recovers and the sub-VC reconnects to the active VC the sub-VC will automatically reboot. The protection units will come up in the protection state; however they will now be part of the active VC but their front panel ports will still be disabled. The master of the combined VC will detect that there are new elements in the protection state. The master will wait for 60 seconds then bring one element at a time from the protection state until all elements are active.

Once complete, the master will generate a trap indicating the VC has been recovered.



VC Split Example

Virtual Chassis - ISSU

Upgrading a virtual chassis can be performed by uploading the new images to the Master chassis and rebooting the entire virtual chassis. Prior to rebooting the Master will copy the new image files to the Slave and once the virtual chassis is back up the entire virtual chassis will be synchronized and running with the upgraded code.

The standard procedure is as follows:

- 1 Connect to the Virtual Chassis using the VC-EMP IP address.
- 2 Upload the new images to the Master chassis.
- 3 Reboot the entire virtual chassis
- 4 The new images are copied to the Slave chassis prior to rebooting and the entire virtual chassis is synchronized when it comes back up.

Virtual Chassis - Upgrading with ISSU

The In-Service Software Upgrade (ISSU) feature is used to upgrade a Virtual Chassis with minimal network disruption. Each switch can be upgraded individually without requiring the entire Virtual Chassis to be rebooted at the same time.

The ISSU upgrade overview:

- 1 The Virtual Chassis will first verify that it is in a state that will allow a successful ISSU upgrade.
- 2 It will then copy the images and configuration files of the ISSU specified directory to the Slave chassis and reload the entire Slave chassis from the ISSU directory. The Slave chassis reloads with the new image files.
- 3 When the Slave has rebooted and rejoined the Virtual Chassis it is now using the new image files. The Master chassis is now reset which causes the Slave chassis to become the new Master chassis.
- 4 When the original Master chassis reloads it comes back as a Slave chassis. To restore the role of Master to the original Master chassis the current Master can be rebooted and the original Master will take-over, assuming the Master role.

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 Virtual Chassis is fully synchronized and certified.

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

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

4 The Master copies the **/flash/issu_dir** directory to the Slave chassis, the Slave is reloaded with the upgraded code and rejoins the Virtual Chassis.

5 The Master chassis is now reloaded with the upgraded code which causes the Slave chassis to become the new Master chassis.

As a result of this process all chassis are now running with the upgraded code and the Master and Slave chassis will have changed roles.

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

7 Enter '**reload chassis-id 2**' to have the original Master resume its Master role.

Interaction with Other Features

This section contains important information about how other OmniSwitch features interact with the virtual chassis feature. Refer to the specific chapter for each feature to get more detailed information about how to configure and use the feature.

VCSP

If the VCSP helper is a virtual chassis, it cannot have the same Group ID as the virtual chassis it is being the helper for.

Configuring Virtual Chassis

This section describes commands to configure virtual chassis on an OmniSwitch.

- [“Virtual Chassis Configuration Guidelines” on page 13-18](#)
- [“Configuring the Chassis Identifier” on page 13-20](#)
- [“Configuring the Virtual Chassis Group Identifier” on page 13-20](#)
- [“Configuring the Hello Interval” on page 13-20](#)
- [“Configuring the Hello Interval” on page 13-20](#)
- [“Configuring the Control VLAN” on page 13-21](#)
- [“Configuring Virtual Chassis EMP IP Addresses” on page 13-21](#)

Note. See [“Quick Steps for Automatically Configuring A Virtual Chassis” on page 13-5](#)” for a brief tutorial on configuring these parameters on an OmniSwitch.

Virtual Chassis Configuration Guidelines

The following sections provide configuration guidelines to follow when configuring a virtual chassis on an OmniSwitch. The configuration commands related to the virtual chassis functionality are used for manually configuring the virtual chassis options. Many of these configuration options can be automatically configured by the AOS software by incorporating switches into the virtual chassis without a `vcsetup.cfg` file.

General

- One of the chassis will become the Master chassis while the remaining switches will become Slaves.
- Some of the virtual chassis parameters runtime modification only take effect after the next reboot of the switch. These parameters are chassis identifier, chassis priority, control VLAN and hello interval. For this type of parameters, the following terminology is used.
 - Operational values - The current or running values, are those in effect at the present time.
 - Configured values - The next or future values are those that are currently configured or set, but that are not in effect at the present time. These values will only become effective after the next reboot of the switch.

Chassis Identifier

- Each chassis requires a chassis identifier that is unique within the virtual chassis group of topology. A chassis identifier can be automatically or manually configured.
- If a duplicate chassis identifier is detected within the virtual chassis group then the chassis role will be reported as *inconsistent* and the chassis status will be *Duplicate-Chassis*. The front-panel ports will not be operational and the configuration must be corrected by accessing the switch directly through the local EMP port or a console connection.
- The chassis identifier is used to generate globally unique values for the module identifiers as well as allowing inter-chassis communication.

- A chassis reboot is required for a newly configured chassis identifier to take effect.

For information about configuring the Chassis ID, see [“Configuring the Chassis Identifier” on page 13-20](#)

Virtual Chassis Group Identifier

- Each chassis also requires a virtual chassis group identifier to identify the chassis belonging to that specific virtual chassis topology.
- The same group identifier must be assigned to each chassis in the virtual chassis topology. Switches belonging to other virtual chassis groups should use a different group identifier.
- If two or more chassis within the same virtual chassis group do not have the same group identifier configured, the chassis role will be reported as *Inconsistent* and the chassis status will be *Mismatch-Chassis-Group*. The front-panel user ports will not be brought to an operational state. The configuration must be corrected by accessing the chassis directly.
- If two or more separate virtual chassis groups use the same group identifier, this inconsistency is not detected or corrected by the virtual chassis functionality. It is up to the administrator to ensure that each domain uses a unique group identifier.
- The virtual chassis group identifier is used to select a globally unique virtual MAC address for each virtual chassis group to avoid duplicate MAC addresses in a network that may contain more than one virtual chassis group.

For information about configuring the chassis group identifier, see [“Configuring the Virtual Chassis Group Identifier” on page 13-20](#).

Virtual Fabric Link (VFL)

- An operational VFL is a basic requirement to support a fully functional virtual chassis. OmniSwitch 6860/6860E have dedicated VFL ports.
- Explicitly configuring the VFL and the physical port members is not supported, this function is automatically configured and is not user-configurable. The lowest numbered VFL port is VFL 0 and the highest numbered VFL port is VFL 1.
- It is recommended to connect the chassis using the VFL ports prior to booting the switch.
- Only the dedicated VFL ports can be members of a VFL.
- With a Virtual Chassis consisting of two chassis, the VFL ports are bundled to form a single link aggregate VFL 0.
- For increased resiliency and to help avoid a split chassis scenario it is strongly recommend to have the VFL ports connected in a ring topology.
- The VFL automatically becomes a member of all VLANs configured on the chassis.

For more information on Virtual Fabric Link, see [“Configuring the Hello Interval” on page 13-20](#)

Control VLAN

- The control VLAN is a reserved VLAN used for transporting control packets among the chassis comprising the virtual chassis.
- Runtime configuration changes to the control VLAN will only take effect after the next reboot of the chassis.

- The control VLAN must be the same between the chassis comprising the virtual chassis.

For more information on the Control VLAN, see [“Configuring the Control VLAN” on page 13-21](#)

Configuring the Chassis Identifier

To configure the Virtual Chassis feature, a unique chassis identifier must first be automatically or manually assigned to each of the switches that will form the Virtual Chassis group. Assigning the chassis identifier also enables the configuration of the additional virtual chassis parameters for the chassis.

The [virtual-chassis configured-chassis-id](#) command is used to configure a unique chassis identifier for a switch within the virtual chassis group. For example:

```
-> virtual-chassis chassis-id 1 configured-chassis-id 2
```

Note: Changing the chassis-id will not cause the Control VLAN or Group ID to change. These parameters must also be manually changed if required.

Duplicate chassis identifier

In the event two chassis have the same operational chassis identifier value, one of them will be reported as **Inconsistent** role (instead of Master or Slave) and **Duplicate-Chassis** status. This will cause the operational chassis identifier of one of the chassis to be automatically renumbered to follow the duplicate chassis ID format. (i.e. 101). This numbering is reserved to represent a chassis whose chassis identifier is duplicate. All management interface commands must use this new operational chassis identifier to affect any configuration. The duplicate chassis identifier must be corrected by re-configuring the switch locally through the console or EMP port access. For example:

```
-> virtual-chassis chassis-id 101 configured-chassis-id 2
```

Configuring the Virtual Chassis Group Identifier

A virtual chassis group identifier must be assigned to each of the switches that will form the Virtual Chassis group. Each of these switches must use the same group identifier, which identifies the switch as belonging to that virtual chassis group.

The [virtual-chassis chassis-group](#) command is used to configure the same group identifier for each chassis within the virtual chassis group. For example:

```
-> virtual-chassis chassis-group 1
```

By default, the virtual chassis group identifier is set to “0”. In a network environment where more than one virtual chassis group may exist, configure each virtual chassis group with its own unique group identifier. Duplicate group identifiers are not supported.

Configuring the Hello Interval

Hello packets are used for establishing and maintaining the neighbor relationship between virtual chassis switches and ensures that communication between switches is bidirectional. Hello packets are sent periodically out VFL interfaces. Bidirectional communication is indicated when the switch sees itself listed in the neighbor's hello packet. The hello interval value determines how often these packets are sent.

To configure the hello interval between the multi-chassis peers, use the [virtual-chassis hello-interval](#) command as shown below:

```
-> virtual-chassis hello-interval 10
```

Configuring the Control VLAN

Under normal circumstances, it is not necessary to change the control VLAN.

However, it is important to note that the VLAN configured as the Control VLAN is reserved specifically for transferring virtual chassis control information purposes and it can no longer be used for normal data traffic.

If necessary, use the [virtual-chassis configured-control-vlan](#) command to modify the Control VLAN. For example:

```
-> virtual-chassis configured-control-vlan 4093
```

Configuring Virtual Chassis EMP IP Addresses

In order to access the virtual chassis through the EMP IP addresses the port's IP address and network mask can be configured. There are multiple IP addresses to consider when configuring the EMP IP addresses in a virtual chassis environment.

- The Virtual Chassis EMP IP address represents the address of the entire virtual chassis (EMP-VC). This address is automatically assigned to the Master chassis and can be used for remote access to the entire Virtual Chassis.
- The Chassis EMP IP address is assigned to each switch comprising the virtual chassis (i.e. EMP-CHAS1, EMP-CHAS2, EMP-CHAS3, etc.). This address can be used for remote access to each switch comprising the virtual chassis. This address is automatically assigned to the local chassis.
- All the EMP IP addresses and CMM's IP addresses must be in the same subnet.
- Each of the IP addresses must be unique.

Note: The Virtual Chassis EMP addresses are only supported on OS680E models (non-E models do not have an EMP port). When configuring VC EMP addresses in a VC with a mix of OS6860E models and OS6860-non-E models the VC could become unreachable via the EMP port if a non-E model becomes Master. In a mixed VC environment it's recommended to create a management VLAN for remote management purposes.

