**Part No. 060605-10, Rev. A**
**July 2019**

# OmniSwitch AOS Release 8 Switch Management Guide

## 8.6R1

**This user guide covers multiple OmniSwitch product lines and describes overall AOS feature configuration information. For platform specific feature support, please refer to the Specifications Guide and the Release Notes.**

Alcatel·Lucent Enterprise

**www.al-enterprise.com**

**This user guide documents AOS Release 8.6R1 for the**
**OmniSwitch 6465, OmniSwitch 6560, OmniSwitch 6860, OmniSwitch 6865,**
**OmniSwitch 6900, and OmniSwitch 9900.**
**The functionality described in this guide is subject to change without notice.**

The Alcatel-Lucent name and logo are trademarks of Nokia used under license by ALE. To view other trademarks used by affiliated companies of ALE Holding, visit: www.al-enterprise.com/en/legal/trademarks-copyright. All other trademarks are the property of their respective owners. The information presented is subject to change without notice. Neither ALE Holding nor any of its affiliates assumes any responsibility for inaccuracies contained herein.

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: businessportal2.alcatel-lucent.com
Email: ebg_global_supportcenter@al-enterprise.com

# Contents


**About This Guide** ..... ix
Supported Platforms ..... ix
Who Should Read this Manual? ..... ix
When Should I Read this Manual? ..... ix
What is in this Manual? ..... x
What is Not in this Manual? ..... x
How is the Information Organized? ..... x
Documentation Roadmap ..... xi
Related Documentation ..... xiii
Technical Support ..... xiv

**Chapter 1** **Getting Started and Upgrading AOS** ..... 1-1
In This Chapter ..... 1-1
Automatic Management Features ..... 1-2
Standalone or Virtual Chassis Mode ..... 1-4
Upgrading the Software ..... 1-5

**Chapter 2** **Logging Into the Switch** ..... 2-1
In This Chapter ..... 2-1
Login Defaults ..... 2-2
Quick Steps for Logging Into the Switch ..... 2-3
Overview of Switch Login Components ..... 2-4
Accessing the Micro-USB or RJ-45 Console Port ..... 2-6
Configuring a USB Adapter with Bluetooth Technology ..... 2-7
Setting the EMP Port's IP Address ..... 2-8
Using Telnet ..... 2-9
Using Secure Shell ..... 2-10
Modifying the Login Banner ..... 2-15
Configuring Login Parameters ..... 2-17
Configuring the Inactivity Timer ..... 2-17
Enabling the DNS Resolver ..... 2-18

Enabling the FIPS Mode ..... 2-18
Verifying Login Settings ..... 2-20

**Chapter 3 Managing System Files** ..... 3-1
In This Chapter ..... 3-1
Switch Administration Overview ..... 3-2
File and Directory Management ..... 3-4
Loading Software onto the Switch ..... 3-12
Code Guardian ..... 3-15
Installing Software Licenses ..... 3-16
Setting the System Clock ..... 3-17
Configuring Keychain Management ..... 3-20

**Chapter 4 Managing CMM Directory Content** ..... 4-1
In This Chapter ..... 4-1
CMM Files ..... 4-2
Managing Switch Configurations - Single CMM ..... 4-10
Managing CMM Redundancy ..... 4-16
Using the USB Flash Drive ..... 4-20
Checking the Integrity of the Image ..... 4-23
Displaying CMM Conditions ..... 4-24

**Chapter 5 Using the CLI** ..... 5-1
CLI Overview ..... 5-2
Command Entry Rules and Syntax ..... 5-3
Command Help ..... 5-5
Logging CLI Commands and Entry Results ..... 5-7
Customizing the Screen Display ..... 5-9
Verifying CLI Usage ..... 5-10

**Chapter 6 Working With Configuration Files** ..... 6-1
In This Chapter ..... 6-1
Tutorial for Creating a Configuration File ..... 6-2
Quick Steps for Applying Configuration Files ..... 6-3
Configuration Files Overview ..... 6-5
Creating Snapshot Configuration Files ..... 6-8
Verifying File Configuration ..... 6-11

**Chapter 7** **Managing Switch User Accounts** ..... 7-1
In This Chapter ..... 7-1
User Account Defaults ..... 7-2
Overview of User Accounts ..... 7-3
Creating a User ..... 7-7
Configuring Password Policy Settings ..... 7-9
Configuring Global User Lockout Settings ..... 7-12
Configuring Privileges for a User ..... 7-14
Setting Up SNMP Access for a User Account ..... 7-15
Multiple User Sessions ..... 7-17
Verifying the User Configuration ..... 7-19

**Chapter 8** **Managing Switch Security** ..... 8-1
In This Chapter ..... 8-1
Switch Security Defaults ..... 8-2
Switch Security Overview ..... 8-3
Authenticated Switch Access ..... 8-4
Configuring Authenticated Switch Access ..... 8-6
Quick Steps for Setting Up ASA ..... 8-7
Setting Up Management Interfaces for ASA ..... 8-9
Configuring Accounting for ASA ..... 8-11
Authenticated Switch Access - Enhanced Mode ..... 8-12
Joint Interoperability Test Command - JITC Mode ..... 8-18
Verifying the ASA Configuration ..... 8-20

**Chapter 9** **Using WebView** ..... 9-1
In This Chapter ..... 9-1
WebView CLI Defaults ..... 9-2
Browser Setup ..... 9-2
WebView CLI Commands ..... 9-3
Quick Steps for Setting Up WebView ..... 9-4
WebView Overview ..... 9-4

**Chapter 10** **Using SNMP** ..... 10-1
In This Chapter ..... 10-1
SNMP Defaults ..... 10-2
Quick Steps for Setting Up An SNMP Management Station ..... 10-3

Quick Steps for Setting Up Trap Filters ..... 10-4
SNMP Overview ..... 10-6
Using SNMP For Switch Security ..... 10-10
Configure SNMP Engine ID ..... 10-15
Working with SNMP Traps ..... 10-16
SNMP MIB Information ..... 10-18
Verifying the SNMP Configuration ..... 10-19

**Chapter 11 Using OmniVista Cirrus** ..... 11-1
In This Chapter ..... 11-1
OmniVista Cirrus Defaults ..... 11-2
Quick Steps for Configuring OmniVista Cirrus ..... 11-3
OmniVista Cirrus Overview ..... 11-5
Components of OmniVista Cirrus ..... 11-5
DHCP Server Option 43 ..... 11-8
Interaction with Other Features ..... 11-9
Dependencies ..... 11-9
OmniVista Cirrus Deployment Scenarios ..... 11-10
Verifying the OmniVista Cirrus Configuration ..... 11-10

**Chapter 12 Web Services, CLI Scripting, OpenFlow, and AOS Micro Services (AMS)** . 12-1
In This Chapter ..... 12-1
Web Services Overview ..... 12-2
Web Services REST Examples ..... 12-5
Using Python ..... 12-15
CLI Scripting ..... 12-20
Embedded Python Scripting ..... 12-25
OpenFlow Agent Overview ..... 12-27
Quick Steps to Configure OpenFlow Agent ..... 12-29
AOS Micro Services (AMS) ..... 12-30

**Chapter 13 Configuring Virtual Chassis** ..... 13-1
In This Chapter ..... 13-2
Virtual Chassis Default Values ..... 13-3
Quick Steps for Configuring A Virtual Chassis ..... 13-4
Virtual Chassis Overview ..... 13-6
Virtual Chassis Topologies ..... 13-14

Interaction with Other Features ..... 13-16
Configuring Virtual Chassis ..... 13-17
Virtual Chassis Configuration Example ..... 13-23
Automatically Setting up a Virtual Chassis ..... 13-28
Virtual Chassis Split Protection (VCSP) ..... 13-35
Displaying Virtual Chassis Configuration and Status ..... 13-38

**Chapter 14 Managing Automatic Remote Configuration Download** ..... 14-1
In This Chapter ..... 14-1
Automatic Remote Configuration Defaults ..... 14-2
Quick Steps for Automatic Remote Configuration ..... 14-4
Overview ..... 14-5
Interaction With Other Features ..... 14-8
Automatic Remote Configuration Download Process ..... 14-9
Download Component Files ..... 14-11
DHCP Client Auto-Configuration Process ..... 14-15
DHCP Server Preference ..... 14-16
Nearest-Edge Mode Operation ..... 14-17
LACP Auto Detection and Automatic Link Aggregate Association ..... 14-19
Troubleshooting ..... 14-20

**Chapter 15 Configuring Automatic Fabric** ..... 15-1
In This Chapter ..... 15-2
Automatic Fabric Default Values ..... 15-3
Quick Steps for Configuring Automatic Fabric ..... 15-4
Automatic Fabric Overview ..... 15-7
Automatic Fabric Discovery Examples ..... 15-17
Interaction with Other Features ..... 15-21
Configuring Automatic Fabric ..... 15-25
Displaying the Automatic Fabric Configuration ..... 15-29

**Chapter 16 Configuring Network Time Protocol (NTP)** ..... 16-1
In This Chapter ..... 16-1
NTP Defaults Table ..... 16-1
NTP Quick Steps ..... 16-3
NTP Overview ..... 16-5
Configuring NTP ..... 16-9

Verifying NTP Configuration ..... 16-13

**Appendix A** **Software License and Copyright Statements** ..... A-1

ALE USA, Inc. License Agreement ..... A-1

Third Party Licenses and Notices ..... A-4

**Appendix B** **SNMP Trap Information** ..... B-1

SNMP Traps Table ..... B-2

MIBS Table ..... B-67

System Events ..... B-73

**Index** ..... Index-1

# About This Guide

This *OmniSwitch AOS Release 8 Switch Management Guide* describes basic attributes of your switch and basic switch administration tasks. The software features described in this manual are shipped standard with your switches. These features are used when readying a switch for integration into a live network environment.

## Supported Platforms

The information in this guide applies only to the following products:

- OmniSwitch 6465 Series
- OmniSwitch 6560 Series
- OmniSwitch 6860 Series
- OmniSwitch 6865 Series
- OmniSwitch 6900 Series
- OmniSwitch 9900 Series

## 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 appropriate Hardware Users 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 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 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 provide specifications, illustrations, and descriptions of all hardware components, such as chassis, power supplies, Chassis Management Modules (CMMs), Network Interface (NI) modules, and cooling fans. It also includes steps for common procedures, such as removing and installing switch components.

The *OmniSwitch AOS Release 8 Switch Management 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*
*OmniSwitch AOS Release 8 Data Center Switching 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).

The *OmniSwitch AOS Release 8 Data Center Switching Guide* includes configuration information for data center networks using virtualization technologies, such as Data Center Bridging (DCB) protocols, Virtual eXtensible LAN (VxLAN), and Fibre Channel over Ethernet (FCoE) network convergence.

## 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 6465, 6560, 6860, 6865, 6900, 9900 Hardware Users Guides*

  Describes the hardware and software procedures for getting an OmniSwitch up and running as well as complete technical specifications and procedures for all OmniSwitch chassis, power supplies, fans, and Network Interface (NI) modules.

- *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 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 Data Center Switching Guide*

  Includes and introduction to the OmniSwitch data center switching architecture as well as network configuration procedures and descriptive information on all the software features and protocols that support this architecture. Chapters cover Data Center Bridging (DCB) protocols, Virtual Network Profile (vNP), VxLAN and FCoE transit and gateway functionality.

- *OmniSwitch AOS Release 8 Transceivers Guide*

  Includes SFP and XFP transceiver specifications and product compatibility information.

- *OmniSwitch AOS Release 8 Specifications Guide*

  Includes Specifications table information for the features documented in the Switch Management Guide, Network Configuration Guide, Advanced Routing Guide, and Data Center Switching Guide.

- Technical Tips, Field Notices

  Includes information published by Alcatel-Lucent Enterprise'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 Enterprise 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 Enterprise 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 Enterprise'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 Enterprise's technical support, open a new case or access helpful release notes, technical bulletins, and manuals.

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

Web: businessportal2.alcatel-lucent.com

Phone: 1-800-995-2696

Email: ebg_global_supportcenter@al-enterprise.com

# 1 Getting Started and Upgrading AOS

This chapter provides an overview of what to expect when first bringing up an OmniSwitch. It describes the Automatic Management features an OmniSwitch runs when booting for the first time as well as whether a switch will come up in standalone or VC mode. This chapter is also helpful for getting started with a new AOS release by covering important information related to upgrading the switch.

## In This Chapter

Configuration procedures described in this chapter include:


- "Automatic Management Features" on page 1-2
- "Standalone or Virtual Chassis Mode" on page 1-4
- "Upgrading the Software" on page 1-5

# Automatic Management Features

All switches that ship from the factory will default to Virtual Chassis mode and attempt to run the automatic VC, automatic remote configuration, and automatic fabric protocols. The automatic features can be disabled during the switch reboot or after the switch has finished booting if desired.

When a switch boots with no configuration file or with a configuration file with a size of 0 bytes, the following boot processes occur:

**1 Automatic Virtual Chassis** - The switch will run the automatic VC protocol and try to automatically configure the VFLs and setup a VC. The time to complete this process will vary depending on the configuration. Please see Chapter 13, "Configuring Virtual Chassis" for additional information.

**2 Automatic Remote Configuration** - Once the automatic VC process is complete, the automatic remote configuration process will begin. It can take approximately 180 seconds for this process to complete if there is no remote configuration server available. Please see Chapter 14, "Managing Automatic Remote Configuration Download" for additional information.