Configuring the Chassis EMP IP Address

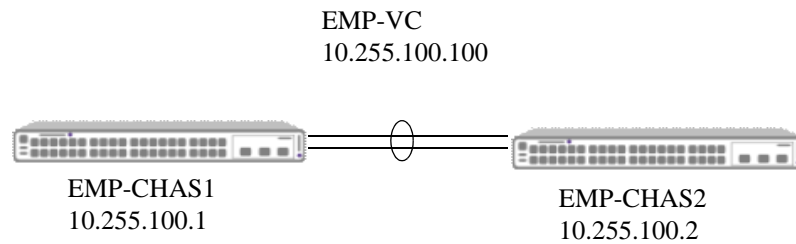
Use the **ip interface** command to modify the Chassis EMP IP address as shown below. These commands would be issued after the virtual chassis is operational:

```
-> ip interface local chassis-id 1 emp address 10.255.100.1 mask 255.255.255.0
-> ip interface local chassis-id 2 emp address 10.255.100.2 mask 255.255.255.0
```

Configuring the Virtual Chassis EMP IP Address

Use the **ip interface** command to modify the Virtual Chassis EMP IP address as shown below. These commands would be issued after the virtual chassis is operational:

```
-> ip interface master emp address 10.255.100.100 mask 255.255.255.0
```



Additional EMP IP Addresses

An additional CMM IP address, stored in NVRAM can also be configured. These addresses can be used to access a specific CMM but are not required for remote access. On an OmniSwitch the IP addresses are named as follows and are associated to each CMM on each chassis.

- EMP-CMMA-CHAS1

A direct connection to the associated CMM's console port is required before attempting to change IP address information using the **modify boot parameters** command as shown in the example below:

```
-> modify boot parameters
Boot > boot empipaddr 255.255.100.50
Boot > boot empmasklength 16
Boot > commit system
Boot > commit
```

Controlled Shutdown of a Virtual Chassis Participant Switch

The **virtual-chassis shutdown** command allows a switch to be brought to an isolated state where all user ports and virtual-fabric link member ports are brought down. This allows for the graceful removal of the switch from the active virtual chassis topology.

Note: This command should not be used on the Master chassis.

Displaying Virtual Chassis Configuration and Status

You can use Command Line Interface (CLI) **show** commands to display the current configuration and status of a virtual chassis group. These commands include the following:

- show virtual-chassis topology** Displays details about the configured and operational parameters related to all switches participating in the virtual chassis topology
- show virtual-chassis vf-link** Displays the configured and operational parameters related to the virtual-fabric link and member ports.
- show virtual-chassis consistency** Displays detailed status of the parameters that are taken into account to determine the consistency of a group of switches participating in the virtual chassis topology.

For more information about the output details that result from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

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 party 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. |
| <p>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.</p> <p>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).</p> <p>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.</p> | | | | |
| 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. |
| <p>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.</p> <p>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).</p> <p>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.</p> | | | | |
| 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 | chassis- TrapsStrLevel chassis- TrapsStrAppID chassis- TrapsStrSnapID chassisTrapsStr- fileName chassisTrapsStr- fileLineNb chassis- TrapsStrErrorNb chassisTrapsStr- comments chassisTrapsStr- dataInfo | 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 Sub-application (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 identifier. 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 chassisTraps- ObjectType chassisTraps- ObjectNum- ber chassisTrap- sAlertNum- ber chassisTrap- sAlertDescr | chassis | A notification that some change has occurred in the chassis. |
| <p>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.</p> | | | | |
| 9 | chassisTrapsStateChange | physicalIndex chassisTraps- ObjectType chassisTraps- ObjectNum- ber chasEntPhys- OperStatus | chassis | An NI status change was detected. |
| <p>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).</p> | | | | |
| 10 | chassisTrapsMacOverlap | physicalIndex chasTrapMac- RangeIndex | module | A MAC range overlap was found in the backplane eeprom. |
| <p>physicalIndex—The physical index of the involved object. chasTrapMacRangeIndex—The MAC range index of the involved object.</p> | | | | |
| 11 | vrrpTrapNewMaster | vrrpOperMas- terIpAddr | vrrp | The SNMP agent has transferred from the backup state to the master state. |
| <p>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.</p> | | | | |
| 12 | vrrpTrapAuthFailure | vrrpTrapPack- etSrc vrrpTrapAu- thErrorType | vrrp | A packet was received from the network whose authentication key conflicts with the switch’s authentication key or type. |
| <p>vrrpTrapPacketSrc—The IP address of an inbound VRRP packet. vrrpTrapAuthErrorType—Potential types of configuration conflicts.</p> | | | | |

| No. | Trap Name | Objects | Family | Description |
|--|-----------------------|---|--------|--|
| 13 | healthMonModuleTrap | healthModuleSlot healthMonRx-Status healthMonRx-TxStatus healthMon-MemoryStatus healthMon-CpuStatus | health | Indicates a module-level threshold was crossed. |
| <p>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.</p> | | | | |
| 14 | healthMonPortTrap | healthPortSlot healthPortIF healthMonRx-Status healthMonRx-TxStatus | health | Indicates a port-level threshold was crossed. |
| <p>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.</p> | | | | |
| 15 | healthMonCmmTrap | healthMon-MemoryStatus, healthMon-CpuStatus | health | This trap is sent when an NI memory or CPU threshold is crossed. |
| <p>healthMonMemoryStatus—Memory threshold status indicating if threshold was crossed or no change. healthMonCpuStatus—CPU threshold status indicating if threshold was crossed or no change.</p> | | | | |
| 16 | bgpEstablished | bgpPeerLastError bgpPeerState | bgp | The BGP routing protocol has entered the established state. |
| <p>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.</p> | | | | |
| 17 | bgpBackwardTransition | bgpPeerLastError bgpPeerState | bgp | This trap is generated when the BGP router port has moved from a more active to a less active state. |
| <p>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.</p> | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---------------------|--|-----------|--|
| 18 | esmDrvTrapDropsLink | esmPortSlot esmPortIF ifInErrors ifOutErrors esmDrvTrapDrops | interface | This trap is sent when the Ethernet code drops the link because of excessive errors. |
| <p>esmPortSlot—The physical slot number for this Ethernet Port. The slot number has been added to be used by the private trap.</p> <p>esmPortIF—The on-board interface number for this Ethernet port. The port number has been added to be used by the private trap.</p> <p>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.</p> <p>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 ifCounterDiscontinuityTime.</p> <p>esmDrvTrapDrops— Partitioned port (separated due to errors).</p> | | | | |
| 19 | portViolationTrap | ifIndex, portViolationSource, portViolationReason | 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 |
| <p>ifIndex—A unique value, greater than zero, for the interface.</p> <p>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".</p> <p>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. pvQosIisis, 14. pvQosDnsReply, 15. pvUdld.</p> | | | | |
| 20 | dvmrpNeighborLoss | dvmrpInterfaceLocalAddress dvmrpNeighborState | 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. |
| <p>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.</p> <p>dvmrpNeighborState—State of the neighbor adjacency.</p> | | | | |

| No. | Trap Name | Objects | Family | Description |
|-----|-------------------------|--|--------|---|
| 21 | dvmrpNeighborNotPruning | dvmrpInterface-LocalAddress dvmrpNeighborCapabilities | 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 alarmSampleType alarmValue alarmRisingThreshold | 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 alarmFalling- Threshold | 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. |
| <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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 risingOrFallingAlarm (3).</p> | | | | |
| 24 | stpNewRoot | vStpNumber | stp | Sent by a bridge that became the new root of the spanning tree. |
| <p>vStpNumber—The Spanning Tree number identifying this instance.</p> | | | | |
| 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. |
| <p>vStpNumber—The Spanning Tree number identifying this instance.</p> <p>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.</p> | | | | |
| 26 | mirrorConfigError | mirmonPrimarySlot mirmonPrimaryPort 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. |
| <p>mirmonPrimarySlot—Slot of mirrored or monitored interface.</p> <p>mirmonPrimaryPort—Port of mirrored or monitored interface.</p> <p>mirroringSlot—Slot of mirroring interface.</p> <p>mirroringPort—Port of mirroring interface.</p> <p>mirMonErrorNi—The NI slot number.</p> <p>mirMonError—The Error returned by the NI which failed to configure Mirroring/Monitoring.</p> | | | | |

| 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. |
| <p>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.</p> | | | | |
| 28 | slbTrapOperStatus | slbTrapInfoEntityGroup slbTrapInfoOperStatus slbTrapInfoClusterName slbTrapInfoServerIpAddr | load balancing | A change occurred in the operational status of the server load balancing entity. |
| <p>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.</p> | | | | |
| 29 | sessionAuthenticationTrap | sessionAccessType sessionUserName sessionUserIpAddress sessionAuthFailure | session | An authentication failure trap is sent each time a user authentication is refused. |
| <p>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.</p> | | | | |
| 30 | trapAbsorptionTrap | trapAbsorStamp trapAbsorTrapId trapAbsorCounter trapAbsorTime | none | The absorption trap is sent when a trap has been absorbed at least once. |
| <p>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.</p> | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|------------------------|--|-----------------|---|
| 31 | alaDoSTrap | alaDoSType alaDoSDetected | ip | Indicates that the sending agent has received a Denial of Service (DoS) attack. |
| <p>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</p> <p>pethMainPseConsumptionPower—Measured usage power expressed in Watts.</p> | | | | |
| 32 | ospfNbrStateChange | ospfRouterId ospfNbrIpAddr ospfNbrAddressLessIndex ospfNbrRtrId ospfNbrState | ospf | Indicates a state change of the neighbor relationship. |
| <p>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.</p> | | | | |
| 33 | ospfVirtNbrStateChange | ospfRouterId ospfVirtNbrArea ospfVirtNbrRtrId ospfVirtNbrState | ospf | Indicates a state change of the virtual neighbor relationship. |
| <p>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.</p> | | | | |
| 34 | lnkaggAggUp | traplnkaggId traplnkaggPortIfIndex | linkaggregation | Indicates the link aggregate is active. This trap is sent when any one port of the link aggregate group goes into the attached state. |
| <p>traplnkaggId—Index value of the Link Aggregate group. traplnkaggIfIndex—Port of the Link Aggregate group.</p> | | | | |
| 35 | lnkaggAggDown | traplnkaggId traplnkaggPortIfIndex | linkaggregation | 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. |
| <p>traplnkaggId—Index value of the Link Aggregate group. traplnkaggIfIndex—Port of the Link Aggregate group.</p> | | | | |

| No. | Trap Name | Objects | Family | Description |
|-----|------------------------|---|-----------------|---|
| 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 | alaVrrp3OperMasterIpAddrType alaVrrp3OperMasterIpAddr alaVrrp3TrapNewMasterReason | vrrp | The newMaster trap indicates that the sending agent has transitioned to Master state. alaVrrp3OperMasterIpAddrType —This specifies the type of alaVrrp3OperMasterIpAddr in this row. alaVrrp3OperMasterIpAddr —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 baseMacAd- dress | chassis | This trap is sent when there is a possibility of duplicate a MAC address in the network. |
| <p>physicalIndex—The Physical index of the involved object. baseMacAddress—The base MAC Address.</p> | | | | |
| 43 | lldpRemTablesChange | lldptatsRem- TablesInserts lldptatsRem- TablesDeletes lldptatsRem- TablesDrops lldptatsRem- TablesAge- outs | aip | This trap is sent when the value of the LLDP Stats Rem Table Last ChangeTime changes. It can be utilized by an NMS to trigger LLDP remote systems table maintenance polls. |
| <p>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.</p> | | | | |
| 44 | pimNeighborLoss | pimNeigh- borUpTime | ipmr | <p>This trap is sent when an adjacency with a neighbor is lost.</p> <p>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.</p> <p>The notification is generated whenever the PIM NeighborLoss Count is incremented, subject to the rate limit specified by the PIM Neighbor Loss Notification-Period.</p> |
| <p>pimNeighborUpTime—The time since this PIM neighbor (last) became a neighbor of the local router.</p> | | | | |
| 45 | pimInvalidRegister | PimGroupMap- pingPim- Mode pimInvalidReg- isterAddress- Type pimInvalidReg- isterOrigin pimInvalidReg- isterGroup pimInvalidReg- isterRp | ipmr | <p>This trap is sent when an invalid PIM Register message is received.</p> <p>The notification is generated whenever the PIM Invalid Register Message Received counter is incremented, subject to the rate limit specified by the Invalid Register NotificationPeriod.</p> |

| No. | Trap Name | Objects | Family | Description |
|-----|---------------------|---|--------|---|
| | | | | <p>pimGroupMappingPimMode—The PIM mode used for groups in this group prefix.</p> <p>pimInvalidRegisterAddressType—The address type stored in pimInvalidRegisterOrigin, pimInvalid RegisterGroup and pimInvalidRegisterRp. If no unexpected Register messages are received, the object is set to “Unknown”.</p> <p>pimInvalidRegisterOrigin—The source address of the last unexpected Register message received by this device</p> <p>pimInvalidRegisterGroup—The IP multicast group address to which the last unexpected Register message received by this device was addressed.</p> <p>pimInvalidRegisterRp—The RP address to which the last unexpected Register message received by this device was delivered.</p> |
| 46 | pimInvalidJoinPrune | pimGroupMappingPim-Mode pimInvalidJoin-PruneAd- dressType pimInvalidJoin-PruneOrigin pimInvalidJoin-PruneGroup pimInvalidJoin-PruneRp pimNeighborUpTime | ipmr | <p>This trap is sent when an invalid PIM Join/Prune message is received.</p> <p>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.</p> <p>pimGroupMappingPimMode—The PIM mode used for groups in this group prefix.</p> <p>pimInvalidRegisterAddressType—The address type stored in pimInvalidRegisterOrigin, pimInvalid RegisterGroup and pimInvalidRegisterRp. If no unexpected Register messages are received, the object is set to “Unknown”.</p> <p>pimInvalidJoinPruneOrigin—The source address of the last unexpected Join/Prune message received</p> <p>pimInvalidJoinPruneGroup—The IP multicast group address carried in the last unexpected Join/Prune message received</p> <p>pimInvalidJoinPruneRp—The RP address carried in the last unexpected Join/Prune message received</p> <p>pimNeighborUpTime—The time since this PIM neighbor (last) became a neighbor of the local router.</p> |
| 47 | PimRPMappingChange | pimGroupMappingPim-Mode pimGroupMappingPrecedence | ipmr | <p>This trap is sent when a change is detected to the active RP mapping on the device.</p> <p>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</p> <p>pimGroupMappingPimMode—The PIM mode used for groups in this group prefix.</p> <p>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</p> |

| No. | Trap Name | Objects | Family | Description |
|---|--------------------------------------|---|--------|---|
| 48 | PimInterfaceElection | pimInter- faceAddress- Type pimInter- faceAddress | 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. |
| <p>pimInterfaceAddressType—The address type of the PIM interface. pimInterfaceAddress—The primary IP address of this router on this PIM interface.</p> | | | | |
| 49 | pimBsrElectedBSRLostElection | pimBsrElected BSRAddress Type, pimBsrElected BSRAddress, pimBsrElected | ipmr | This trap is sent when the current E-BSR loses an election to a new Candidate-BSR. |
| <p>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).</p> | | | | |
| 50 | pimBsrCandidateBSRWinElection pim | pimBsrCandi- dateBSRElect- edBSR | ipmr | This trap is sent when a C-BSR wins a BSR Election. |
| <p>pimBsrCandidateBSR ElectedBSR—Indicates whether the local router is the elected BSR for this zone.</p> | | | | |

| No. | Trap Name | Objects | Family | Description |
|--|--------------------------------------|--|--------|--|
| 51 | lpsViolationTrap | lpsTrap-SwitchName lpsTrapSwitchIpAddr lpsTrap-SwitchSlice lpsTrapSwitchPort lpsTrapViolatingMac lpsTrapViolationType systemServicesDate systemServicesTime | bridge | A Learned Port Security (LPS) violation has occurred. |
| <p>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.</p> | | | | |
| 52 | lpsPortUpAfterLearningWindowExpiredT | lpsTrap-SwitchName lpsTrap-SwitchSlice lpsTrapSwitchPort systemServicesDate systemServicesTime | 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. |
| <p>lpsTrapSwitchName—The name of the switch. lpsTrapSwitchSlice—The slot number for the LPS port on which the violation occurred lpsTrapSwitchPort—The port number for the LPS port on which the violation occurred systemServicesDate—The current System Date in the following format: MM/DD/YYYY. systemServicesTime—The current System Time in the following format: HH:MM:SS.</p> | | | | |
| 53 | lpsLearnTrap | lpsLearnTrapThreshold | bridge | This trap is sent when the number of bridged MACs learned matches the configured Learned Trap Threshold. A trap is then generated or every additional MAC that is learned. |
| <p>lpsLearnTrapThreshold—The number of bridged MAC addresses that can be learned before a trap is sent.</p> | | | | |
| 54 | gvrpVlanLimitReachedEvent | alaGvrpMaxVlanLimit | 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-TrapInfo-Anomaly, alaNetSecPort-TrapInfoType | netsec | This trap is sent when and anomaly port quarantine is detected. |
| | alaNetSecPortTrapInfoIfId —The interface index of port on which anomaly is detected. alaNetSecPortTrapInfoAnomaly —The type of anomaly detected on the interface. alaNetSecPortTrapInfoType —The nature 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 anomaly 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 managed 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 | alaUdldPortIfIndex alaUdld-PrevState alaUdldCurrent-State | interface | This trap is sent when the UDLD state of a port has changed. |
| | alaUdldPortIfIndex —The interface index of the port which triggered the UDLD trap. alaUdldPrevState —The previous UDLD state of the port - notapplicable (0), shutdown (1), undetermined (2), bidirectional (3). alaUdldCurrentState —The current UDLD state of the port - notapplicable (0), shutdown (1), undetermined (2), bidirectional (3). | | | |
| 59 | ndpMaxLimitReached | none | ipv6 | This trap is sent when the hardware table has reached the maximum number of entries supported. 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”. |
| <p>alaErpRingId—The unique Ring identifier. alaErpRingState—The current state of the Ring (0=Idle, 1=Protection).</p> | | | | |
| 63 | alaErpRingMultipleRpl | alaErpRingId | erp | This trap is sent when multiple RPLs are detected in the Ring. |
| <p>alaErpRingId—The unique Ring identifier.</p> | | | | |
| 64 | alaErpRingRemoved | alaErpRingId | erp | This trap is sent when the Ring is removed dynamically. |
| <p>alaErpRingId—The unique Ring identifier.</p> | | | | |
| 65 | ntpMaxAssociation | | ntp | This trap is generated when the the maximum number of peer and client associations configured for the switch is exceeded. |
| <p>NtpMaxAssociation—The maximum number of peer and client associations that the switch will serve.</p> | | | | |
| 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. |
| <p>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.</p> | | | | |
| 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. |
| <p>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.</p> | | | | |

| 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. |
| <p>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).</p> | | | | |
| 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. |
| <p>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).</p> | | | | |
| 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 |
| <p>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).</p> | | | | |
| 71 | webMgtServerErrorTrap | webMgtServerError | webmgt | This trap is sent when the Web Management server goes into error state after crashing twice within a minute. |
| <p>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.</p> | | | | |
| 72 | multiChassisIpcVlanUp | multiChassisTrapIpcVlan | multi-chassis | Indicates the operational status for the multi-chassis communication VLAN is Up. |
| <p>multiChassisTrapIpcVlan—The multi-chassis IPC VLAN number.</p> | | | | |
| 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-chassis | This trap is sent when there is an MCM misconfiguration (e.g., inconsistent chassis ID or IPC VLAN). |
| multiChassisTrapFailure —Indicates multi-chassis failure. | | | | |
| 75 | multiChassisHelloIntervalConsistencyFailure | multiChassis-TrapFailure | multi-chassis | This trap is sent when there is an MCM Hello Interval consistency failure. |
| multiChassisTrapFailure —Indicates multi-chassis failure. | | | | |
| 76 | multiChassisStpModeConsistencyFailure | multiChassis-TrapFailure | multi-chassis | This trap is sent when there is an STP mode consistency failure. |
| multiChassisTrapFailure —Indicates multi-chassis failure. | | | | |
| 77 | multiChassisStpPathCostModeConsistencyFailure | multiChassis-TrapFailure | multi-chassis | This trap is sent when there is an STP path cost mode consistency failure. |
| multiChassisTrapFailure —Indicates multi-chassis failure.. | | | | |
| 78 | multiChassisVfLinkStatusConsistencyFailure | multiChassis-TrapFailure | multi-chassis | This trap is sent when there is an MCM Virtual Fabric Link status consistency failure |
| multiChassisTrapFailure —Indicates multi-chassis failure. | | | | |
| 79 | multiChassisStpBlockingStatus | multiChassis-TrapStpBlockingVlanList | multi-chassis | 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, separated by commas. | | | | |
| 80 | multiChassisLoopDetected | multiChassis-TrapFailure | multi-chassis | This trap is sent when a loop is detected. |
| multiChassisTrapFailure —Indicates multi-chassis failure. | | | | |
| 81 | multiChassisHelloTimeout | multiChassis-TrapFailure | multi-chassis | This trap is sent when the Hello Timer expires. |
| multiChassisTrapFailure —Indicates multi-chassis failure. | | | | |
| 82 | multiChassisVfLinkDown | multiChassis-TrapFailure | multi-chassis | This trap is sent when the Virtual Fabric Link goes down. |
| multiChassisTrapFailure —Indicates multi-chassis failure. | | | | |
| 83 | multiChassisVFLMemberJoinFailure | multiChassis-TrapVFL, multiChassisTrapVFLMemberPort, multiChassisTrapDiagnostic | multi-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 |
|-----|---------------------------------------|--|-----------|--|
| | multiChassisTrapVFL | —The multi-chassis Virtual Fabric Link interface. | | |
| | multiChassisTrap VFLMemberPort | —The multi-chassis VFL member port number. | | |
| | multiChassisTrapDiagnostic | —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 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 | vRtrIsisSystemLevel isisSysL1 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 | isis- ManAreaAd- drExistState | isis | <p>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.</p> <p>This trap is edge triggered, and should not be regenerated until an address that was used in the previous computation has been dropped.</p> <p>isisManAreaAddrExistState—The area ID that was ignored when computing routes.</p> |
| 90 | vRtrIsisCorruptedLSPDetected | vRtrIsisSystem- Level vRtrIsis- TrapLSPID | isis | <p>This trap is sent when an LSP that was stored in memory has become corrupted.</p> <p>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.</p> <p>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.</p> <p>vRtrIsisTrapLSPID—An Octet String that uniquely identifies a Link State PDU.</p> |
| 91 | vRtrIsisMaxSeqExceedAttempt | vRtrIsisSys- temLevel vRtrIsis- TrapLSPID | isis | <p>This trap is sent when the sequence number on an LSP wraps the 32 bit sequence counter.</p> <p>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.</p> <p>vRtrIsisTrapLSPID—An Octet String that uniquely identifies a Link State PDU.</p> |
| 92 | vRtrIsisIDLenMismatch | vRtrIsisField- Len vRtrIsisIfIndex vRtrIsisPDU- Fragment | isis | <p>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.</p> <p>vRtrIsisFieldLen—The System ID Field length.</p> <p>vRtrIsisIfIndex—The ISIS interface on which the PDU was received.</p> <p>vRtrIsisPDUFragment—The first 64 bytes of a PDU that triggered the trap.</p> |