---

**Note.** The automatic remote configuration download process can be aborted at any time by entering **auto-config-abort** command, for example:

```
-> auto-config-abort
```

---

**3 Automatic Fabric** - Once the automatic remote configuration process completes, the automatic fabric process will begin. Please see Chapter 15, "Configuring Automatic Fabric" for additional information.

---

**Note.** The automatic fabric process can be disabled at any time by entering **auto-fabric admin-state** command, for example:

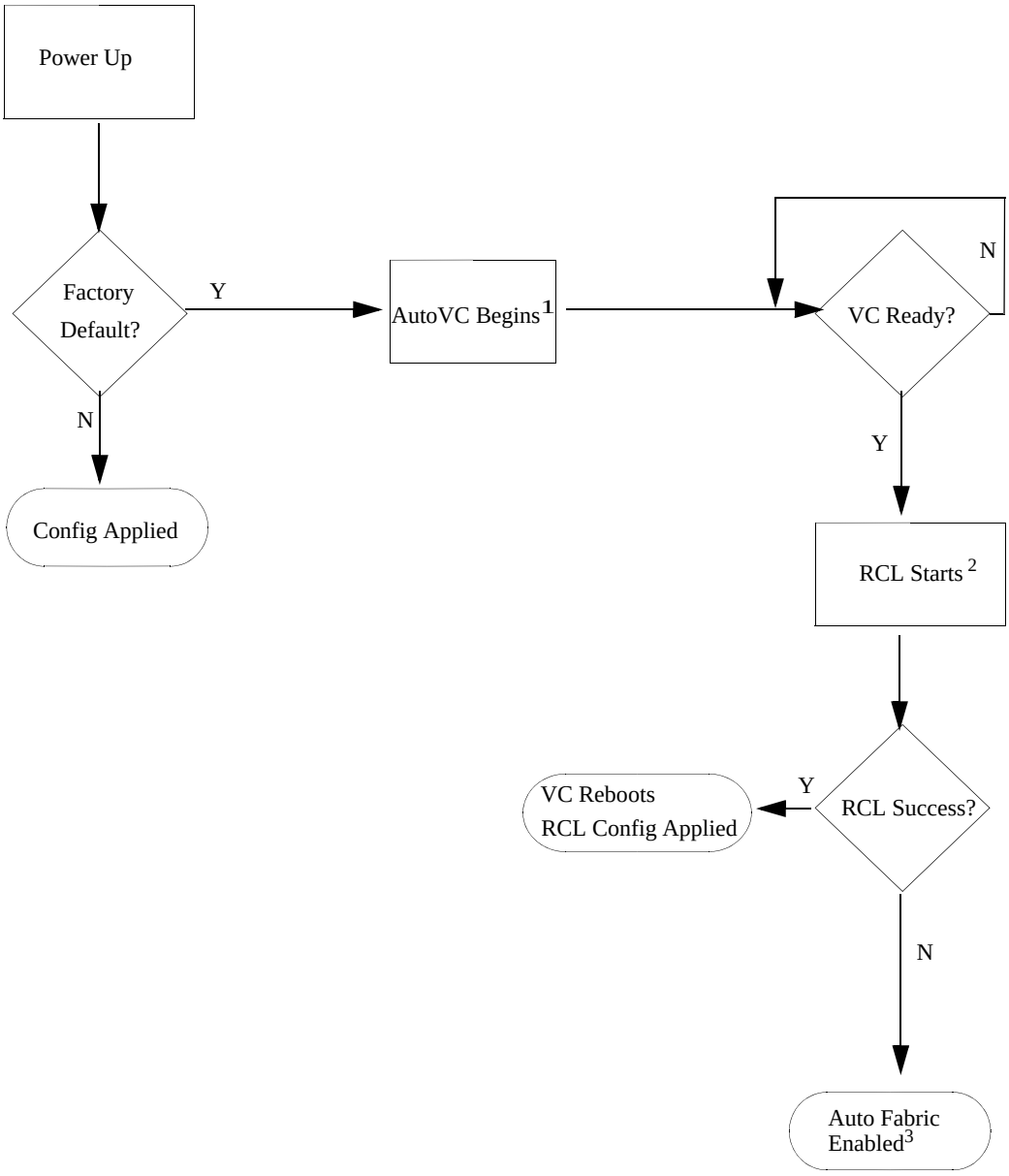
```
-> auto-fabric admin-state disable
```

---

## Automatic Management Feature Guidelines

- This boot process only applies to switches that boot without a configuration file, such as newly shipped switches from the factory.
- The automatic features can be disabled at the start of the switch boot process by pressing '**y**' when prompted. The switch will boot into standalone mode with all automatic features disabled. Please see "Standalone or Virtual Chassis Mode" on page 1-4 for additional information.
- To prevent a switch from re-running the automatic fabric process upon the next reboot enter **write memory** to save the configuration to the configuration file.
- A message similar to the one below may be seen during the Automatic Remote Configuration process. This is normal as the switch attempts the process between VLANs 1 and 127.

```
Wed Mar 18 19:16:12 : ipv4 vlan warning message:
+++ vm vlan dereg fail 117 (127)
```




**Automatic Management Features Flow Overview**

1. See Chapter 13, “Configuring Virtual Chassis” for additional information on Auto VC.

2. See Chapter 14, “Managing Automatic Remote Configuration Download” for additional information on Automatic Remote Configuration Download.

3. See Chapter 15, “Configuring Automatic Fabric” for additional information on Automatic Fabric.

# Standalone or Virtual Chassis Mode

When a chassis boots with its default factory configuration it will run in VC mode. There may be times when standalone mode is preferred such as when introducing the chassis into an already existing network.

There are multiple ways to have the switch come up in standalone mode instead of VC mode.

## Automatic Management features disabled during switch boot

If the automatic management features were disabled while the switch was booting by pressing '**y**' at the prompt, the switch will boot into standalone mode.

```
Do you want to disable auto-configurations on this switch [Y/N]? y
Auto-configurations disabled
```

The switch automatically creates a configuration file so that it will no longer run the automatic protocols upon boot up.

## Automatic Management features not disabled during switch boot

If the automatic management features were not disabled while the switch was booting issue the **auto-fabric admin-state** command with the **remove-vc-reload** parameter. This will do the following:

1. Clear any automatic fabric configuration
2. Disable the automatic fabric features
3. Create a *vcboot.cfg* file in the **/flash/working** directory
4. Reload the switch
5. Since the switch will reboot with a configuration file the automatic management features will no longer run.

# Upgrading the Software

This section is to assist with upgrading an OmniSwitch. The goal is to provide a clear understanding of the basic steps and types of upgrade processes available for an OmniSwitch. Depending upon the AOS version, model, and configuration of the OmniSwitch various upgrade procedures are supported. This section provides an overview. Since each AOS release has different upgrade requirements please refer to the Release Notes for step-by-step instructions.

**Standard Upgrade**—The standard upgrade of a standalone chassis or virtual chassis (VC) is nearly identical. All that's required is to upload the new image files to the Running directory and reload the switch. In the case of a VC, prior to rebooting the Master will copy the new image files to the Slave and once the VC is back up the entire VC will be synchronized and running with the upgraded code.

**ISSU**—The In Service Software Upgrade (ISSU) is used to upgrade the software on a VC or modular chassis with minimal network disruption. Each element is upgraded individually allowing hosts and switches which are dual-homed to maintain connectivity to the network. The actual downtime experienced by a host on the network can vary depending upon the overall network design and configuration. Having a redundant configuration is suggested and will help to minimize recovery times.

- **Virtual Chassi**s—The VC will first verify that it is in a state that will allow a successful ISSU upgrade. It will then copy the image and configuration files of the ISSU specified directory to all of the Slave chassis and reload each Slave chassis from the ISSU directory in order from lowest to highest chassis-id. For example, assuming chassis-id 1 is the Master, the Slave with chassis-id 2 will reload with the new image files. When Slave chassis-id 2 has rebooted and rejoined the VC, the Slave with chassis-id 3 will reboot and rejoin the VC. Once the Slaves are complete they are now using the new image files. The Master chassis is now rebooted which causes the Slave chassis to become the new Master chassis. 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 takeover, re-assuming the Master role.
- **Modular Chassi**s—The chassis will first verify that it is in a state that will allow a successful upgrade. It will then copy the image and configuration files of the specified directory to the secondary CMM and reload the secondary CMM which becomes the new primary CMM. The old primary CMM becomes the secondary CMM and reloads using the upgraded code. As a result of this process both CMMs are now running with the upgraded code and the primary and secondary CMMs will have changed roles (i.e., primary will act as secondary and the secondary as primary). The individual NIs can be reset either manually or automatically.

## Prerequisites

Before upgrading, the individual performing the upgrade must:

- Read the release notes for the appropriate AOS release.
- Be the responsible party for maintaining the switch's configuration
- Be aware of any issues that may arise from a network outage caused by improperly loading this code
- Understand that the switch must be rebooted and network access may be affected by following this procedure.
- Have a working knowledge of the switch to configure it to accept an FTP connection through the EMP or Network Interface (NI) Ethernet port
- Read the GA Release Notes prior to performing any upgrade for information specific to this release.
- Ensure there is a current certified configuration on the switch so that the upgrade can be rolled-back if required.
- Verify the current versions of Uboot and FPGA. If they meet the minimum requirements, (i.e. they were already upgraded during a previous AOS upgrade) then only an upgrade of the AOS images is required.
- Depending on whether a standalone chassis or VC is being upgraded, upgrading can take from 5 to 20 minutes. Additional time will be needed for the network to re-converge.
- The examples below use various models and directories to demonstrate the upgrade procedure. However any user-defined directory can be used for the upgrade.
- If possible, have EMP or serial console access to all chassis during the upgrade. This will allow monitoring of the VC during the ISSU process and before the virtual chassis has been re-established.

Knowledge of various aspects of AOS directory structure, operation, and CLI commands can be found in the Alcatel-Lucent Enterprise OmniSwitch User Guides. Recommended reading from the Switch Management Guide includes the following chapters:

- Chapter 1, “Getting Started and Upgrading AOS”
- Chapter 2, “Logging Into the Switch”
- Chapter 3, “Managing System Files”
- Chapter 4, “Managing CMM Directory Content”
- Chapter 5, “Using the CLI”
- Chapter 6, “Working With Configuration Files”
- Chapter 13, “Configuring Virtual Chassis”
- Release Notes - for the version of software you're planning to upgrade to.

## Switch Maintenance

It's recommended to perform switch maintenance prior to performing any upgrade. This can help with preparing for the upgrade and removing unnecessary files. The following steps can be performed at any time prior to a software upgrade. These procedures can be done using Telnet and FTP, however using SSH and SFTP/SCP are recommended as a security best-practice since Telnet and FTP are not secure.

**1** Verify current date, time, AOS and model of the switch:

```
-> show system
```

**2** Remove any old tech_support.log files, tech_support_eng.tar files:

```
-> rm *.log
-> rm *.tar
```

**3** Verify that the /pmd and /pmd/work directories are empty. If they have files in them check the date on the files. If they are recently created files (<10 days), contact Alcatel-Lucent Enterprise Service & Support. If not, they can be deleted.

```
-> rm /flash/pmd/*.*
-> rm /flash/pmd/work/*.*
```

**4** Use the 'show running-directory' command to determine what directory the switch is running from and that the configuration is certified and synchronized. If the configuration is not certified and synchronized,issue the command 'write memory flash-synchro’.

```
-> show running-directory
```

**5** If you do not already have established baselines to determine the health of the switch you are upgrading, now would be a good time to collect them. Using the show tech-support series of commands is an excellent way to collect data on the state of the switch. The show tech support commands automatically create log files of useful show commands in the /flash directory. You can create the tech-support log files with the following commands:

```
-> show tech-support
-> show tech-support layer2
-> show tech-support layer3
```

**6** Additionally, the following command will create a TAR file with multiple tech-support log files as well as the SWLOG files from the switches.

```
-> show tech-support eng complete
```

## Standard Upgrade

This section describes the basic steps for upgrading an OmniSwitch standalone or virtual chassis using a standard upgrade. This section is an overview. For specific step-by-step instructions please refer to the Release Notes.

1. Follow the instructions in the "Switch Maintenance" on page 1-7 section.
2. Download the upgrade files from the Service & Support website.
3. FTP the upgrade files to the *RUNNING* directory of the switch.
4. Upgrade the image files by reloading the switch from the *RUNNING* directory.
5. After the switch reboots, verify the software upgrade.
6. Certify the upgrade.

## In-Service Software Upgrade (ISSU)

This section describes the basic steps for upgrading an OmniSwitch standalone or virtual chassis using ISSU. This section is an overview. For specific step-by-step instructions please refer to the Release Notes.

1. Follow the instructions in the "Switch Maintenance" on page 1-7 section.
2. Download the upgrade files.
3. Create the new directory on the Master/Primary CMM for the ISSU upgrade.
4. Clean up any existing ISSU directories.
5. On the Master chassis / Primary CMM copy the current Running configuration files to the ISSU directory.
6. FTP the new image files and the validation file to the ISSU directory.
7. Upgrade the image files using ISSU.
8. Verify the software upgrade.
9. Certify the software upgrade.
10. Reset NIs (modular chassis)

### The Validation File

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

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

**Resetting NIs - Modular Chassis**

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

# 2 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 Enterprise'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 2-3
- "Accessing the Micro-USB or RJ-45 Console Port" on page 2-6
- "Configuring a USB Adapter with Bluetooth Technology" on page 2-7
- "Setting the EMP Port's IP Address" on page 2-8
- "Using Telnet" on page 2-9
- "Using Secure Shell" on page 2-10
- "Using Secure Shell" on page 2-10
- "Modifying the Login Banner" on page 2-15
- "Configuring Login Parameters" on page 2-17
- "Enabling the DNS Resolver" on page 2-18
- "Enabling the FIPS Mode" on page 2-18


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 8, "Managing Switch Security" |
| Authenticating users to manage the switch | Chapter 8, "Managing Switch Security" |

| For more information about... | See... |
|---|---|
| Creating user accounts directly on the switch | Chapter 7, “Managing Switch User Accounts” |
| Using the CLI | Chapter 5, “Using the CLI” |
| Using WebView to manage the switch | Chapter 9, “Using WebView” |
| Using SNMP to manage the switch | Chapter 10, “Using SNMP” |

# Login Defaults

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

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

# Quick Steps for Logging Into the Switch

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

- Connecting to the switch via the console port.
- Setting up the Ethernet Management Port (EMP).
- Enabling (or "unlocking") management interfaces types through the **aaa authentication** command for the interface you are using. For detailed information about enabling session types, see Chapter 8, "Managing Switch Security."