| No. | Trap Name | Objects | Family | Description |
|--|-----------------------------|---|--------|---|
| 93 | vRtrIsisMaxAreaAdrsMismatch | vRtrIsisMax- AreaAddress, vRtrIsisIfIndex vRtrIsisPDU- Fragment | isis | This trap is sent when a PDU with a different Maximum Area Addresses value is received. The notification includes the header of the packet, which may help a network manager identify the source of the problem. |
| <p>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.</p> | | | | |
| 94 | vRtrIsisOwnLSPPurge | vRtrIsisIfIndex, vRtrIsis- TrapLSPID vRtrIsisSystem- Level | isis | This trap is 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 the problem. |
| <p>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.</p> | | | | |
| 95 | vRtrIsisSequenceNumberSkip | vRtrIsis- TrapLSPID 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. |
| <p>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.</p> | | | | |
| 96 | vRtrIsisAutTypeFail | vRtrIsisSystem- Level, vRtrIsisPDU- Fragment, 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. |
| <p>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.</p> | | | | |

| No. | Trap Name | Objects | Family | Description |
|-----|---------------------------|---|--------|--|
| 97 | vRtrIsisAuthFail | vRtrIsisSystem- Level, vRtrIsisPDU- Fragment, 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 vRtrIsisPDU- Fragment 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 vRtrIsisPDU- Fragment | 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 vRtrIsis- TrapLSPID vRtrIsisIfIndex | isis | This trap is sent when an LSP is larger than the Data Link Block Size for a circuit. |
| <p>vRtrIsisLSPSize—The size of the LSP received.</p> <p>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.</p> <p>vRtrIsisTrapLSPID—An Octet String that uniquely identifies a Link State PDU.</p> <p>vRtrIsisIfIndex—The ISIS interface on which the LSP was received.</p> | | | | |
| 102 | vRtrIsisOrigLSPBufSizeMismatch | vRtrIsisOriginatingBuffer- Size vRtrIsisSystem- Level vRtrIsis- TrapLSPID 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. |
| <p>vRtrIsisOriginatingBufferSize—The buffer size advertised by the peer.</p> <p>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.</p> <p>vRtrIsisTrapLSPID—An Octet String that uniquely identifies a Link State PDU.</p> <p>vRtrIsisIfIndex—The ISIS interface on which the LSP was received.</p> | | | | |
| 103 | vRtrIsisProtoSuppMismatch | vRtrIsisProtocolsSupported vRtrIsisSystem- Level vRtrIsis- TrapLSPID vRtrIsisIfIndex | isis | <p>This trap is sent when a non-pseudonode segment 0 LSP is received that has no matching protocols supported.</p> <p>This may be because the system does not generate the field, or because there are no common elements.</p> <p>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.</p> |
| <p>vRtrIsisProtocolsSupported—The protocols supported by an adjacent system. This may be empty</p> <p>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.</p> <p>vRtrIsisTrapLSPID—An Octet String that uniquely identifies a Link State PDU.</p> <p>vRtrIsisIfIndex—The ISIS interface on which the LSP was received.</p> | | | | |

| No. | Trap Name | Objects | Family | Description |
|-----|--------------------------------|---|--------|---|
| 104 | vRtrIsisAdjacencyChange | vRtrIsisSystemLevel vRtrIsisIfIndex vRtrIsisTrapLSPID isisISAdjState | isis | <p>This trap is sent when adjacency changes state, entering or leaving state up.</p> <p>The first 6 bytes of the vRtrIsisTrapLSPID are the SystemID of the adjacent IS. The isisISAdjState is the new state of the adjacency.</p> <p>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.</p> <p>vRtrIsisIfIndex—The ISIS interface on which the trap was received.</p> <p>vRtrIsisTrapLSPID—An Octet String that uniquely identifies a Link State PDU.</p> <p>isisISAdjState—The state of the adjacent router.</p> |
| 105 | vRtrIsisCircIdExhausted | vRtrIsisIfIndex | isis | <p>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.</p> <p>When this happens, the interface is marked operationally down. When an operationally up interface is deleted, the Circuit ID can be reused by any interface waiting to receive a unique Circuit ID.</p> <p>vRtrIsisIfIndex—The ISIS interface.</p> |
| 106 | vRtrIsisAdjRestartStatusChange | vRtrIsisSystemLevel vRtrIsisIfIndex vRtrIsisISAdjRestartStatus | isis | <p>This trap is sent when an adjacency's graceful restart status changes.</p> <p>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.</p> <p>vRtrIsisIfIndex—The ISIS interface.</p> <p>vRtrIsisISAdjRestartStatus—The new graceful restart state of the adjacency.</p> |
| 107 | alaMvrpVlanLimitReachedEvent | alaMvrpMaxVlanLimit | bridge | <p>This trap is sent when the number of VLANs learned dynamically by MVRP reaches the configured limit.</p> <p>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 creating additional VLANs (32 - 4094, Default = 256).</p> |

| No. | Trap Name | Objects | Family | Description |
|--|------------------------------|--|---------|---|
| 108 | alaHAVlanClusterPeerMismatch | alaHAVlanClusterId | ha-vlan | This trap is sent when parameters configured for this cluster ID (Level 1 check) does not match across the MLAG 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 MLAG 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 MLAG. |
| 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 MLAG on which the MAC ingressed. | | | | |
| alaDaUnpVlan —The VLAN on which the MAC was classified on the local chassis. | | | | |
| alaDaUnpMCLAGId —The Link Agg Id for MLAG. | | | | |

| No. | Trap Name | Objects | Family | Description |
|-----|---|--|---------|--|
| 112 | unpMcLagConfigInconsistency | alaDaUnpCom- mandType alaDaUnpName alaDaUnp- MacAddr1 alaDaUnp- MacAddr2 alaDaUn- pIpAddr alaDaUnpIp- Mask alaDaUnpVlan- Tag alaDaUnpMC- LAGId | da-unp | This trap is sent when a configuration becomes "Out of Sync". |
| | <p>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).</p> <p>alaDaUnpName—Indicates which UNP Profile is out-of-sync. If there is no UNP Profile associated, a zero length string is sent.</p> <p>alaDaUnpMacAddr1—The MAC for MAC rule or the lower limit of MAC Range Rule.</p> <p>alaDaUnpMacAddr2—The upper limit of MAC Range Rule.</p> <p>alaDaUnpIpAddr—The IP address in the IP Rule.</p> <p>alaDaUnpIpMask—The IP Mask of the IP address in the IP Rule.</p> <p>alaDaUnpVlanTag—The VLAN VLAN Tag Rule. A zero value means it is not applicable.</p> <p>alaDaUnpMCLAGId—The Link Agg ID for MLAG.</p> | | | |
| 113 | multiChassisGroupConsisFailure | multiChassis- TrapFailure | 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 | pimNeighbor- AdressType, pimNeighboAd- dress | 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. |
| | <p>pimNeighborAdressType—The address type of the PIM neighbor.</p> <p>pimNeighborAddress—The primary IP address of the PIM neighbor. The InetAddressType is given by the pimNeighborAddressType object.</p> | | | |
| 116 | dot1agCfmFaultAlarm | dot1agCfmMe- pHighest- PrDefect | 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 |
|--|--------------------------------|--|-----------------|---|
| <p>dot1agCfmMepHighestPrDefect—The highest priority defect that has been present since the MEPs Fault Notification Generator State Machine was last in the FNG_RESET state.</p> | | | | |
| 117 | alaSaaIPIterationCompleteTrap | alaSaaCtrlOwnerIndex, alaSaaCtrlTestIndex, alaSaaIpResultsTestRunIndex, alaSaaCtrlLastRunResult, alaSaaCtrlLastRunTime | saa | This trap is sent when an IP SAA iteration is completed. |
| <p>alaSaaCtrlOwnerIndex—The Owner name to identify entries in the table. This is currently not supported and its value will always be the string 'USER'.</p> <p>alaSaaCtrlTestIndex—A Unique name to identify the entries in the table. The name is unique across various SNMP users.</p> <p>alaSaaIpResultsTestRunIndex—The row entry that reports results for a single OAM test run. The value of this object starts at 1 and can go up to a maximum of alaSaaCtrlMaxHistoryRows.</p> <p>alaSaaCtrlLastRunResult—The result of the latest SAA test iteration (Undetermined/Success/Failed/Aborted).</p> <p>alaSaaCtrlLastRunTime—The time the last iteration of the SAA was run.</p> | | | | |
| 118 | alaSaaEthIterationCompleteTrap | alaSaaCtrlOwnerIndex, alaSaaCtrlTestIndex, alaSaaEthoamResultsTestRunIndex, alaSaaCtrlLastRunResult, alaSaaCtrlLastRunTime | saa | This trap is sent when an eth-LB or Eth-DMM SAA iteration is completed. |
| <p>alaSaaCtrlOwnerIndex—The Owner name to identify entries in the table. This is currently not supported and its value will always be the string 'USER'.</p> <p>alaSaaCtrlTestIndex—A Unique name to identify the entries in the table. The name is unique across various SNMP users.</p> <p>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 up to a maximum of alaSaaCtrlMaxHistoryRows.</p> <p>alaSaaCtrlLastRunResult—The result of the latest SAA test iteration (Undetermined/Success/Failed/Aborted).</p> <p>alaSaaCtrlLastRunTime—The time the last iteration of the SAA was run.</p> | | | | |
| 119 | alaSaaMacIterationCompleteTrap | NA | saa | This trap is sent when a MAC iteration is complete. |
| NA | | | | |
| 120 | virtualChassisStatusChange | virtualChassisOperChasId, virtualChassisStatus | virtual chassis | This trap is sent when a chassis status change is detected. |

| No. | Trap Name | Objects | Family | Description |
|--|-------------------------------------|--|-----------------|--|
| virtualChassisOperChasId —The operational Virtual Chassis ID. virtualChassisStatus —The Virtual Chassis status. | | | | |
| 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) : Chassis is in slave role after election. inconsistent(3) : Chassis 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 |
|-----|--|---|--------|---|
| | <p>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).</p> | | | |
| 125 | lldpRemTablesChange | lldpStatsRem TablesInserts, lldpStatsRem TablesDe- letes, lldpStatsRem TablesDrops, lldpStatsRem TablesAge- outs | lldp | This trap is sent when the value of lldpStatsRemTablelastChange Time changes. It can be utilized by an NMS to trigger LLDP remote systems table maintenance polls. |
| | <p>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.</p> | | | |
| 126 | vRtrLdpInstanceStateChange | vRtrLdpGen- AdminState, vRtrLd- pGenOper- State, vRtrLdpInstan- ceNotifyRea- sonCode | lldp | This trap is sent when the LDP module changes state either administratively or operationally. |
| | <p>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)</p> | | | |
| 127 | evbFailedCdcplvTrap | evbPortId | evb | This trap is sent when bridge receives a CDCP packet with: <ul style="list-style-type: none"> - Wrong TLV type, or - Wrong OUI, or - Role is set to Bridge, or - Wrong default channel(scid), or - Incorrect channel number(scid). |
| | <p>evbPortId—The IfIndex that uniquely identifies this port.</p> | | | |
| 128 | evbFailedEvlvTrap | evbPortId, ieee8021Bridge Evlv- SIVlanId | evb | This trap is sent when bridge receives an EVBTLV packet with: <ul style="list-style-type: none"> - Wrong TLV type. or - Incorrect TLV length, or - Wrong OUI. |

| No. | Trap Name | Objects | Family | Description |
|--|----------------------------|---|--------|---|
| <p>evbPortId—The IfIndex that uniquely identifies this port. ieee8021BridgeEvbVSIvlanId—The bridge EVB VSI VLAN.</p> | | | | |
| 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. |
| <p>evbPortId—The IfIndex that uniquely identifies this port. ieee8021BridgeEvbSbpPortNumber—The bridge EVN SBP Port.</p> | | | | |
| 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 |
| <p>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:</p> <ul style="list-style-type: none"> • vsiidIpv4 (1) • vsiidIpv6 (2) • vsiidMAC (3) • vsiidLocal (4) • vsiidUUID (5) <p>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.</p> | | | | |
| 131 | evbCdcplldpExpiredTrap | | evb | This trap is sent when an LLDP Timer expires in bridge. The timer expires when LLDP does not receive CDCP TLV within a specified interval. |
| 132 | evbTlvExpiredTrap | | evb | This trap is sent when an LLDP Timer expires in bridge. The timer expires when LLDP does not receive EVB TLV within a specified interval. |
| 133 | evbVdpKeepaliveExpiredTrap | | evb | This trap is sent when a VDP Keep Alive Timer expires in bridge. The timer expires when the bridge does not receive VDP Keep Alive message within a specified interval. |

| No. | Trap Name | Objects | Family | Description |
|--|--------------------|--|--------------------|---|
| 134 | smgrServiceError | alaSvcId, alaSvcType, alaSvcIsid, alaSvcBVlan, alaSvcMulti- castMode | service manager | This trap is sent when there is a failure to create/delete a service. |
| <p>alaSvcId—The Service identifier.</p> <p>alaSvcType—The service type (e.g., vpls, spb).</p> <p>alaSvcIsid—The I-Domain Service Identifier (I-SID), which identifies the service instance in a PBB network in a BEB switch. For a customer packet flowing to the B-Domain, the I-SID is derived from the VFI and inserted into the packet. For a packet flowing from the B-Domain, the I-SID is used to identify the VFI for the I-Domain processing.</p> <p>alaSvcBVlan—The Backbone VLAN ID (BVLAN), which defines the B-Domain for the PBB traffic.</p> <p>alaSvcMulticastMode— The multicast replication mode for each service:</p> <ul style="list-style-type: none"> • Head-End (1) - where a non-unicast packet entering the SAP port is replicated once for each receiver in the B-Domain using its unicast BMAC. • Tandem (2) - where a non-unicast packet entering the SAP port is replicated once at each node using the multicast group address. | | | | |
| 135 | smgrServiceHwError | alaSvcId, alaSvcType, alaSvcVFI, alaSvcMcIndex | service manager | This trap is sent when there is a failure to allocate/de-allocate a hardware resource for a service, or to program the hardware tables for a service. |
| <p>alaSvcId—The Service identifier.</p> <p>alaSvcType—The service type (e.g., vpls, spb).</p> <p>alaSvcVFI—The Virtual Forwarding Instance (VFI) allocated for a service on an LER or BEB switch. This service instance defines the forwarding plane for the data packets among virtual port members associated with the VFI. The VFI has one-to-one mapping relationship with the Service ID for this service instance.</p> <p>alaSvcMcIndex—The Multicast Index associated with a VFI, which is used to setup the multicast replication logic for this service instance on the LER or BEB switch. The McIndex has one-to-one mapping relationship with the Service ID for this service instance.</p> | | | | |
| 136 | smgrSapError | alaSvcId, alaSapPortId, alaSapEn- capValue | service manager | This trap is sent when there is a failure to create/delete a Service Access Point. |
| <p>alaSvcId—The Service identifier.</p> <p>alaSapPortId—The ID of the access port where this SAP is defined.</p> <p>alaSapEncapValue—The value of the label used to identify this SAP on the access port specified by the SAP Port ID.</p> | | | | |
| 137 | smgrSapHwError | alaSvcId, alaSapPortId, alaSapEn- capValue, alaSvcVFI, alaSapVirtual- Port | service manager | This trap is sent when there is a failure to allocate/de-allocate a hardware resource for a SAP, or to program the hardware tables for a SAP. |

| No. | Trap Name | Objects | Family | Description |
|-----|------------------|---|--------------------|---|
| | | | | <p>alaSvcId—The Service identifier.</p> <p>alaSapPortId—The ID of the access port where this SAP is defined.</p> <p>alaSapEncapValue—The value of the label used to identify this SAP on the access port specified by SAP Port ID.</p> <p>alaSvcVFI—The Virtual Forwarding Instance (VFI) allocated for a service on an LER or BEB switch. This service instance defines the forwarding plane for the data packets among virtual port members associated with the VFI. The VFI has one-to-one mapping relationship with the Service ID for this service instance.</p> <p>alaSapVirtualPort—The</p> |
| 138 | smgrSdpError | alaSdpId, alaSdpNetwork- Port, alaSdpBVlan, alaSdpSystemId | service manager | This trap is sent when there is a failure to create/delete a Service Distribution Point. |
| | | | | <p>alaSvcId—The Service identifier.</p> <p>alaSdpNetworkPort—The network port where ISIS discovered the neighbor node information (B-VLAN and B-MAC).</p> <p>alaSdpBVlan—The Backbone VLAN (B-VLAN) where ISIS discovered the neighbor node information (B-VLAN and B-MAC).</p> <p>alaSdpSystemId—The Backbone MAC (B-MAC) where ISIS discovered the neighbor node information (B-VLAN and B-MAC).</p> |
| 139 | smgrSdpHwErrorr | alaSdpId, alaSdpNetwork- Port, alaSdpBVlan, alaSdpSystemId | service manager | This trap is sent when there is a failure to allocate/de-allocate a hardware resource for an SDP, or to program the hardware tables for an SDP. |
| | | | | <p>alaSvcId—The Service identifier.</p> <p>alaSdpNetworkPort—The network port where ISIS discovered the neighbor node information (B-VLAN and B-MAC).</p> <p>alaSdpBVlan—The Backbone VLAN (B-VLAN) where ISIS discovered the neighbor node information (B-VLAN and B-MAC).</p> <p>alaSdpSystemId—The Backbone MAC (B-MAC) where ISIS discovered the neighbor node information (B-VLAN and B-MAC).</p> |
| 140 | smgrSdpBindError | alaSvcId, alaSdpBindId, alaSdpBindNet- workPort, alaSdpBindB- Vlan, alaSdpBindSys- temId | service manager | This trap is sent when there is a failure to create/delete an SDP Bind. |
| | | | | <p>alaSvcId—The Service identifier.</p> <p>alaSdpBindId—The SDP Binding identifier.</p> <p>alaSdpBindNetworkPort—The network port associated with a service instance where MPLS-labeled or B-Domain packets ingress and egress.</p> <p>alaSdpBindBVlan—The Backbone VLAN ID (B-VLAN) associated with the SDP Bind object.</p> <p>alaSdpBindSystemId—The neighbor Backbone MAC (B-MAC) associated with the SDP Bind object.</p> |