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

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

```
Welcome to the Alcatel-Lucent Enterprise OS6900 8.3.1.313.R01 GA, August 31, 2016.

Copyright (c) 1994-2014 Alcatel-Lucent.  All Rights Reserved.
Copyright (c) 2014-2016 Alcatel-Lucent Enterprise.  All Rights Reserved.

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

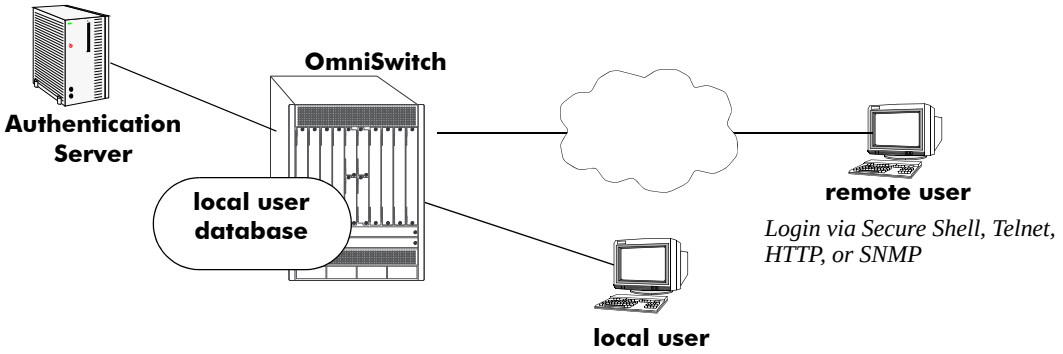
You are now logged into the CLI. For information about changing the welcome banner, see "Modifying the Login Banner" on page 2-15.

For information about changing the login prompt, see Chapter 5, "Using the CLI."

For information about setting up additional user accounts locally on the switch, see Chapter 7, "Managing Switch User Accounts."

# Overview of Switch Login Components

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




**Switch Login Components**

## Management Interfaces

Logging into the switch may be done locally or remotely. Remote connections may be secure or insecure, depending on the method. Management interfaces are enabled using the **aaa authentication** command. This command also requires specifying the external servers and/or local user database that will be used to authenticate users. The process of authenticating users to manage the switch is called Authenticated Switch Access (ASA). Authenticated Switch Access is described in detail in Chapter 8, "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 or RJ-45 Console Port" on page 2-6.
- **USB Adapter with Bluetooth Technology**—A direct connection to the switch using a USB adapter with Bluetooth technology. The console port is always enabled for the default user account, see "Configuring a USB Adapter with Bluetooth Technology" on page 2-7.
- **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 2-8
- **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 2-9.
- **Secure Shell**—Any standard Secure Shell client may be used for remote login to the switch. See "Using Secure Shell" on page 2-10.

### 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 9, "Using WebView."

### Using SNMP to Manage the Switch

**SNMP**—Any standard SNMP application may be used for configuring the switch. See Chapter 10, "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 7, "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 or RJ-45 Console Port

## Micro-USB

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:

**www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers**

**2** Connect the OmniSwitch to the USB port of your device using the a 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 welcome banner of the switch is displayed, followed by the CLI prompt.

```
Welcome to the Alcatel-Lucent Enterprise OS6860 8.3.1.313.R01 GA, August 31, 2016.

Copyright (c) 1994-2014 Alcatel-Lucent.  All Rights Reserved.
Copyright (c) 2014-2016 Alcatel-Lucent Enterprise.  All Rights Reserved.

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

## RJ45-to-DB9

The following procedure is used for accessing the switch using the RJ-45 console connection.

**1** Connect the OmniSwitch to the serial port of your device using an RJ-45 cable and an RJ-45-to-DB9 connector.

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

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.

## Console Port Parameters

If you wish to modify the default serial connection settings (i.e., baud rate, parity, data bits, stop bits, and mode), use the **modify boot parameters** command as shown:

```
-> modify boot parameters

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

Boot > show
```

```
Serial (console) baud: 19200
Serial (console) parity: even
Serial (console) wordsize: 7
Serial (console) stopbits: 2
Serial (console) mode: modemControlOn

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

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

# Configuring a USB Adapter with Bluetooth Technology

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

**1** Enable access for a USB adapter with Bluetooth technology on the OmniSwitch using the **bluetooth** command.

**2** Insert the USB adapter with Bluetooth technology 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 USB adapter with Bluetooth technology 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 Enterprise OS6860 8.3.1.313.R01 GA, August 31, 2016.

Copyright (c) 1994-2014 Alcatel-Lucent.  All Rights Reserved.
Copyright (c) 2014-2016 Alcatel-Lucent Enterprise.  All Rights Reserved.

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

## Identifying the Current Connection

When configuring multiple switches using a USB adapter with Bluetooth technology it may be difficult to determine which switch has the active 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 shared between both CMMs, stored in the **vcboot.cfg** file.
- The Primary or Secondary's CMM's IP address, stored in NVRAM. (Not required for remote access)

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

- All the EMP IP addresses and CMM's IP addreses must be in the same subnet.
- Each of the IP addresses must be unique.
- There is no dedicated routing table for the EMP interface. All management interfaces use the same routing table with EMP and non-EMP routes.
- Changes stored in NVRAM will remain with the CMM if the CMM is moved to a different chassis.

## Modifying the Shared EMP IP Address

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

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

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

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

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

```
-> modify boot parameters

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

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

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

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

# Using Telnet

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

## Logging Into the Switch Through Telnet

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

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

## Starting a Telnet Session from the Switch

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

The following shows an example of telnetting to another OmniSwitch:

```
-> telnet 198.51.100.100
Trying 198.51.100.100...
Connected to 198.51.100.100
Escape character is '^]'.
login : admin
password :
Welcome to the Alcatel-Lucent Enterprise OS6900-X40 8.3.1.313.R01 GA, August 31,
2016.

Copyright (c) 1994-2014 Alcatel-Lucent.  All Rights Reserved.
Copyright (c) 2014-2016 Alcatel-Lucent Enterprise.  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

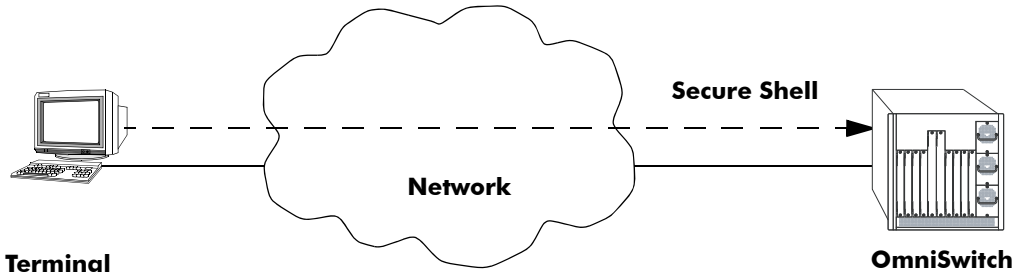
Secure Shell FTP is the standard file transfer protocol used with Secure Shell. Secure Shell FTP is an interactive file transfer program (similar to the industry standard FTP) which performs all file transfer operations over a Secure Shell connection.

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 2-14 for detailed information.

# Secure Shell Application Overview

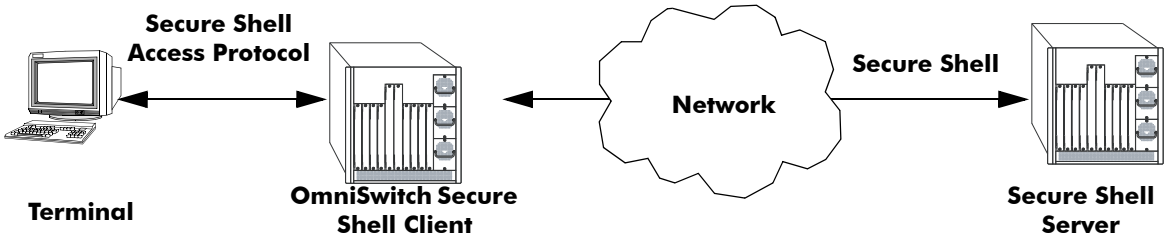
Secure Shell is an access protocol used to establish secured access to your OmniSwitch. The Secure Shell protocol can be used to manage an OmniSwitch directly or it can provide a secure mechanism for managing network servers through the OmniSwitch.

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<br>diffie-hellman-group1-sha1 |
| Key Location | /flash/system |
| Key File Names | Public<br>- ssh_host_key.pub, ssh_host_dsa_key.pub, ssh_host_rsa_key.pub<br>Private<br>- 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 (remote_ssh_user@device) and the key must be in the following format:

**<ssh-rsa | ssh-dsa> <encrypted key> <remote_ssh_user@device>**

---

**Example Key:**

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

**1** Use the ssh-keygen utility of the OpenSSH software suite to generate a private and public key pair as shown 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 filename 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 to which you are connecting.

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 Enterprise OS6900-X40 8.3.1.313.R01 GA, August 31,
2016.

Copyright (c) 1994-2014 Alcatel-Lucent.  All Rights Reserved.
Copyright (c) 2014-2016 Alcatel-Lucent Enterprise.  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 Enterprise OS6900-X40 8.3.1.313.R01 GA, August 31,
2016.

Copyright (c) 1994-2014 Alcatel-Lucent.  All Rights Reserved.
Copyright (c) 2014-2016 Alcatel-Lucent Enterprise.  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 Enterprise OS6900-X40 8.3.1.313.R01 GA, August 31,
2016.

Copyright (c) 1994-2014 Alcatel-Lucent.  All Rights Reserved.
Copyright (c) 2014-2016 Alcatel-Lucent Enterprise.  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.

---

**Note.** The banner text files located in the /**flash/switch** directory are not synchronized across CMMs when using the '**copy running certified**' command, they must manually copied to each CMM.

---

# 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.

You can configure the session time-out for incomplete or broken SSH session using the **ssh login-grace-time** command as follows:

```
-> ssh login-grace-time 200
```

In this example, the incomplete or broken SSH session will time-out after 200 seconds. This means the user must establish a SSH session within 200 seconds.

The time-out period can be configured between 30 seconds to 600 seconds. By default the time-out period is set to 120 seconds.

To view the configured time-out period use the **show ssh** command.

# Configuring the Inactivity Timer

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

For example:

```
-> session cli timeout 8
-> session ftp timeout 5
-> session http timeout 10
```

# Enabling the DNS Resolver

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

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

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

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

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

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

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

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

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.

## 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.

---

**Note.** A 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.

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. |

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

# 3 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 3-2
- "File and Directory Management" on page 3-4
- "Loading Software onto the Switch" on page 3-12
- "Code Guardian" on page 3-15
- "Installing Software Licenses" on page 3-16
- "Setting the System Clock" on page 3-17
- "Configuring Keychain Management" on page 3-20


For related information about connecting a terminal to the switch, see your *Hardware Users Guide*. For information about switch command privileges, see Chapter 8, "Managing Switch Security."

# 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 Enterprise. If you change your configuration to upgrade your network, you must understand how to install switch files and to manage switch directories.

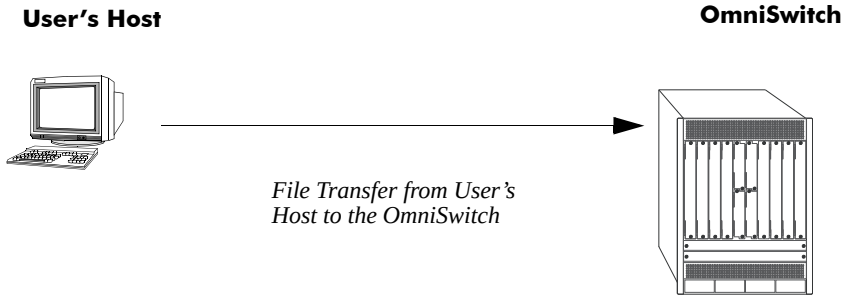
The OmniSwitch Series uses flash memory store files, including executable files (used to operate switch features and applications), configuration files, and log files.

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.

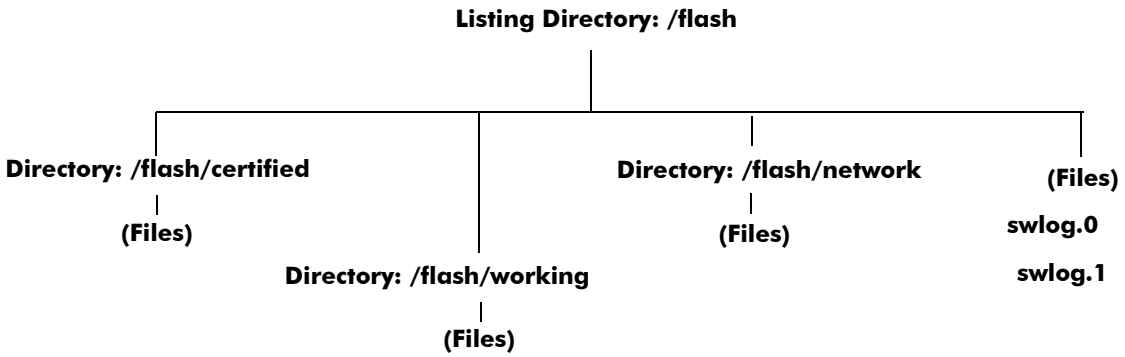



**File Transfer to OmniSwitch**

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**

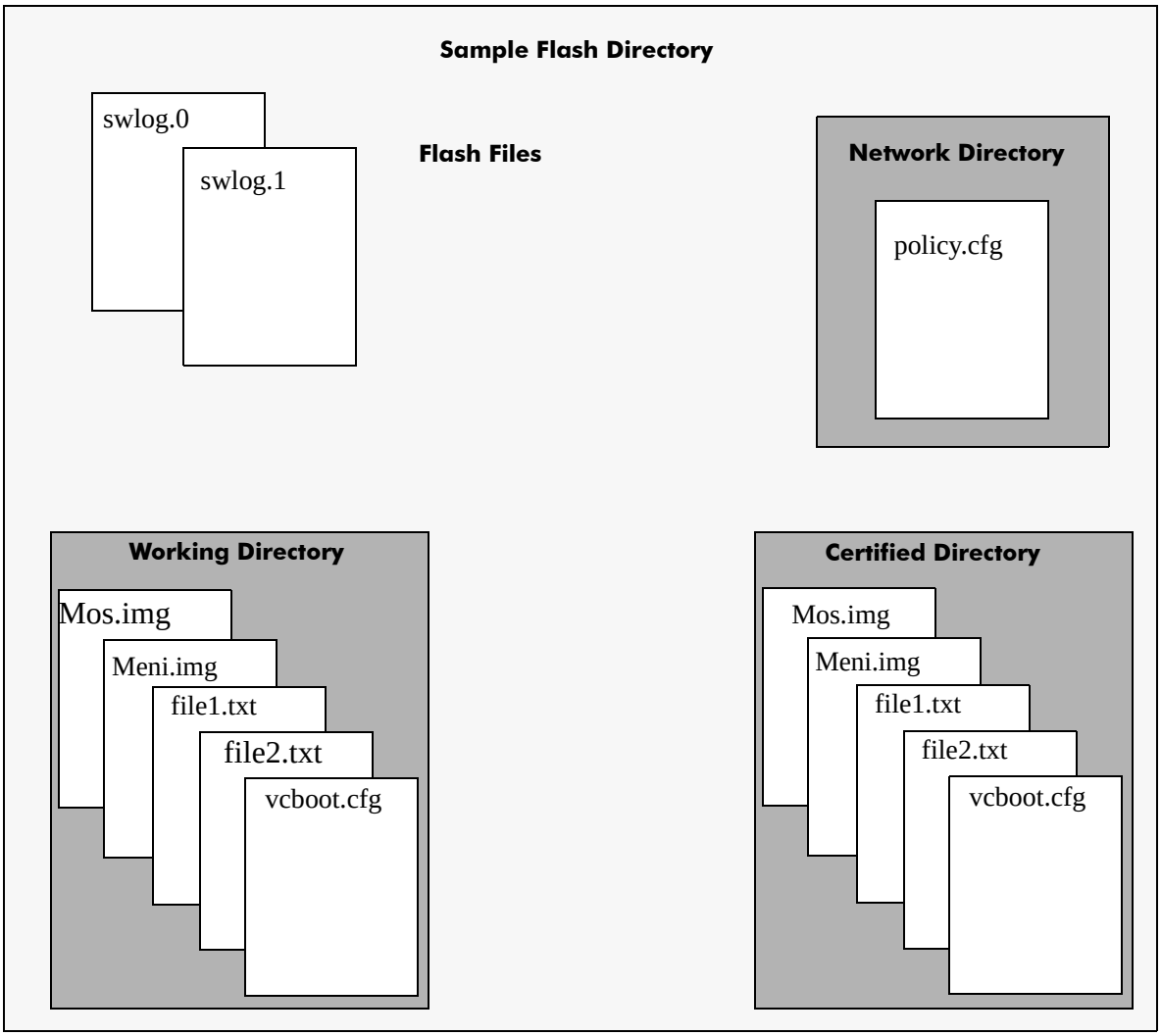
# File and Directory Management

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.



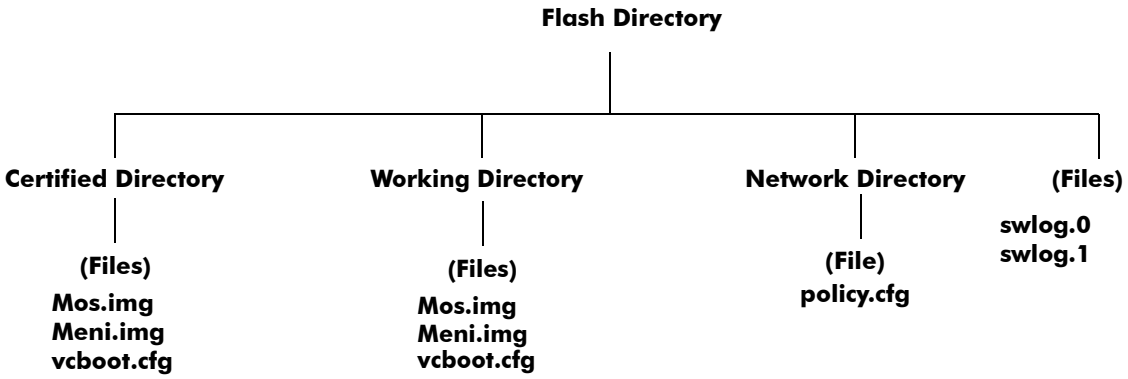
To list all the files and directories in the current directory, use the **ls** command. Here is a sample display of the flash directory.

```
-> ls -l
-rw-r--r--    1 root     0              342 Aug 30 18:28 vcboot.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 3-4.




**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.

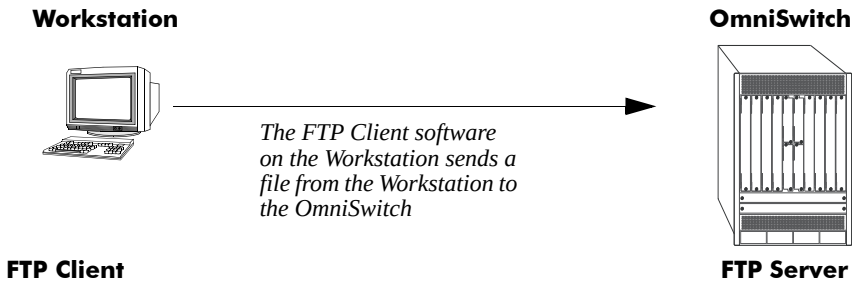
# Loading Software onto the Switch

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.




**OmniSwitch as a Server**

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

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

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

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

## Using the Switch as an FTP Client

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

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

**1** Establish a connection to the switch as explained in your appropriate *Hardware Users Guide.*

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

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

---

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

---

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

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

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

```
ftp->?

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

## Using Secure Shell FTP

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

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

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

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

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

## Closing a Secure Shell FTP Session

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

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

## Using TFTP to Transfer Files

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

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

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

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

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

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

---

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

---

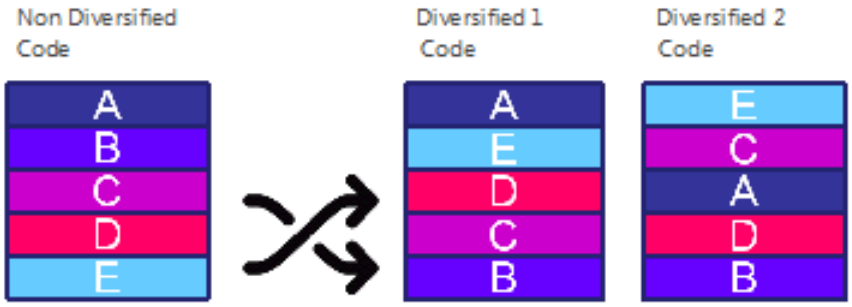
# Code Guardian

Alcatel-Lucent Enterprise and LGS Innovations have combined to provide the first network equipment to be hardened by an independent group. CodeGuardian promotes security and assurance at the network device level using independent verification and validation of source code, software diversification to prevent exploitation and secure delivery of software to customers.

CodeGuardian employs multiple techniques to identify vulnerabilities such as software architecture reviews, source code analysis (using both manual techniques and automated tools), vulnerability scanning tools and techniques, as well as analysis of known vulnerabilities in third party code.

**Software diversification**

Software diversification randomizes the executable program so that various instances of the same software, while functionally identical, are arranged differently. The CodeGuardian solution rearranges internal software while maintaining the same functionality and performance and modifies the deliverable application to limit or prevent/impede software exploitation.




| Chassis | Standard AOS Release | AOS CodeGuardian Release | LGS AOS CodeGuardian Release |
|---|---|---|---|
| OmniSwitch | 8.6R1 | 8.6RX | 8.6LX |

**CodeGuardian AOS Releases**

- X=Diversified image 1-3
- ALE will have 3different diversified images per AOS release (R11 through R31)
- Our partner LGS will have 3 different diversified images per AOS release (L11 through L31)

# 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 Enterprise 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://businessportal2.alcatel-lucent.com and provide the customer number, order number, activation code along with serial number and MAC address of the switch. 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.

- The MACSec license is a site license and uses an order ID instead of a serial number and MAC address.

- For MACsSec and 10G license, a reboot is not required.

---

# 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 3-18 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 3-18 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 |

**Time Zone and DST Information Table (continued)**

| Abbreviation | Name | Hours from UTC | DST Start | DST End | DST Change |
|---|---|---|---|---|---|
| **zm3** | No standard name | -03:00 | No default | No default | No default |
| **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 |

# Configuring Keychain Management

The keychain module is a centralized key management mechanism in AOS. Any module using key management service ensures enhanced security with regular rotation of the keys. Each keychain defines set of keys with start time and end time.

To configure a key chain, an administrator defines a series of keys, and the router software rotates through them. Each key also has an associated time interval, or 'lifetime' for which it will be activated. The authentication code included in each key is called the key string.

When a user application (like OSPF, ISIS) receives a packet, it has to be authenticated as per the authentication type, key, and message digest. When a keychain is associated with a user application, hello packets are authenticated using key provided by the keychain module.

The authentication is passed when following conditions are satisfied. Else, the adjacency is not formed and hello packet is discarded.

- Current active key defined in keychain and key in the packet are same.
- Authentication type of the current key in the keychain and the authentication type mentioned in the packet are same.
- Message digest calculated by the keychain manager based on the active key and message digest carried in the packet are same.

## Generating Random Key

Use **security key-chain gen-random-key** command to generate a 32-byte random key. The generated key can be used as an input value for hex-key or encrypt-key while creating a security key.

```
-> security key-chain gen-random-key
0x0102030405060708090A0B0C0D0E0F
```

## Configuring a Key

Configure a key by defining the key ID, the key format in hexadecimal format or plain text to provide security consideration on the authentication key, key activation time/start-time in date and minutes, and lifetime (validity duration of the key in terms of days and time) by using the **security key** command. The switch can have a maximum of 256 keys.

The following command configures SHA256 authentication algorithm as authentication key.

```
-> security key 5 algorithm sha256 key "passwordstring123" start-time 1/31/2017
00:00 life-time 180 10:30
```

The following command configures aes-gcm-128 authentication algorithm as the authentication key.

```
-> security key 1 algorithm aes-gcm-128 hex-key 0x0102030405060708090A0B0C0D0E0F
```

The following command configures aes-cmac-128 authentication algorithm as the authentication key.

```
-> security key 1 algorithm aes-cmac-128 hex-key
0x0102030405060708090A0B0C0D0E0F keyed-name 0x0102030405060708090A0B0C0D0FFF
```

**Note.**
- A keychain using the aes-gcm-128 authentication algorithm must be attached only to MACSec interface for its static-SAK configured under MACSec mode "static" only.

- A keychain using the aes-cmac-128 authentication algorithm must be attached only to MACSec interface for its Static Connectivity Association Key (static-CAK) using Pre-Shared key configured under MACSec mode "dynamic" only.

Use the **no** form of this command to delete a key. To delete a key, first disassociate the keychain from a user application, and detach the key from the keychain.

```
-> no security key 5
```

Use **show security key** command to view the configured keys in the system.

## Creating a Keychain

Use **security key-chain** command to create a keychain. There can be a maximum of 32 keychains in the device, and each of them can hold multiple keys of the same algorithm type.

```
-> security key-chain 1 globalKeyChain
```

Use the **no** form of this command to delete a keychain.

```
-> no security key-chain 1
```

To delete a keychain, first disassociate the keychain from the user application. Deleting a keychain will not delete the keys associated with the keychain. Keys will subsequently remain configured, but will not be associated to any keychain, until reassociation.

Use **show security key-chain** to view the configured keychains in the system

## Associating a Key into Keychain

Associate a key into the specified keychain by using the **security key-chain key** command.

```
-> security key-chain 1 key 5
```

Use the **no** form of this command disassociate a key from a security keychain.