| No. | Trap Name | Objects | Family | Description |
|---|------------------------------------|---|--------------------|---|
| 141 | smgrSdpBindHwError | alaSvcId, alaSdpBindId, alaSdpBindNetworkPort, alaSdpBindB-Vlan, alaSdpBindSystemId, alaSdpBindVirtualPort | service manager | This trap is sent when there is a failure to allocate/de-allocate a hardware resource for an SDP Bind, or to program the hardware tables for an SDP Bind. |
| <p>alaSvcId—The Service identifier. alaSdpBindId—The SDP Binding identifier. alaSdpBindNetworkPort—The network port associated with a service instance where MPLS-labeled or B-Domain packets ingress and egress. alaSdpBindB-Vlan—The Backbone VLAN ID (B-VLAN) associated with the SDP Bind object. alaSdpBindSystemId—The neighbor Backbone MAC (B-MAC) associated with the SDP Bind object. alaSdpBindVirtualPort—The logical representation of a network port associated with a service instance where MPLS-labeled or B-Domain packets ingress and egress.</p> | | | | |
| 142 | smgrGeneralError | alaSvcId, alaSvcType | service manager | This trap is sent when there is a .general system failure detected during normal system operation |
| <p>alaSvcId—The Service identifier. alaSvcType—The service type (e.g. vpls, spb).</p> | | | | |
| 143 | smgrStatusChange | alaSvcId, alaSvcType, alaSvcOperStatus, alaSvcNumSaps, alaSvcNumSdps | service manager | This trap is sent when there is a status change for a group of selected services. |
| <p>alaSvcId—The Service identifier. alaSvcType—The service type (e.g., vpls, spb). alaSvcOperStatus—The operating state of this service. The requirements for a service to be operationally up depend on the service type: TLS Services are 'up' when the service is administratively up and either at least two SAP's or spoke SDP Bind's, or one SAP or spoke SDP Bind and at least one mesh SDP Bind are operationally up. alaSvcNumSaps—The number of SAPs defined on this service. alaSvcNumSdps—The number of SDPs bound to this service.</p> | | | | |
| 144 | portViolationNotificationTrap | ifIndex | port | This trap is sent when a port violation is cleared. |
| <p>ifIndex—A unique value, greater than zero, for the interface.</p> | | | | |
| 145 | multiChassisConsisFailureRecovered | multiChassisConsisFailureRecovered | multi-chassis | This trap is sent when the system has recovered from a multi-chassis inconsistency between the local and peer switches. |
| <p>multiChassisConsisFailureRecovered—Indicates that the system has recovered from a multi-chassis failure.</p> | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---------------------------------|--|--------|--|
| 146 | alaSaaPacketLossTrap | alaSaaCtrlOwnerIndex, alaSaaCtrlTestIndex, alaSaaCtrlLastRunResult, alaSaaCtrlLastRunTime, alaSaaMacResultsPktsSent, alaSaaMacResultsPktsRcvd | saa | This trap is sent when a packet is lost during a test. |
| <p>alaSaaCtrlOwnerIndex—The Owner name to identify the responsibility of the entries in the table (Default = User).</p> <p>alaSaaCtrlTestIndex—Unique name to identify the entries in the table. The name is unique across various SNMP users (up to 32 characters).</p> <p>alaSaaCtrlLastRunResult—The result of the latest SAA test iteration (Undertermined (0), Success (1), Failed (2), Aborted (3)).</p> <p>alaSaaCtrlLastRunTime—The time the last iteration of the SAA was run.</p> <p>alaSaaMacResultsPktsSent—The number of packets sent during a single MAC-Ping iteration.</p> <p>alaSaaMacResultsPktsRcvd—The number of packets received during a single MAC-Ping iteration.</p> | | | | |
| 147 | alaSaaJitterThresholdYellowTrap | alaSaaCtrlOwnerIndex, alaSaaCtrlTestIndex, alaSaaCtrlLastRunResult, alaSaaCtrlLastRunTime, alaSaaCtrlJitterThreshold, alaSaaMacResultsAvgJitter | saa | This trap is sent when the Jitter Threshold crosses 90%. |
| <p>alaSaaCtrlOwnerIndex—The Owner name to identify the responsibility of the entries in the table (Default = User).</p> <p>alaSaaCtrlTestIndex—Unique name to identify the entries in the table. The name is unique across various SNMP users (up to 32 characters).</p> <p>alaSaaCtrlLastRunResult—The result of the latest SAA test iteration (Undertermined (0), Success (1), Failed (2), Aborted (3)).</p> <p>alaSaaCtrlLastRunTime—The time the last iteration of the SAA was run.</p> <p>alaSaaCtrlJitterThreshold—The configured Jitter Threshold (Range = 0 - 1,000,000, Default = 0)</p> <p>alaSaaMacResultsAvgJitter—The average jitter value.</p> | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|------------------------------|--|--------|---|
| 148 | alaSaaRTTThresholdYellowTrap | alaSaaCtrlOwnerIndex, alaSaaCtrlTestIndex, alaSaaCtrlLastRunResult, alaSaaCtrlLastRunTime, alaSaaCtrlRTTThreshold, alaSaaMacResultsAvgRTT | saa | This trap is sent when the RTT Threshold crosses 90%. |
| <p>alaSaaCtrlOwnerIndex—The Owner name to identify the responsibility of the entries in the table (Default = User).</p> <p>alaSaaCtrlTestIndex—Unique name to identify the entries in the table. The name is unique across various SNMP users (up to 32 characters).</p> <p>alaSaaCtrlLastRunResult—The result of the latest SAA test iteration (Underdetermined (0), Success (1), Failed (2), Aborted (3)).</p> <p>alaSaaCtrlLastRunTime—The time the last iteration of the SAA was run.</p> <p>alaSaaCtrlRTTThreshold—The configured RTT Threshold, micro-seconds (Range = 0 - 1,000,000, Default = 0)</p> <p>alaSaaMacResultsAvgRTT—The average Round Trip Time.</p> | | | | |
| 149 | alaSaaJitterThresholdRedTrap | alaSaaCtrlOwnerIndex, alaSaaCtrlTestIndex, alaSaaCtrlLastRunResult, alaSaaCtrlLastRunTime, alaSaaCtrlJitterThreshold, alaSaaMacResultsAvgJitter | saa | This trap is sent when the Jitter threshold is crossed. |
| <p>alaSaaCtrlOwnerIndex—The Owner name to identify the responsibility of the entries in the table (Default = User).</p> <p>alaSaaCtrlTestIndex—Unique name to identify the entries in the table. The name is unique across various SNMP users (up to 32 characters).</p> <p>alaSaaCtrlLastRunResult—The result of the latest SAA test iteration (Underdetermined (0), Success (1), Failed (2), Aborted (3)).</p> <p>alaSaaCtrlLastRunTime—The time the last iteration of the SAA was run.</p> <p>alaSaaCtrlJitterThreshold—The configured Jitter Threshold (Range = 0 - 1,000,000, Default = 0)</p> <p>alaSaaMacResultsAvgJitter—The average jitter value.</p> | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|----------------------------------|--|-----------------|--|
| 150 | alaSaaRTTThresholdRedTrap | alaSaaCtrlOwnerIndex, alaSaaCtrlTestIndex, alaSaaCtrlLastRunResult, alaSaaCtrlLastRunTime, alaSaaCtrlRTTThreshold, alaSaaMacResultsAvgRTT | saa | This trap is sent when the RTT threshold is crossed. |
| <p>alaSaaCtrlOwnerIndex—The Owner name to identify the responsibility of the entries in the table (Default = User).</p> <p>alaSaaCtrlTestIndex—Unique name to identify the entries in the table. The name is unique across various SNMP users (up to 32 characters).</p> <p>alaSaaCtrlLastRunResult—The result of the latest SAA test iteration (Underdetermined (0), Success (1), Failed (2), Aborted (3)).</p> <p>alaSaaCtrlLastRunTime—The time the last iteration of the SAA was run.</p> <p>alaSaaCtrlRTTThreshold—The configured RTT Threshold, micro-seconds (Range = 0 - 1,000,000, Default = 0)</p> <p>alaSaaMacResultsAvgRTT—The average Round Trip Time.</p> | | | | |
| 151 | chassisTrapsDuplicateMacClear | physicalIndex, baseMacAddress | chassis | This trap is sent when the old Master Chassis has rejoined the Virtual Chassis as a slave. There is no longer a possibility of duplicate MAC address in the network. |
| <p>physicalIndex—The physical index of the involved object.</p> <p>baseMacAddress—The base MAC Address.</p> | | | | |
| 152 | alaFipsConfigFilterResourceLimit | NA | fips | The allowed maximum percentage of filter resources configured from the allocated FIPS resources is exceeded. (Range = 0 - 100, Default = 80) |
| NA | | | | |
| 153 | virtualChassisUpgradeComplete | virtualChassisUpgradeCompleteStatus | virtual chassis | Critical trap indicates whether the software upgrade process has failed after a timeout or completed successfully. Note that if the process fails, it may be still possible for the system to recover if the process successfully completes later after the expired timeout. |
| <p>virtualChassisUpgradeCompleteStatus—The Virtual Chassis upgrade completion status (Success (1), Failure (2)).</p> | | | | |

| No. | Trap Name | Objects | Family | Description |
|--|----------------------------------|--|----------------------|---|
| 154 | appFPSignatureMatchTrap | alaAppFPPort, alaAppFPD- bAppGroup- Name, alaAppFPD- bAppName, alaAppF- PDbSrc- MacAddr, alaAppFPDb- VlanId, alaAppFPDbSr- cIpAddrType, alaAppFPDbSr- cIpAddr, alaAppF- PDbSrcPort | app finger- print | This trap is sent when a traffic flow matches an application signature. |
| <p>alaAppFPPort—The port on which the flow was classified</p> <p>alaAppFPDbAppGroupName—The name of the application group and signature that matched the flow.</p> <p>alaAppFPDbAppName—The name of the application that matched the flow.</p> <p>alaAppFPDbSrcMacAddr—The source MAC addresses of the flow.</p> <p>alaAppFPDbVlanId—The VLAN on which the flow was learned and forwarded.</p> <p>alaAppFPDbSrcIpAddrType—The source IP address type.</p> <p>alaAppFPDbSrcIpAddr—The source IP address.</p> <p>alaAppFPDbSrcPort—The source port number.</p> | | | | |
| 155 | virtualChassisVflSpeedTypeChange | virtualChas- sisOper- ChasId, virtualChassis- VflId, virtualChassis- VflSpeed- Type | virtual- chassis | This trap is sent when the VFL speed type is changed. |
| <p>virtualChassisOperChasId—The operational Virtual-Chassis ID.</p> <p>virtualChassisVflId—The Virtual Fabric Link Interface If Index.</p> <p>virtualChassisVflSpeedType—The Virtual Chassis VFL speed type:</p> <ul style="list-style-type: none"> • Unassigned - VFL speed type is unassigned. • Unknown - VFL speed is unknown. • Mismatch - This VFL has member ports operating at different speeds. • Ten GB - All member ports of this VFL are operating at 10 Gbps. • Forty GB - All member ports of this VFL are operating at 40 Gbps. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|--------------------------------------|---|--------------|---|
| 156 | alaSIPsnoopingACLPreemptedBySO-SCall | physicalIndex, alaSIPsnoopingEndedCallIpAddrA, alaSIPsnoopingEndedCallIpAddrB, alaSIPsnoopingEnded-CallL4portA, alaSIPsnoopingEnded-CallL4portB | sip snooping | This trap is sent when a SIP snooping RTP/RTCP ACL entry is preempted by an SOS call. |
| <p>physicalIndex—The physical index of the involved object. alaSIPsnoopingEndedCallIpAddrA—The Ended Call IP address for direction A to B. alaSIPsnoopingEndedCallIpAddrB—The Ended Call IP address for direction B to A. alaSIPsnoopingEndedCallL4portA—The Ended call L4port for call direction A to B. alaSIPsnoopingEndedCallL4portB—The Ended call L4port for call direction A to B.</p> | | | | |
| 157 | alaSIPsnoopingRTCPOverThreshold | alaSIPsnoopingActiveCallIpAddrA, alaSIPsnoopingActiveCallIpAddrB, alaSIPsnoopingActive-CallL4portB, alaSIPsnoopingActiveCallSipMediaType, alaSIPsnoopingCallViolationType | sip snooping | This trap is sent when one or more RTCP parameters exceeds the threshold limit. |
| <p>alaSIPsnoopingActiveCallIpAddrA—The Active Call IP address for direction A to B. alaSIPsnoopingActiveCallIpAddrB—The Active Call IP address for direction B to A. alaSIPsnoopingActiveCallL4portB—The Active call L4port for call direction B to A. alaSIPsnoopingCallViolationType—The type of the Active Call violation: jitterViolation (1), --jitter violation:</p> <ul style="list-style-type: none"> • RTD • MOS • Rfactor • Packet Loss | | | | |
| 158 | alaSIPsnoopingRTCPPktsLost | physicalIndex | sip snooping | This trap is sent when RTCP packets are lost due to rate limiting. |
| <p>physicalIndex—The physical index of the involved object.</p> | | | | |
| 159 | alaSIPsnoopingSignallingLost | physicalIndex | sip snooping | This trap is sent when SIP signalling messages are lost due to rate limiting. |

| No. | Trap Name | Objects | Family | Description |
|---|--|---|-------------------|--|
| physicalIndex —The physical index of the involved object. | | | | |
| 160 | alaSIPsnoopingCallRecordsFile-Moved | alaSIPsnoop- ingThreshold- NumberOfCa lls | sip snoop- ing | This trap is sent when the SIP Snooping Ended Call Records flash file is moved from /flash/switch/sip_call_record.txt to /flash/switch/sip_call_re- cord.txt.old. This happens when the configured call record storage limit is reached and possibly at boot-up if /flash/switch/sip_ call_record.txt from previous run exists at the first check. |
| alaSIPsnoopingThresholdNumberOfCalls —The number of call records that can be stored on the device (Range = 50 - 500, Default = 200). | | | | |
| 161 | alaIPv6NeighborLimitExceeded | | ipv6 | This trap is sent when |
| 162 | alaIPv6NeighborVRFLimitExceeded | | ipv6 | This trap is sent when |
| 163 | alaIPv6InterfaceNeighborLimitExceed | | ipv6 | This trap is sent when |
| 164 | alaDyingGaspTrap | alaDying- GaspSlot, alaDyingGasp- PowerSupply- Type, alaDyingGasp- Time | interface | This trap is sent when a switch has lost all power. |
| alaDyingGaspSlot —The slot number of the chassis whose NI is going down. alaDyingGaspPowerSupplyType —The type of the power supply. alaDyingGaspTime —The time of the failure. | | | | |
| 165 | alaDhcpSrvLeaseUtilizationThreshold | alaDhcpSrv- LeaseThresh- oldStatus, alaDhcpSrv- SubnetDescrip- tor | dhcpsrv | This trap is sent when the lease utilization on a subnet exceeds or falls below the configured threshold value. |
| alaDhcpSrvLeaseThresholdStatus —The threshold status of subnet utilization. alaDhcpSrvSubnetDescriptor —The subnet Descriptor. If the subnet belongs to a shared network, this object specifies the shared network name; otherwise, it specifies the Subnet IP | | | | |
| 166 | alaDHCPv6SrvLeaseUtilization- Threshold | alaDhcpv6Srv- LeaseThresh- oldStatus, alaDhcpSrv- SubnetDescrip- tor | dhcpsrv | This trap is sent when the lease utilization on a subnet exceeds or falls below the configured threshold value. |

| No. | Trap Name | Objects | Family | Description |
|--|-------------------------|---|--------|---|
| 167 | smgrServiceStatusChange | alaSvcId, alaSvcType, alaSvcOperSta- tus, alaSvcIsid, alaSvcBVlan, alaSvcMulti- castMode | svcmgr | This trap is sent when there is a change in service operating status. A service is operationally up when it's admin-up and there's at least one active SAP or one active bind that is operationally up. |
| <p>alaSvcId—The unique Service identifier.</p> <p>alaSvcType—The Service Type (e.g. VPLS, SPB).</p> <p>alaSvcOperStatus—The operating state of the Service. The requirements for a Service to be operationally up depend on the Service Type: TLS Services are 'up' when the Service is administratively up and either at least two SAP's or spoke SDP Bind's, or one SAP or spoke SDP Bind and at least one mesh SDP Bind are operationally up.</p> <p>alaSvcIsid—The I-Domain Service Identifier (I-SID) which identifies the Service instance in a PBB network on a BEB switch. For a customer packet flowing toward the B-Domain, the I-SID is derived from the VFI and inserted into the packet. For a packet flowing from B-Domain, the I-SID is used to identify the VFI for the i-domain processing.</p> <p>alaSvcBVlan—The Backbone VLAN ID (BVLAN), which defines the B-Domain for the PBB traffic.</p> <p>alaSvcMulticastMode—The multicast replication mode for the service (Head End/Tandem).</p> | | | | |
| 168 | smgrSapStatusChange | alaSvcId, alaSapPortId, alaSapEncapValue, alaSapOperSta- tus | svcmgr | This trap is sent when there is a change in SAP operating status. A SAP is operationally up when it's admin-up and the link status of the physical or logical port of the SAP is operationally up. |
| <p>alaSvcId—The unique Service identifier.</p> <p>alaSapPortId—The ID of the access port where the SAP is defined.</p> <p>alaSapEncapValue—The value of the label used to identify the SAP on the access port.</p> <p>alaSapOperStatus—The SAP operational status on the port:</p> <ul style="list-style-type: none"> • up • down • ingressQosMismatch • egressQosMismatch • portMtuTooSmall • svcAdminDown • iesIfAdminDown | | | | |
| 169 | smgrSdpStatusChange | alaSdpId, alaSdpOperSta- tus, alaSdpNetwork- Port, alaSdpBVlan, alaSdpSys- temId, alaSdpSystem- Name, alaSdpDynam- icType, alaSdpIsid | svcmgr | This trap is sent when there is a change in SDP operating status. For SPB, the SDP is dynamically created or destroyed as calculated by ISIS protocol when a unicast/multicast path to reach a neighbor node is determined. |

| No. | Trap Name | Objects | Family | Description |
|-----|--|---|--------|---|
| | alaSdpId —The SDP identifier. | | | |
| | alaSdpOperStatus —The SDP operational status: | | | |
| | <ul style="list-style-type: none"> • up • notAlive • notReady • invalidEgressInterface • transportTunnelDown • down • created - dynamically created for SPB • destroyed - dynamically destroyed for SPB | | | |
| | Note: The value 'notAlive' is valid only when keep-alive is enabled, and it means that the keep-alive operating status is not alive. The value 'notReady' is valid only when this SDP uses a label signaling protocol (e.g. TLDP) and it means that the signaling session with the far-end peer has not been established. The value 'invalidEgressInterface' indicates that the IOM's have detected that the egress interface towards the far-end device is not a network port. | | | |
| | alaSdpNetworkPort —The network port where isis discovered the neighbor node information (BVLAN and BMAC). | | | |
| | alaSdpBVlan —The Backbone (BVLAN) where ISIS discovered the neighbor node information (BVLAN and BMAC). | | | |
| | alaSdpSystemId —The Backbone (BVLAN) where ISIS discovered the neighbor node information (BVLAN and BMAC). | | | |
| | alaSdpSystemName —The neighbor host name associated with the SDP object. | | | |
| | alaSdpDynamicType —SDP dynamically allocated for Unicast or Multicast Path. | | | |
| | alaSdpIsid —The I-Domain Service Identifier (I-SID) for the Group Mac assigned to this Multicast SDP. | | | |
| 170 | smgrSdpBindStatusChange | alaSvcId, alaSdpBindId, alaSdpBindOperStatus, alaSdpBindNetworkPort, alaSdpBindBVlan, alaSdpBindSystemId, alaSdpBindSystemName, alaSdpBindIsid | svcmgr | This trap is sent when there is a change in SDP Bind operating status. For SPB, the SDP Bind is dynamically created or destroyed as detected by ISIS when the same ISID is configured in the neighbor node. |

| No. | Trap Name | Objects | Family | Description |
|-----|--|--|---------|--|
| | <p>alaSvcId—The unique Service identifier. alaSdpBindId—The SDP Binding identifier. alaSdpBindOperStatus—The SDP Binding status:</p> <ul style="list-style-type: none"> • up • noEgressLabel • noIngressLabel • noLabels • down • svcMtuMismatch • sdpPathMtuTooSmall • sdpNotReady • sdpDown • sapDown • created - dynamically created for SPB • destroyed - dynamically destroyed for SPB <p>alaSdpBindNetworkPort—The network port associated with a service instance where MPLS labeled or B-Domain packets ingress and egress. alaSdpBindBVlan—The Backbone VLAN ID (BVLAN) associated with the SDP Binding object. alaSdpBindSystemId—The neighbor Backbone MAC (B-MAC) associated with the SDP Binding object. alaSdpBindSystemName,—The neighbor host name associated with the SDP Binding object. alaSdpBindIsid—The I-Domain Service Identifier (I-SID) for the Group MAC assigned to this Multicast SDP Bind.</p> | | | |
| 171 | alaPethPwrSupplyConflictTrap | pethPsePort-GroupIndex | module | This trap is sent when there is a power supply type conflict. |
| | pethPsePortGroupIndex —The slot of the involved object. | | | |
| 172 | alaPethPwrSupplyNotSupported | | module | This trap is sent when the power supply is not supported. |
| | pethPsePortGroupIndex —The slot of the involved object. pethPsePortIndex —The port of the involved object. | | | |
| 173 | chasTrapsBPSLessAllocSysPwr | physicalIndex, chasTrapsNiRqstdBpsSysPower, chasTrapsNiGrantdBpsSysPower | chassis | This trap is sent when there is insufficient system power provided by the BPS. |
| | physicalIndex —The physical index of the involved object. chasTrapsNiRqstdBpsSysPower —The requested system power from OS-BPS. chasTrapsNiGrantdBpsSysPower —The granted system power from OS-BPS. | | | |
| 174 | chasTrapsBPSSStateChange | chasTrapBPSSshelfId, chasTrapsBPSPowerSupply, chasTrapsBPSEventAlert | chassis | This trap is sent when BPS Power Supplies are inserted or removed. |
| | chasTrapBPSSshelfId —The BPS Shelf identifier. chasTrapsBPSPowerSupply —The BPS Power Supply involved in the trap. chasTrapsBPSEventAlert —An enumerated value identifying alert. | | | |

| No. | Trap Name | Objects | Family | Description |
|--|----------------------------------|--|-----------|---|
| 175 | chasTrapsNiBPSFETStateChange | physicalIndex, chasTrapsB- PSSystem- FETChange, chasTrapsBPS- PoeFETChan- ge | chassis | This trap is sent when there is a BPS FET change of state. |
| <p>physicalIndex—The physical index of the involved object. chasTrapsBPSSystemFETChange—An enumerated value indicating the BPS FET state. chasTrapsBSPoeFETChange—An enumerated value indicating the PoE FET state.</p> | | | | |
| 176 | alaDhcpBindingDuplicateEntry | dhcpSnooping- Binding- MacAddress, dhcpSnooping- BindingVlan, dhcpSnooping- BindingIfIn- dex, | ip-helper | This trap is sent when there is a MAC movement in the DHCP-Binding Table. |
| <p>dhcpSnoopingBindingMacAddress—The MAC Address sub-index identifying this instance. dhcpSnoopingBindingVlan—The DHCP client VLAN. dhcpSnoopingBindingIfIndex—The the interface from which the DHCP request is coming.</p> | | | | |
| 177 | alaVCSPProtectionTrap | alaVCSPTable- ChassisID | vcm | This trap is sent when a virtual chassis enters into VCSP Protection state |
| <p>alaVCSPTableChassisID—The virtual chassis ID (1 - 8).</p> | | | | |
| 178 | alaVCSRecoveryTrap | alaVCSPTable- ChassisID | vcm | This trap is sent when a virtual chassis enters into VCSP Active state |
| <p>alaVCSPTableChassisID—The virtual chassis ID (1 - 8).</p> | | | | |
| 179 | pethPsePortOnOffNotification | NA | module | This trap is sent to indicate whether or not the PSE Port is delivering power to the PD. This notification SHOULD be sent on every status change except in the searching mode. |
| <p>NA</p> | | | | |
| 180 | pethMainPowerUsageOnNotification | NA | module | This trap is sent to indicate that the PSE Threshold usage indication is on, the usage power is above the threshold. At least 500 msec must elapse between notifications being emitted by the same object instance. |
| <p>NA</p> | | | | |

| No. | Trap Name | Objects | Family | Description |
|--|-----------------------------------|---|---------|--|
| 181 | pethMainPowerUsageOffNotification | NA | module | This trap is sent to indicate that the PSE Threshold usage indication is off, the usage power is below the threshold. At least 500 msec must elapse between notifications being emitted by the same object instance. |
| NA | | | | |
| 182 | chasTrapsBPSFwUpgradeAlert | chasTrapBPSShelfId, chasTrapsBPSFwType, chasTrapsBPSFwVersion | chassis | This trap is sent when a BPS firmware upgrade is required. |
| <p>chasTrapBPSShelfId—The BPS Shelf identifier. chasTrapsBPSFwType—The BPS firmware type. chasTrapsBPSFwVersion—The BPS firmware version.</p> | | | | |
| 183 | alaAppMonAppRecordFileCreated | NA | app-mon | This trap is sent when the application records monitored in the past hour are written to the flash file. |
| NA | | | | |
| 184 | alaAppMonFlowRecordFileCreated | NA | app-mon | This trap is sent when a pre-configured number of flow records (configured by setting alaAppMonLoggingThresholdFlows) is written to the flash file. |
| NA | | | | |
| 185 | alaDPIFlowRecordFileCreated | NA | dpi | This trap is sent when a pre-configured number of flow records (configured by setting alaDPI-LoggingThresholdFlows) is written to the flash file. |