```
-> no security key-chain 1
```

To disassociate a key from the keychain, first disassociate the keychain from the user application, and detach the key from the keychain.

# 4 Managing CMM Directory Content

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

## In This Chapter

This chapter describes the basic functions of CMM software directory management and how to implement them by using the Command Line Interface (CLI). CLI commands are used in the configuration examples; for more details about the syntax of commands, see the *OmniSwitch 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 4-2.
- A description of how to restore older versions of files and prevent switch downtime is described in "Software Rollback Feature" on page 4-3.
- The CLI commands available for use and the correct way to implement them are listed in "Managing Switch Configurations - Single CMM" on page 4-10.
- Managing, upgrading and restoring files using a USB flash drive described in "Using the USB Flash Drive" on page 4-20.
- The CLI command used to check the integrity of image files is described in "Checking the Integrity of the Image" on page 4-23

---

**Notes.**

- The format of the Alcatel-Lucent Enterprise certified USB Flash Drive must be FAT32. To avoid file corruption issues the USB Drive should be stopped before removing from a PC. Directory names are case sensitive and must be lower case.
- Many of the examples in this chapter 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 Enterprise. 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.

## 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 CERTIFIED directory if contents (images and vcboot.cfg) are different from the RUNNING DIRECTORY. If contents are the same, the switch will reboot from the RUNNING DIRECTORY.

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

You can determine where the switch is running from by using the **show running-directory** command described in "Show Currently Used Configuration" on page 4-15.

# Software Rollback Feature

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

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

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

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

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

## Software Rollback Configuration Scenarios

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

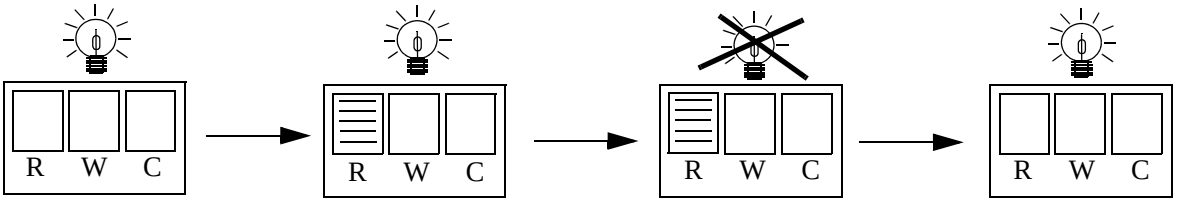
In the examples below, **R** represents the RUNNING CONFIGURATION, **W** represents the ***working*** directory, and **C** represents the ***certified*** directory.

### Scenario 1: Running Configuration Lost After Reboot

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

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

This is illustrated in the diagram below:




1. Switch boots from ***certified*** directory.

2. Changes are made to the running configuration but not saved.

3. Power is interrupted and the switch reboots.

4. Switch reboots from *certified* and all running configuration changes are lost since they weren't saved or certified.

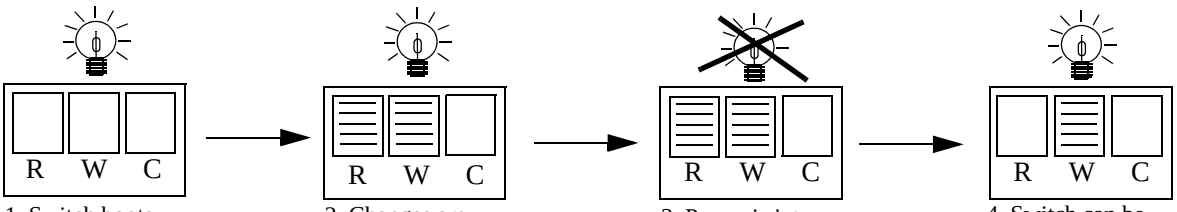
**Running Configuration is Overwritten by the Certified Directory on Reboot**

## Scenario 2: Running Configuration Saved to the Working Directory

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

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

This is illustrated in the diagram below:

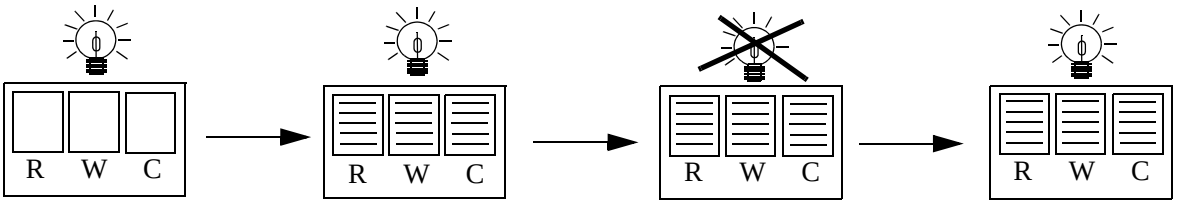



1. Switch boots from *certified* directory.
2. Changes are made to the running configuration and saved to *working* directory.
3. Power is interrupted and the switch goes down, rebooting from *certified* directory.
4. Switch can be rebooted from the *working* directory restoring the configuration.

**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.




1. Switch boots from *certified* directory.
2. Changes are made to the running configuration, stored in the *working* directory, and then saved to the *certified* directory.
3. Power is interrupted and the switch goes down.
4. Switch reboots from *certified* directory which now contains the updated switch configuration copied from the *working* directory.

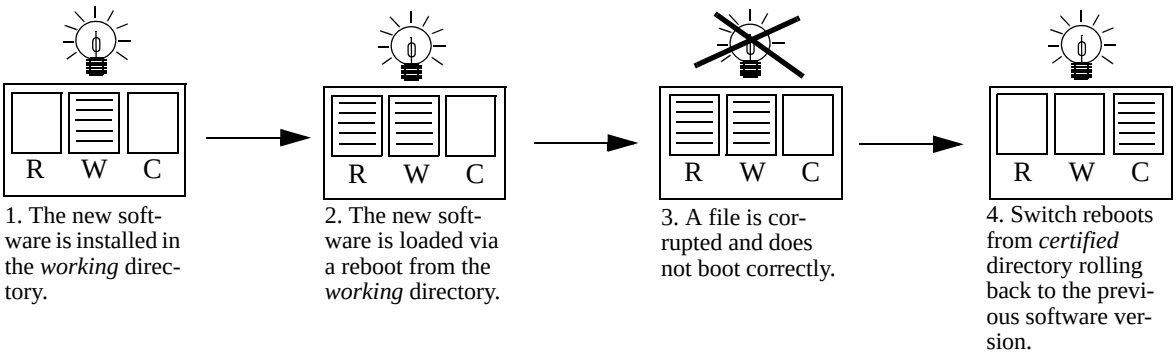
**Running Configuration is Saved to Working Directory, then to the Certified Directory**

## 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**

# Redundancy

CMM software redundancy is one of the switch's most important fail over features. For CMM software redundancy, two fully-operational CMM modules must be installed at all times. In addition, the CMM software must be synchronized. (Refer to "Synchronizing the Primary and Secondary CMMs" on page 4-17 for more information.)

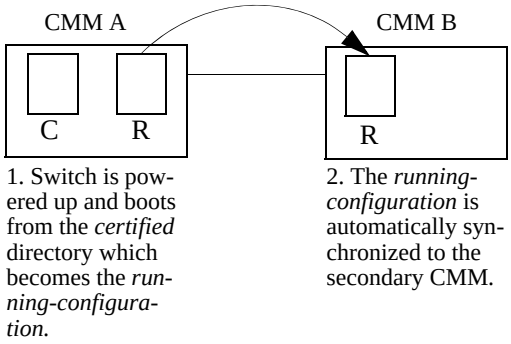
When two CMMs are running one CMM has the primary role and the other has the secondary role at any given time. The primary CMM manages the current switch operations while the secondary CMM provides backup (also referred to as "fail over").

## Redundancy Scenarios

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

### Scenario 1: Booting the Switch

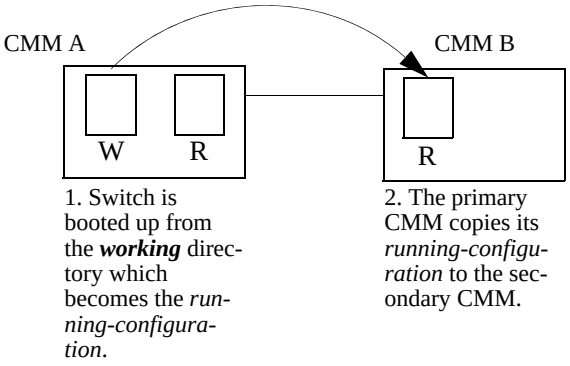
The following diagram illustrates what occurs when a switch powers up.




**Powering Up a Switch**

### Scenario 2: Rebooting from the Working Directory

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



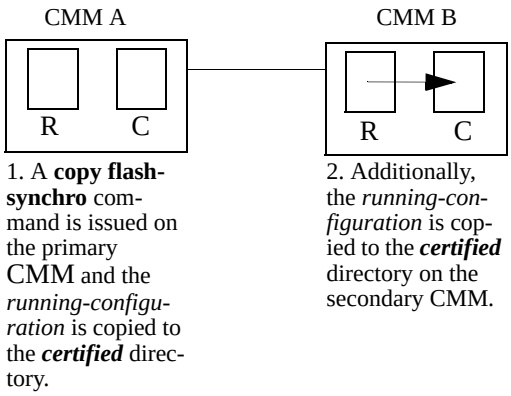

**Booting from the Working Directory**

**Note.** It is important to certify the *RUNNING-DIRECTORY* and synchronize the CMMS as soon as the validity of the software is established. Switches booted from the *RUNNING-DIRECTORY* are at risk of mismanaging data traffic due to incompatibilities in different versions of switch software. Certifying the *RUNNING-DIRECTORY* is described in "Copying the RUNNING DIRECTORY to the Certified Directory" on page 4-14, while synchronizing the switch is described in "Synchronizing the Primary and Secondary CMMs" on page 4-17.

## Scenario 3: Synchronizing CMMs

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

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




**Synchronizing CMMs**

The **copy flash-synchro** command (described in "Synchronizing the Primary and Secondary CMMs" on page 4-17) can be issued on its own, or in conjunction with the **copy running certified** command (described in "Synchronizing the Primary and Secondary CMMs" on page 4-17).

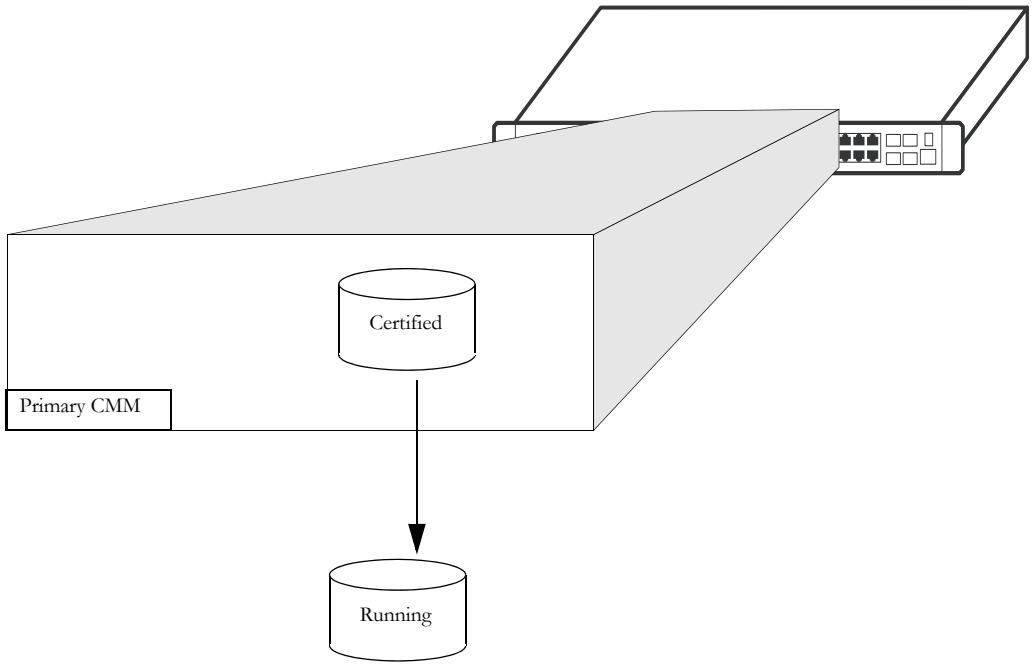
**Note.** It is important to certify the CMMs as soon as the validity of the software is established. Unsynchronized CMMs are at risk of mismanaging data traffic due to incompatibilities in different versions of switch software.

# Managing Switch Configurations - Single CMM

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

## Rebooting the Switch

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




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

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

```
-> reload all
```

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

---

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

---

### Scheduling a Reboot

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

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

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

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

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

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

### Canceling 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
```