Index

Symbols

!! command 4-6

A

aaa authentication command 7-7, 7-8, 7-9, 8-4

aaa radius-server command 7-7

accounting

for Authenticated Switch Access 7-11

application examples

applying configuration files 5-4

Authenticated Switch Access 7-7

CMM 3-5

configuration file 5-2

dynamic link aggregation 13-5

Emergency Restore 3-16

file management 2-16

logging into the switch 1-3

network administrator user accounts 6-6

NTP 11-3

Server Load Balancing 10-27

SNMP 9-3

Trap Filters 9-4

WebView 8-4

applying configuration files

application examples 5-4

ASA

see Authenticated Switch Access

ASA Configuration

verify information about 7-12

Authenticated Switch Access 7-4

accounting 7-11

application examples 7-7

management interfaces 7-9

authentication

MD5 9-10

SHA 9-10

traps 9-13

Automatic Remote Configuration 12-5

Bootup Configuration File 12-12

Debug Configuration File 12-12

Firmware upgrade Files 12-12

Instruction File 12-12

Script File 12-12

Troubleshooting 12-21

Automatic Remote Configuration network components 12-6

TFTP File Server 12-6

B

banner

login 1-16

pre-login text 1-17

boot.cfg file 3-3

C

cd command 2-8

certified directory 3-4

copying to working directory 3-14

Chassis Management Module

see CMM

chmod command 2-10

CLI 4-1, 10-1

domains and families 6-15

logging commands 4-7-4-8

specifications 4-2, 10-2

CLI usage

verify information about 4-10

CMM 3-1

application examples 3-5

boot.cfg file 3-3

cancelling a reboot 3-9, 3-11

certified directory 3-4

checking reboot status 3-9

configuration files 3-3

copying

certified directory to working

directory 3-14

running configuration to working

directory 3-10

displaying current configuration 3-13

displaying switch files 3-13

image files 3-3

rebooting 3-8

rebooting from the working directory 3-11, 3-14

running configuration 3-4

scheduling a reboot 3-9

specifications 3-2

swapping primary for secondary 3-14

synchronizing primary and secondary 3-14

working directory 3-4

CMM Conditions

verify information about 3-16

CMM scenarios 3-5

lost running configuration 3-5

rollback to previous software 3-7

running configuration saved to working directory 3-6

working directory saved to certified directory 3-6

Command Line Interface

see CLI

community strings 9-9

configuration apply command 5-2, 5-4

for a specific timeperiod 5-5

configuration cancel command 5-7

configuration error-file limit command 5-7

configuration file

application examples 5-2

specifications 5-2

configuration files 3-3, 4-2
 errors 5-7
configuration snapshot all command 5-10
configuration syntax check 5-7
 console port 1-4

D

date 2-17, 5-4
 Daylight Savings Time
see DST
 defaults
 dynamic link aggregation 13-4
 login 1-2
 NTP 11-2
 SNMP 9-2
 startup 6-4
 switch security 7-2
 user accounts 6-2
 WebView 8-2
delete command 2-10
 DES encryption 9-10
 directories
 certified 3-4
 flash 2-7
 working 3-4
 DNS resolver 1-19
 Domain Name Server
see DNS resolver
 DST 2-18
 dynamic link aggregation
 application examples 13-5
 defaults 13-4
 specifications 13-3

E

editor
 vi 5-8
 Emergency Restore
 application examples 3-16
 encryption
 DES 9-10
 errors 5-7
exit command 2-15

F

File Configuration
 verify information about 5-12
 file management
 application examples 2-16
 specifications 2-2
 files
 attributes 2-10
 boot.cfg 3-3
 configuration 3-3
 image 3-3
 names 5-10
 permissions 2-10

 snapshots 5-9
 filters
 traps 9-4
freespace command 2-11
fsck command 2-11
 FTP client 2-14
ftp command 2-14
 FTP server 2-13

H

help 4-5
 HTTP
 web browser 1-5
http port command 8-3
http ssl command 8-3
https port command 8-3

I

image files 3-3
ip domain-lookup command 1-19
ip domain-name command 1-19
ip name-server command 1-19

K

keywords 4-4

L

lacp linkagg size command 13-20
 LDAP accounting servers
 Authenticated Switch Access 7-11
 LDAP servers
 for switch security 7-4
 logging into the switch
 application examples 1-3
 login
 defaults 1-2
 specifications 1-2
 login banner 1-16
 login settings
 verify information about 1-22
ls command 2-6, 4-5

M

Management Information Bases
see MIBs
 MD5
 authentication 9-10
 memory 2-11
mkdir command 2-8

N

network administrator user accounts
 application examples 6-6
 Network Management Station
see NMS

Network Time Protocol

see NTP

NMS 9-7

NTP 11-1

application examples 11-3

configuring 11-9

client 11-9

defaults 11-2

overview 11-5

specifications 11-2

stratum 11-6

using in a network 11-6

ntp broadcast command 11-9**ntp broadcast-delay** command 11-9

NTP client

broadcast delay 11-9

broadcast mode 11-9

ntp client command 11-3, 11-9

NTP Configuration

verify information about 11-13

ntp key command 11-12**ntp key load** command 11-12

NTP server

designating 11-10

minimum poll time 11-10

preferred server 11-11

Synchronization Tests 11-10

version number 11-11

ntp server command 11-3, 11-10**O**

OSPF

specifications 1-20

P

partition management 9-12

password command 6-8

passwords

expiration 6-11

global settings 6-7

user-configured 6-8

pre_banner.txt file 1-17

Prefix Recognition 4-6

primary CMM

swapping with the secondary 3-14

synchronizing with secondary 3-14

prompt 4-9

pwd command 2-7**R**

RADIUS accounting servers

Authenticated Switch Access 7-11

RADIUS servers

for switch security 7-4

RAM 3-3

reboot

cancelling 3-9, 3-11

checking status 3-9

primary 3-8

scheduling 3-9

working directory 3-11, 3-14

reload cancel command 3-11**reload** command 3-8, 3-9**reload working** command 3-11**rmdir** command 2-9

running configuration 3-4

copying to working directory 3-10

S

screen

display 4-9

prompt 4-9

secondary CMM

swapping with the primary 3-14

synchronizing with primary 3-14

Secure Shell 1-4, 1-11

algorithms 1-13

key exchange 1-13

security

SNMP 9-9

Server Load Balancing

application examples 10-27

session banner command 1-16**session login-attempt** command 1-18**session login-timeout** command 1-18**session prompt** command 4-9**session timeout** command 1-18**sftp** command 2-15

SHA

authentication 9-10

show command-log command 4-8**show command-log status** command 4-8**show configuration status** command 5-2, 5-7**show history** command 4-6**show ip helper** command 5-3**show microcode** command 3-13, 4-6**show ntp client** command 11-4**show ntp client server-list** command 11-3**show ntp server status** command 11-3**show reload** command 3-9**show running-directory** command 3-13**show snmp community map** command 9-9**show snmp mib family** command 9-14**show snmp station** command 9-3**show snmp trap replay** command 9-13**show user** command 6-6, 9-4, 9-10**show vlan svlan** command 13-23**show vlan svlan port-config** command 13-23

snapshots 5-9, 5-12

SNMP

access for user accounts 6-17

agent 9-6

application examples 9-3

defaults 9-2

management station 9-7

- manager 9-6
- security 9-9, 9-11
- specifications 9-2
- traps table B-2
- versions 9-7
- snmp community map mode** command 6-16
- SNMP configuration
 - verify information about 9-15
- snmp security** command 6-16, 9-11
- snmp trap filter** command 9-5
- software rollback
 - configuration scenarios 3-5
- specifications
 - CLI 4-2, 10-2
 - CMM 3-2
 - configuration file 5-2
 - dynamic link aggregation 13-3
 - file management 2-2
 - login 1-2
 - NTP 11-2
 - OSPF 1-20
 - SNMP 9-2
 - switch security 7-2
 - user database 6-2, 8-2, 10-25
- ssh** command 1-15
- SSL
 - see* Secure Socket Layer
- startup
 - defaults 6-4
- switch
 - rebooting 3-8
- switch security
 - defaults 7-2
 - specifications 7-2
- syntax 4-3, 10-15, 10-20
 - syntax checking 4-6
- System Clock 2-17
- system date** command 2-17
- system time** command 2-17
- system timezone** command 2-17

T

- Telnet 1-4, 1-10
- telnet** command 1-10
- time 2-17, 5-4
- time zone 2-17
- timed sessions 5-4
 - cancelling 5-7
 - future timed session 5-5
- Trap Filters
 - application examples 9-4
- Traps 9-12
- traps
 - authentication 9-13
 - families 9-12
 - filters 9-12
 - management 9-13
- tty** command 4-9

U

- user accounts
 - defaults 6-2
 - for switch access 6-4
 - saving settings 6-7
 - SNMP access 6-17
- user** command 6-12, 7-7, 9-4
 - creating a user 6-8
- user configuration
 - verify information about 6-18
- user database
 - specifications 6-2, 8-2, 10-25
 - switch management 7-4
- user password-expiration** command 6-11
- user password-size min** command 6-10
- users
 - see* user accounts
- UTC 11-1

V

- verbose mode 5-8
- vi** command 2-9

W

- WebView 8-1
 - application examples 8-4
 - browser setup 8-2
 - CLI commands 8-3
 - defaults 8-2
 - disabling 8-3
 - enabling 8-3
 - Secure Socket Layer 8-3
- who** command 6-18
- whoami** command 6-19
- working directory 3-4