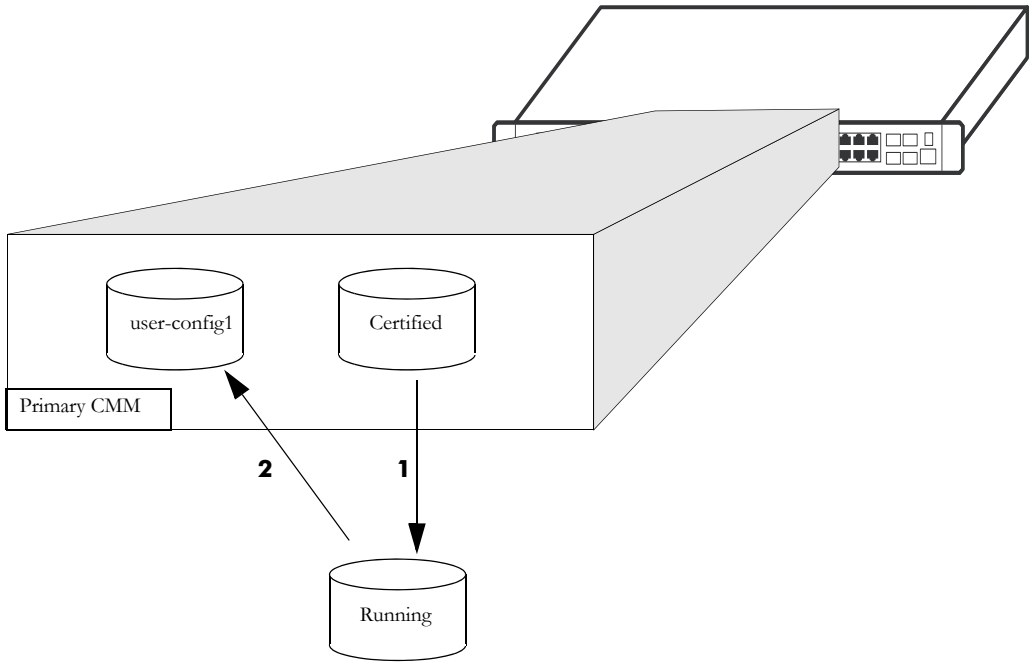
### Checking the Status of a Scheduled Reboot

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
```

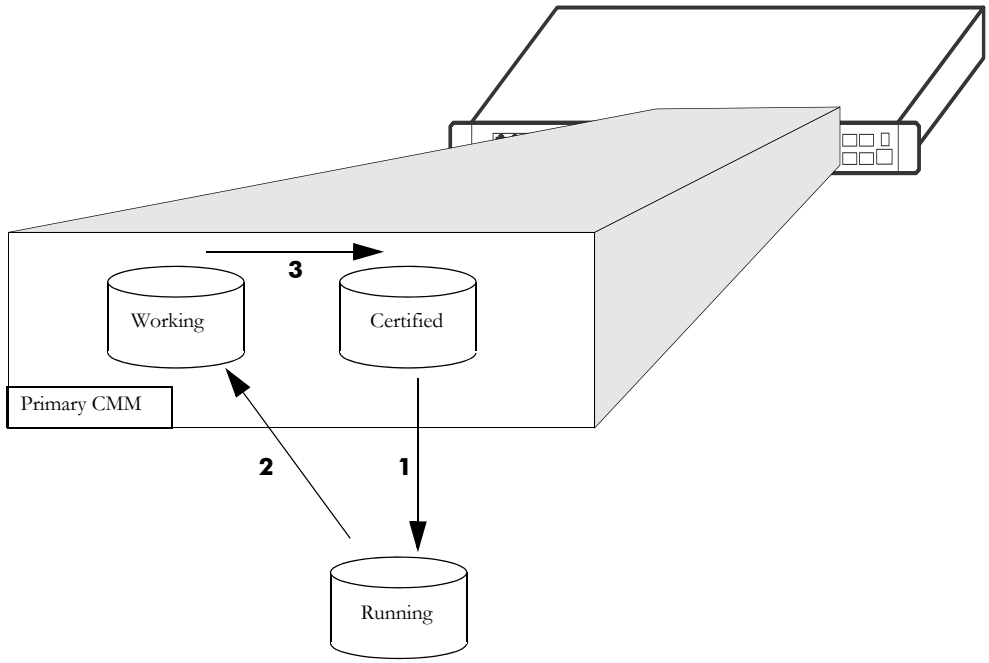
## Canceling a Rollback Timeout

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.

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

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

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

```
-> write memory
```

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

```
-> copy running certified
```

## Show Currently Used Configuration

Depending on how a a switch is booted different directories can become the RUNNING DIRECTORY. See "Where is the Switch Running From?" on page 4-2. for additional information.

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

```
-> show running-directory

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

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

## Show Switch Files

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

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

```
-> show microcode certified
   Package           Release       Size     Description
-----------------+--------------+--------+---------------------------------
Mos.img            8.3.1.211.R01    2486643 Alcatel-Lucent OS
Meni.img           8.3.1.211.R01     941331 Alcatel-Lucent NI
```

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

# Managing CMM Redundancy

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

## Rebooting the Secondary CMM

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

```
-> reload secondary
```

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

### Scheduling a Reboot

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

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

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

### Cancelling a Scheduled Reboot

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

```
-> reload secondary cancel
```

### Secondary CMM Fail Over

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

Synchronizing the primary and secondary CMMs is done using the **copy flash-synchro** command described in “Synchronizing the Primary and Secondary CMMs” on page 4-17.

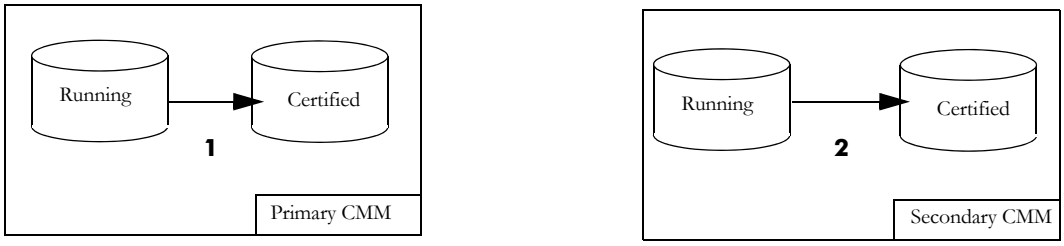
# Synchronizing the Primary and Secondary CMMs

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

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

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

The synchronization process is shown in the diagram below:




In the above diagram:

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

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

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

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

The **copy flash-synchro** command is described in detail in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

## Swapping the Primary CMM for the Secondary CMM

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

**Note.** It is important that the software for the secondary CMM has been synchronized with the primary CMM before you initiate a secondary CMM takeover. If the CMMs are not synchronized, the takeover could result in the switch running old or out-of-date software. Synchronizing the primary and secondary CMMs is described in "Synchronizing the Primary and Secondary CMMs" on page 4-17.

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

```
-> takeover
```

The **takeover** command is described in detail in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

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

## Show Currently Used Configuration

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

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

```
-> show running-directory

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

The **show running-directory** command is described in detail in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

# Using the USB Flash Drive

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

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

| Product Family Name | Auto-Copy Support | Disaster-Recovery Support |
|---|---|---|
| OmniSwitch 9900 | 9900/working | 9900/working<br>9900/certified |
| OmniSwitch 6900 | 6900/working | 6900/working<br>6900/certified |
| OmniSwitch 6860/6865 | 6860/working | 6860/working<br>6860/certified |
| OmniSwitch 6465 | 6465/working | 6465/working<br>6465/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 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 *6900/working* on the USB flash drive with all the proper image files, and issue the following commands.

```
-> usb enable

-> usb auto-copy enable copy config enable
```

**3** To encrypt all the configuration files and images to be copied in the USB, use a **key** or **hash-key** along with the command. For example:

```
-> usb auto-copy enable copy-config enable key “abc12345”
-> usb auto-copy enable copy-config enable hash-key “a05234d”
```

**4** 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.

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

---

**Note.** If **copy-config** is enabled, configuration files will also be copied in addition to the image files to the */flash/working* directory from */uflash/6900/working* directory

---

## Backup Files Using a USB Flash Drive

The following is an example of how to backup the images and configuration from certified and running directories to an USB Flash Drive using the **usb** and **usb backup admin-state** commands.

```
-> usb enable
-> usb backup admin-state enable
-> write memory
```

To encrypt all the configuration files and images to be copied in the USB, use a **key** or **hash-key** along with the command. For example::

```
-> usb backup admin-state enable key “abc12345”
-> usb backup admin-state enable hash-key “a05234d”
```

When the **write memory** command is executed in this example, the configuration files from */flash/running-directory* are copied to */uflash/6900/running-directory name.*

When the **copy running certified** command and the **write memory** command with the **flash-synchro** option is executed, the configuration and images from */flash/certified* are copied to */uflash/6900/certified.*

---

**Note.** Back-up cannot be enabled if auto-copy is enabled and auto-copy cannot be enabled if back-up is enabled. So only one of these features can be enabled at any given time.

---

## Disaster Recovery Using a USB Flash Drive

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

---

**Note.** Preparing the USB flash drive prior to needing it for disaster recovery is recommended. This example is for an OmniSwitch 6900, use the proper directory names based on the platform.

---

**1** Create the directory structure *6900/certified* and *6900/working* on the USB flash drive with the backup system and configuration files.

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

**3** Connect the USB flash drive to the CMM and reboot. The switch will automatically stop and display

the option to perform a disaster recovery.

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

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

# Checking the Integrity of the Image

To verify whether the SHA256 hash key of an image file located in the specified directory matches the SHA256 hash key in the specified key file, use the **image integrity-check** command with the name of the directory in "/flash" or include the full path (for example, "working" or "/flash/working"), and the name of the key file or include the full path (for example, "hash.txt" or "/flash/hash.txt").

If the name of the key file is specified without the directory path, the switch will look for the key file in the same directory specified for the image file.

The following format is used to store the hash key values in the key file:

Uos.img:f0ff173eff38e43e0598663da2185a363fcba5bd407201d7537d0a6b9f58670e

For example,

```
-> image integrity check /flash/working key-file /flash/hash.txt
This operation may take several minutes...
Success: Key matched.
```

To display the SHA256 hash key of the image present in the specified location, use the **image integrity get-key** command with the directory on the switch that contains the image file. Either the name of the directory in "/flash" or include the full path (for example, "working" or "/flash/working") can be provided.

When this command is entered, the SHA256 hash of the image files in the specified directory is calculated and displayed. It can be manually verified against the hash provided in the file.

To store the hash key value in a text file that can be used with the **image file integrity check** command, use the following format:

Uos.img:f0ff173eff38e43e0598663da2185a363fcba5bd407201d7537d0a6b9f58670e

```
-> image integrity get-key /working
This operation may take several minutes...

Image Name                  SHA256 Key
------------+----------------------------------------
Uos.img       c64d6b23312a6f9c4b99642b31ed0e87e600bce58d6fdd089d09e1f8077bd208

-> image integrity get-key /flash/certified
This operation may take several minutes...

Image Name                  SHA256 Key
------------+----------------------------------------
Uos.img       3d4d488a73eb798325bacb5793ef0d67bdf377527278a6732270d3a4801bb44b
```

# 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*.

# 5 Using the CLI

Alcatel-Lucent Enterprise'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 5-2
- "Command Entry Rules and Syntax" on page 5-3
- "Recalling the Previous Command Line" on page 5-5
- "Logging CLI Commands and Entry Results" on page 5-7

# 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 6, "Working With Configuration Files," for detailed information about configuration files.

# Command Entry Rules and Syntax

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

- Enter only one command per line.
- Passwords are case sensitive.
- Commands are *not* case sensitive. The switch accepts commands entered in upper case, lower case or a combination of both.
- Press Enter to complete each command line entry.
- To use spaces within a user-defined text string, you must enclose the entry in quotation marks (“ ”).
- If you receive a syntax error (i.e., ERROR: Invalid entry:), double-check your command as written and re-enter it exactly as described in the *OmniSwitch AOS Release 8 CLI Reference Guide*. Be sure to include all syntax option parameters.
- To exit the CLI, type **exit** and press Enter.
- AOS uses the Bash shell for CLI input. This could result in certain special characters being interpreted by Bash instead of being applied to an AOS command or password. For example, the '$' when interpreted by Bash causes the next characters to be interpreted as a variable or command line argument. If using special Bash characters (i.e. ‘$’ or ‘!’) in the CLI they should be enclosed in single quotes.

## 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.<br>Example: **show snmp station** |
| **“ ”** (Quotation Marks) | Used to enclose text strings that contain spaces<br>Example: **vlan 2 name “new test vlan”** |
| **‘ ‘** (Single Quotation Marks) | Used to enclose text strings that contain special Bash characters.<br>Example: **system name ‘system$name’** |

## 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 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 FSCK FREESPACE DEBUG
  COMMAND-LOG CHMOD
(System Service & File Mgmt Command Set)

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

*(Additional output not shown)*

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

### Recalling the Previous Command Line

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

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

```
-> ls
afn                     default_cportalCert.pem    dhcpdv6.pid
ca.d                    dhcpBind.db                lldpTrustedRemoteAgent.db
captive_portal          dhcpBind.db.backup         pre_banner.txt
cert.d                  dhcpClient.db              web
cloud                   dhcpd.pid                  zcfg
->
```

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 5-7.

---

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 5-8.

## 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 7, “Managing Switch User Accounts.”
- **Date.** Shows the date and time, down to the second, when the command was originally entered.
- **IP Addr.** The IP address of the terminal from which the command was entered.
- **Result.** The outcome of the command entry. If a command was entered successfully, the syntax **SUCCESS** displays in the Result field. If a syntax or configuration error occurred at the time a command was entered, details of the error display. For example:

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

# Customizing the Screen Display

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

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

## Changing the Screen Size

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

```
-> tty 10 150
```

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

## Changing the CLI Prompt

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

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

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

# Verifying CLI Usage

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

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

For more information about the resulting displays from these commands, see the *OmniSwitch AOS Release 8 CLI Reference Guide*. Additional information can also be found in “Using “Show” Commands” on page 5-4.

# 6 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 usability.

## In This Chapter

Configuration procedures described in this chapter include:


- "Tutorial for Creating a Configuration File" on page 6-2
- "Applying Configuration Files to the Switch" on page 6-5
- "Configuration File Error Reporting" on page 6-6
- "Text Editing on the Switch" on page 6-7
- "Creating Snapshot Configuration Files" on page 6-8

# 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 3, "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.

```
File configuration: none scheduled

Error file limit: 1
```

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

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

---

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

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

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

For more information about this display, see "Configuration File Manager Commands" in the *OmniSwitch AOS Release 8 CLI Reference Guide.*

---

## Setting an Application Session for a Date and Time

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

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

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

Error file limit: 1
```

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

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

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

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

Error file limit: 1

Running configuration and saved configuration are different
```

For more information about this display see "Configuration File Manager Commands" in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

## Setting an Application Session for a Specified Time Period

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

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

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

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

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

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

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

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

For more information about this display see "Configuration File Manager Commands" in the *OmniSwitch AOS Release 8 CLI Reference Guide*.

# Configuration Files Overview

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

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

## Applying Configuration Files to the Switch

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

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

- For information on applying a file immediately, refer to "Setting a File for Immediate Application" on page 6-3.
- 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 6-3.
- 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 6-4.

### 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
```

## Canceling a Timed Session

You may cancel a pending timed session by using the **configuration cancel** command. To confirm that your timer session has been canceled, 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 6-7.

### 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, use the '?' to display the list of 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 OS6900
! Configuration:
! VLAN :
! IP :
ip service all
icmp unreachable net-unreachable disable
ip interface "vlan-1" address 10.255.211.70 mask 255.255.255.192 vlan 1 mtu 1500
ifindex 1
! IPMS :
! AAA :
aaa authentication default "local"
aaa authentication console "local"
! PARTM :
! AVLAN :
! 802.1x :
! QOS :
! Policy manager :
! Session manager :
! SNMP :
snmp security no security
snmp community map mode off
! IP route manager :
ip static-route 0.0.0.0 mask 0.0.0.0 gateway 10.255.211.65 metric 1
! RIP :
! OSPF :
! BGP :
! IP multicast :
! IPv6 :
! RIPng :
! Health monitor :
! Interface :
! Link Aggregate :
! VLAN AGG:
! 802.1Q :
! Spanning tree :
bridge mode 1x1
! Bridging :
source-learning chassis hardware
! Bridging :
! Port mirroring :
! UDP Relay :
! Server load balance :
! System service :
! VRRP :
! Web :
! Module :
! NTP :
! RDP :
```

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

# Verifying File Configuration

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

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

# 7 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 (for example, 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 8, "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 7-7.
- "Configuring Password Policy Settings" on page 7-9.
- "Configuring Privileges for a User" on page 7-14.
- "Setting Up SNMP Access for a User Account" on page 7-15.
- "Multiple User Sessions" on page 7-17.


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 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 7-3 and "Default User Settings" on page 7-6.
- New users inherit the privileges of the **default** user if the specific privileges for the user are not configured; the default user is modifiable.
- Password defaults are as follows:

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

- Global user account lockout defaults are as follows:

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

# Overview of User Accounts

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

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

---

**Note.** For information about setting up user information on an authentication (AAA) server, see the "Managing Authentication Servers" chapter of the *OmniSwitch AOS Release 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 7-10.

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 7-14 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 7-15 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 7-7.

**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 7-14.

---

**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 7-9 and “Configuring Global User Lockout Settings” on page 7-12 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 7-6.

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.

An option to enter the password in a obscured format rather than as clear text is provided. While creating a user, **password-prompt** option can be used with the 'user' command to configure the password for the user. When this option is selected, a password prompt appears and the password can be provided. Password needs to be re-entered, and only if both the passwords match, command is accepted. Password provided in this mode is not displayed on the CLI as text.

For example,

```
-> user techpubs password-prompt
Password: *******
Re-enter password: ********
```

## Removing a User

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

```
-> no user thomas
```

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

## User-Configured Password

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

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

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

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

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

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

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

**4** Enter the password again.

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

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

# Configuring Password Policy Settings

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

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

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

To view the current policy configuration, use the **show user password-policy** command. For more information about this command and those used in the configuration examples throughout this section, see the *OmniSwitch AOS Release 8 CLI Reference Guide*.

## Setting a Minimum Password Size

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

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

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

## Configuring the Username Password Exception

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

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

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

# Configuring Password Character Requirements

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

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

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

# Configuring Password Expiration

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

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

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

## Default Password Expiration

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

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

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

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

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

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

### Specific User Password Expiration

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

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

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

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

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

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

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

## Configuring the Password History

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

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

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

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

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

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

## Configuring the Minimum Age for a Password

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

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

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

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

# Configuring Global User Lockout Settings

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

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

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

---

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

---

Lockout settings are saved *automatically*; that is, these settings do not require the **write memory** 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 AOS Release 8 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 7-13) 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 7-13).

## 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 7-12). 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 7-13) 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 7-12).

## 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 7-13) 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.

Use the keyword **all-except** to disable the function privileges for a specific family or domain for a user. The following example creates a user with read-write privileges for all families except 'aaa'.

```
-> user techpubs password writer read-write all-except aaa
```

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

# 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.
- SHA and AES— The SHA authentication algorithm and AES encryption standard is used for authenticating and encrypting SNMP PDU for the user.
- SHA224— The SHA224 authentication algorithm is used for storing the user passwords.
- SHA256— The SHA256 authentication algorithm is used for storing the user passwords.

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 10, "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 10, "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 6, "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 7-5.

# 8 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 8-6
- "Setting Up Management Interfaces for ASA" on page 8-9
- "Configuring Accounting for ASA" on page 8-11
- "Authenticated Switch Access - Enhanced Mode" on page 8-12
- "Joint Interoperability Test Command - JITC Mode" on page 8-18


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 7, "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 2, "Logging Into the Switch."

# Switch Security Defaults

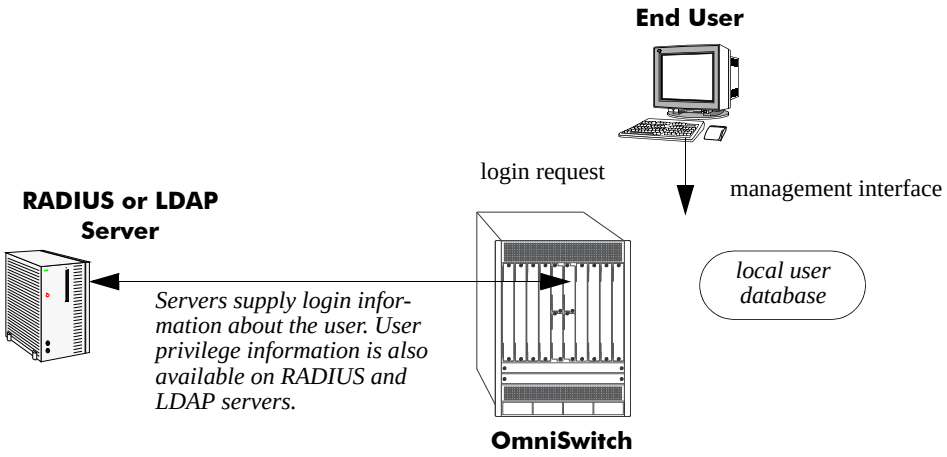
Access to managing the switch is always available for the **admin** user through the console port, even if management access to the console port is disabled for other users.

| 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 7, "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 8-11. 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.

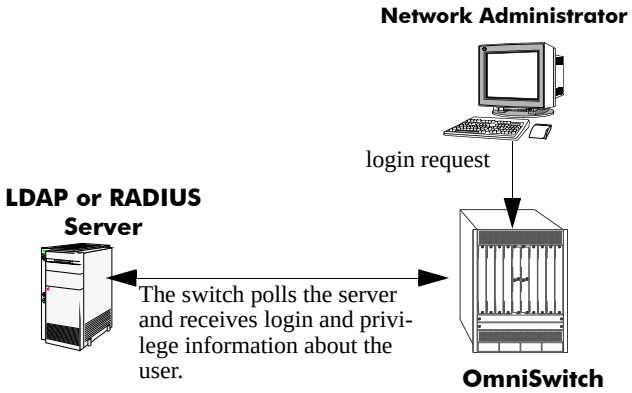
This section describes how to configure management interfaces for authenticated access as well as how to specify external servers that the switch can poll for login information. The type of server may be an authentication-only mechanism or an authentication, authorization, and accounting (AAA) mechanism.

## 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:




**AAA Server (LDAP or RADIUS)**

For more information about types of users, see Chapter 7, “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 7, "Managing Switch User Accounts," for more information about setting up the user database.

## Secure Console Session for Admin User Only

This feature can be used to restrict all users except the user "admin" from accessing the switch through the secure console session.

To enable the restriction, use the **aaa console admin-only** command. For example:

```
-> aaa console admin-only enable
```

To disable the restriction, use the **aaa console admin-only** command. For example:

```
-> aaa console admin-only disable
```

---

**Note.**

- The secure console sessions which are already established before enabling this feature will continue to work until the session timer is refreshed.
- If the root access is enabled, the root user shall be able to connect to the super user mode.

---

# 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 7, "Managing Switch User Accounts."

**3 Set Up the Management Interfaces**. This procedure is described in "Setting Up Management Interfaces for ASA" on page 8-9.

**4 Set Up Accounting**. This step is optional and is described in "Configuring Accounting for ASA" on page 8-11.

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 7, "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**<br>**aaa tacacs+-server** | Setting up the switch to communicate with external RADIUS or LDAP authentication servers. |
| **aaa authentication** | Configuring the management interface and specifying the servers and/or local user database to be used for the interface. |
| **aaa accounting session** | *Optional*. Specifies servers to be used for accounting. |

# Quick Steps for Setting Up ASA

**1** If the local user database is used for user login information, set up user accounts through the **user** command. In this example, user privileges are configured:

```
-> user thomas password mypassword read-write all
```

**2** If an external RADIUS or LDAP server is used for user login information, use the **aaa radius-server** or **aaa tacacs+-server** commands to configure the switch to communicate with these servers. For example:

```
-> aaa radius-server rad1 host 10.10.1.2 timeout 3
```

For more information, see the "Managing Authentication Servers" chapter in the *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 2, "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 8-11.

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 |
| **Note:** Remote authentication is not supported on secondary CMMs or Slave chassis. Use local authentication on secondary CMMs and Slave chassis. | |

## 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.

# Authenticated Switch Access - Enhanced Mode

Authenticated Switch Access - Enhanced Mode feature allows configuration of enhanced security restrictions to the OmniSwitch.

## Configuring the ASA Mode

Set the access mode to enhanced or default mode by using the **aaa switch-access mode** command. Enhanced mode enables enhanced set of security options for switch access.

---

**Note.** It is recommended to save the configuration and reboot the switch when the ASA access mode is configured.

---

For example, the following command sets the access mode to default.

```
-> aaa switch-access mode default
```

The following command sets the access mode to enhanced mode.

```
-> aaa switch-access mode enhanced
```

The following functionality come into effect when the ASA enhanced mode is activated:

- When the enhanced mode is initially activated, the default password-policy and lockout settings are automatically set to enhanced mode default values. When the switch boots up with a vcboot.cfg configuration file that has the enhanced ASA mode activated, LockoutSetting file will be considered for the modified lockout settings as the modified values will not be stored in vcboot.cfg.
- The user has to re-authenticate before entering to super user mode. The switch verifies whether the user of the current session has the privilege to access the super user mode. If the user has enough privilege, then the switch prompts for a password, if not, the switch prompts for the user credentials too with enough privilege. Only if the authentication is successful, then the user shall be allowed to access the mode prompt.
- Default password **switch** cannot be set anymore as it does not meet the enhanced mode password policy. User ‘admin’ shall be forced to change the password upon login if the password was not changed from the default password ‘switch’.
- The following table lists the factory default and the ASA enhanced mode values for password policy and user lockout parameters:

| Parameters | ASA enhanced mode default values | Factory default values |
|---|---|---|
| **User Password Policy** | | |
| Minimum size | 9 | 8 |
| Password expiration | Disable | Disable |
| Password cannot contain username | No | No |
| Minimum number of English uppercase characters | 1 | Disable |

| Parameters | ASA enhanced mode default values | Factory default values |
|---|---|---|
| Minimum number of English lowercase characters | 1 | Disable |
| Minimum number of base-10 digit | 1 | Disable |
| Minimum number of non-alphanumeric | 1 | Disable |
| Password history | 4 | 4 |
| Password minimum age | Disable | Disable |
| **User Lockout Setting** | | |
| Observation window | 1 minute | Disable |
| Duration | 5 minutes | Disable |
| Threshold | 3 | Disable |

- If the mode is changed from default to enhanced and if the user password policy settings and the user lockout settings have the default mode default values, then corresponding enhanced mode default values will be assigned. If the user password policy settings and the user lockout settings are assigned with non-default values in the default mode, then the same values will be carried to the enhanced mode.
- When the mode is changed from enhanced to default, user password policy settings and user lockout settings will be restored back to switch's default mode default values. Only those configurations modified in the enhanced mode will be retained on the switch after reload.
- When the enhanced mode is initially activated, since the password policy is automatically set to enhanced mode default values, any login request through SNMP and FTP that does not follow the enhanced mode password policy shall be considered as authentication failure.
- In enhanced mode, a given user is restricted to only one session. For example, if a user 'admin' has already logged in a session, another session with the same user 'admin' is not allowed, and the new session login is refused. This is applicable for both local and external users. If the user authentication fails, the login failure attempt is considered as an invalid login attempt for IP lockout count.
- A user account will be locked after the authentication failure based on the threshold value within the observation window duration, irrespective of the access method. The user account will remain locked for the lockout duration (lockout-window, lockout-threshold, and lockout-duration is based on the configured or default values.) This is only supported for local users.
- When the enhanced mode is activated, other existing sessions will not be logged out. The change of password for internal or external user will not impact existing sessions until they log out.
- When the ASA mode is set to enhanced or default, the changes will take effect in secondary after write memory flash-synchro.
- Any local user who logs in with the password that does not comply with the enhanced mode password policy will be prompted to change the password.
- Enhanced mode allows the dynamic alignment of IP services like telnet, FTP, SSH, to the AAA authentication status in the default VRF. However, existing command **[no] ip service** can be used to enable or disable individual IP services.

- When enhanced mode is activated, TLS connections use only TLS version 1.2. Connection requests with TLS version 1.1 and lower shall be rejected. This is applicable only for Captive Portal and WebView.
- In the enhanced mode, all login attempts to the switch is logged along with the user name, IP address of the host, switch access type like telnet, SSH, console and so on along with the authentication status.
- In the enhanced mode, when the switch logging file reaches 90% of the configured threshold value, a SWLOG message is displayed in the console and a trap is generated to alert the administrator to take a backup of the SWLOG file before it is overwritten.
  For example, following message is displayed:

  ```
  Sun Mar 29 12:42:15 : SSAPP main info message:
  +++ Switch log file reached 90%, Backup files before overwritten
  ```

- AOS supports both DSA 1024 and RSA 2048 public key algorithms for SSH private and SSH public keys in enhanced mode. WebView access supports connection over TLS. In the enhanced mode, the default certificates are generated with RSA 2048 bit keys.
- When the switch is in ASA enhanced mode, both user name and password is prompted to view the SWLOG data when **show log swlog** commands are used by the users. Only those users who provide valid ASA credentials are allowed to view the SWLOG data. For more information on the switch logging commands, refer chapter Switch Logging Commands in *OmniSwitch AOS Release 8 CLI Reference Guide*.

## Configuring the IP Lockout Threshold Value

The lockout threshold number specifies the number of failed login attempts from an IP address after which the IP address will be banned from switch access.

By default, the lockout threshold value is set to 6. To configure a lockout threshold number, use the **aaa switch-access ip-lockout-threshold** command. For example:

```
-> aaa switch-access ip-lockout-threshold 2
```

IP address is permanently blocked/banned if the number of authentication failures from a particular IP reaches the IP lockout threshold within the window, which is two times of the user lockout window.

A maximum of 128 IPs will be added to the banned list. When the maximum limit has reached, oldest entry from the list is removed to accommodate the new entries.

## Unlock/Release Banned or Locked IP

To release the banned IP addresses that are blocked due to failed login attempts, use the **aaa switch-access banned-ip release** command. For example:

```
-> aaa switch-access banned-ip all release

-> aaa switch-access banned-ip 100.2.45.56 release
```

## Configuring Privileges for an Access Type

Configure the functional privileges mask for the switch access based on the access type on top of the user privilege. The access privileges for the SSH, TELNET, Console, HTTP, HTTPS can be defined 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.

Possible values for domains and families are listed in the table 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 ipmr ipms |
| domain-layer2 | vlan bridge stp 802.1q linkagg ip-helper |
| domain-service | dns |
| domain-policy | qos policy slb |
| domain-security | session avlan aaa |
| domain-mpls | mpls |
| domain-datacenter | fips, auto-fabric |
| domain-afn | sip-snooping, dpi, app-mon |

In addition to command families, the keywords **all** or **none** can be used to set privileges for all command families or no command families respectively. And, use the **all-except** keyword to disable functional privileges for specific families for an access type.

An example of setting up access type privileges:

```
-> aaa switch-access priv-mask ssh read-write ripng rip rdp qos port-mapping pmm
```

Use the keyword **all** to specify that all command families and domains are available to the user for a specific access type.

```
-> aaa switch-access priv-mask ssh read-write all
```

Use the keyword **all-except** to disable function privileges for a specific family for an access type. The following example creates read-only privileges for SSH for all the families except VLAN.

```
-> aaa switch-access priv-mask ssh read-only all-except vlan
```

If privileges for specific families need to be re-applied, then remove the existing privilege using the **no** command, and re-apply the required family privilege.

```
-> no aaa switch-access priv-mask telnet read-write all
-> aaa switch-access priv-mask telnet read-write vlan aaa
```

## Configuring Management Station

Enable or disable the IP management station feature in a switch.

When the IP management station is disabled, the switch access from any IP address is allowed. After login failure, based on the lockout threshold value, (**ip-lockout threshold**) those IP address are banned/blocked and are added to the banned IP address list.

When the management station is enabled, the switch access is allowed only from those IP addresses configured as management station IP, and only if they are not in the banned list.

To enable the IP management station feature in a switch, use the enable option in the **aaa switch-access management-stations admin-state** command. Enable the management station from the console to avoid termination of any session.

```
-> aaa switch-access management stations admin-state enable
```

To configure the IP address for the management station, use the **aaa switch-access management-stations** command. The remote access is allowed only from these IP addresses. A maximum of 64 management stations can be configured.

```
-> aaa switch-access management stations 100.15.5.9

-> aaa switch-access management stations 100.15.5.9 255.255.255.0
```

To disable the IP management station feature in a switch, use the disable option in the **aaa switch-access management-stations admin-state** command. By default, the IP management station feature state is disabled.

```
-> aaa switch-access management stations admin-state disable
```

## Process Self-test Function Commands

When the switch is in the ASA enhanced mode, an option is provided to check the hardware and software status during boot up. The following commands can be used to perform a self-test for the hardware components and software processes sanity as and when necessary. This functionality is applicable only in ASA enhanced mode.

To displays the major hardware components status, use the **show aaa switch-access hardware-self-test** command.

```
-> show aaa switch-access hardware-self-test
Checking CPU status -> Ok
Checking Memory status -> Ok
Checking Flash Status -> Ok
Checking NI Module status -> Ok
Checking Power Supply status -> Ok
Checking Lanpower Status -> Ok
Checking GBIC Status -> Ok
```

To display the major software process status, use the **show aaa switch-access process-self-test** command:

```
-> show aaa switch-access process-self-test
------------+-------------------
      Process Self Test

------------+-------------------
Checking Chassis Supervision Process -> Ok
Checking AAA Process -> Ok
Checking Configuration Manager Process -> Ok
Checking Network Process -> Ok
Checking QoS Process -> Ok
Checking VLAN Manager Process -> Ok
Checking H/W Driver Process -> Ok
Checking Layer2/Switching -> Ok
Checking Layer3/Switching -> Ok
```

# Joint Interoperability Test Command - JITC Mode

Joint Interoperability Test Command (JITC) is a certification agency which provides risk based Test Evaluation & Certification services, tools, and environments for certifying IT products that are used in military and defense networks.

In JITC mode, the OmniSwitch enforces additional security measures as per the JITC certification agency requirements.

## Configuring the JITC Mode

To enable JITC mode on the switch, use the **aaa jitc admin-state** CLI command.

```
-> aaa jitc admin-state enable

WARNING: JITC mode configuration is applied only after reload
```

Save the configuration and reboot the switch for the JITC mode to be activated.

---

**Note.** Before enabling JITC mode, ensure enhanced mode or common criteria mode is disabled. JITC mode is mutually exclusive of enhanced mode and common criteria mode.

---

The following functionality comes into effect when the JITC mode is activated:

- The switch will display the date and time, the location of the last logon, the number of unsuccessful and successful login attempts of the administrator account on the SSH and Console session.
- The switch will store the successful and unsuccessful login attempts of the user and is displayed in the console session when the administrator logs into the switch. The record is stored for a 24 hour time period after which the login statistics are reset.
- The following user authentication changes are applied when JITC is activated:
  - The minimum password length must be 15 characters or more. The user's with shorter password (less than 15 characters) will be forced to change the password.
  - The new password cannot be same as last five password.
  - The password expiration is by default set to 60 days.
  - The password expiration policy is applied to all the users except the admin (user with read and write privilege for all domains) user.
  - During password change it is required the characters are changed in at least eight positions within the password.
- When a user account is created, modified, and deleted on the switch, the administrator is notified in the swlog messages and SNMP traps.
- The switch will capture the successful and unsuccessful attempts to access, modify, or delete privileges. The information can be viewed in the Swlog of the switch.
- The SSH sessions will rekey at a minimum every one gigabyte or every 60 minutes of data received or transmitted.
- SSH uses Diffie-Hellman-Group14-SHA1 algorithm as the preferred key exchange mechanism.

- When the external TLS server does not support renegotiation_info extension (RFC 5746), the AOS TLS client applications actively terminates the TLS session.
- No compression is enabled in TLS communication by default.
- Site-Local IPV6 addresses of range FEC0::/10 (FEC, FED, FEE and FEF) cannot be configured.
- Software upgrades are allowed only after the digital signature of the software component is verified. During software upgrade the SHA256 checksum of the images is verified against a file "imgsha256sum" stored in the image directory. If the checksum matches, the software upgrade is allowed.
- Swlog displays the start and end time of the administrator access to the system.

To verify the operational status of JITC, use the **show aaa jitc config** CLI command.

# Verifying the ASA Configuration

To display information about management interfaces used for Authenticated Switch Access, and ASA enhanced mode configuration, 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. |
| **show aaa switch-access mode** | Displays the global access mode configuration. |
| **show aaa switch-access ip-lockout-threshold** | Displays the lockout threshold configured for the remote IP addresses. |
| **show aaa switch-access banned-ip** | Displays the list of banned ip addresses. |
| **show aaa switch-access priv-mask** | Displays the privilege details for access types. |
| **show aaa switch-access management-stations** | Displays the list of configured management stations. |

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 8-7.

# 9 Using WebView

The switch can be monitored and configured using WebView, Alcatel-Lucent Enterprise'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 9-2)
- WebView Quick Steps (see "WebView Page Layout" on page 9-4)

# WebView CLI Defaults

Web Management Command Line Interface (CLI) commands allow you to enable/disable WebView, enable/disable Secure Socket Layer (SSL), and view basic WebView parameters. These configuration options are also available in WebView. The following table lists the defaults for WebView configuration.

| Description | Command | Default |
|---|---|---|
| WebView Server | **webview server** | enabled |
| WebView Access | **webview access** | enabled |
| Force SSL | **webview force-ssl** | enabled |
| HTTPS port | **webview https-port** | 443 |
| HTTP port | **webview http-port** | 80 |
| WebView WLAN Cluster-Virtual-IP Precedence | **webview wlan cluster-virtual-ip precedence** | lldp |

# 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 **webview wlan cluster-virtual-ip** 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.

---

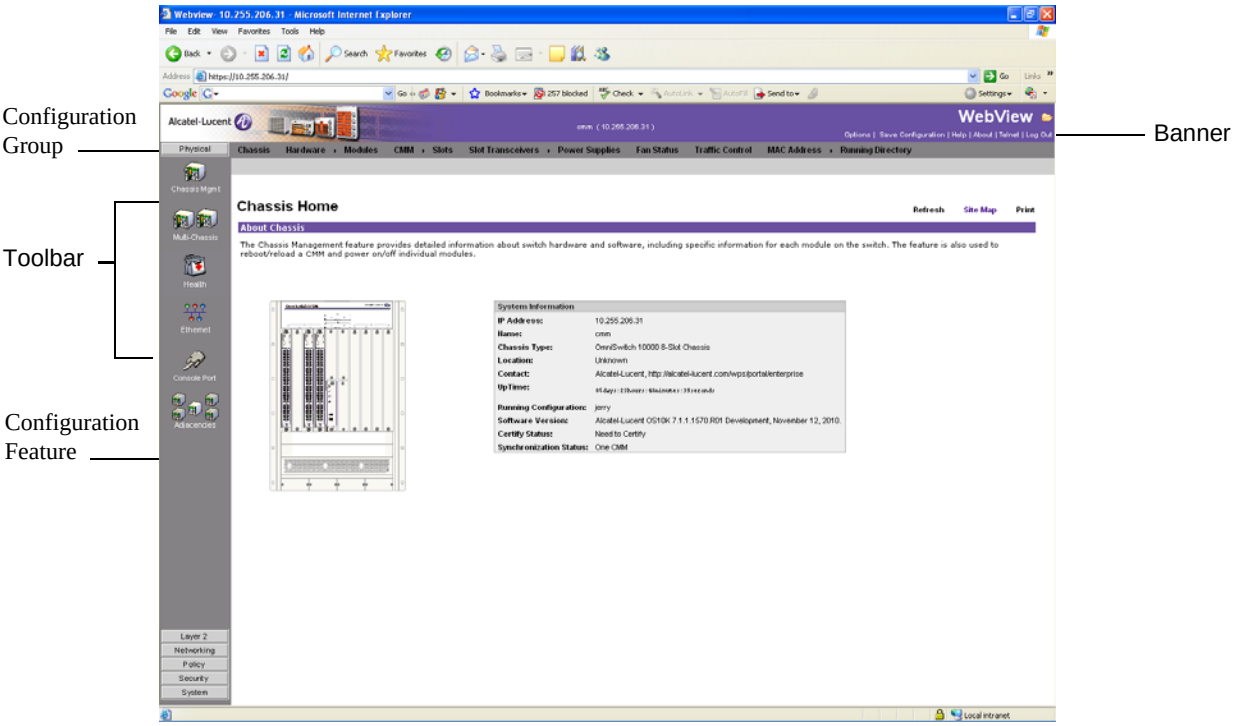
# WebView Overview

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

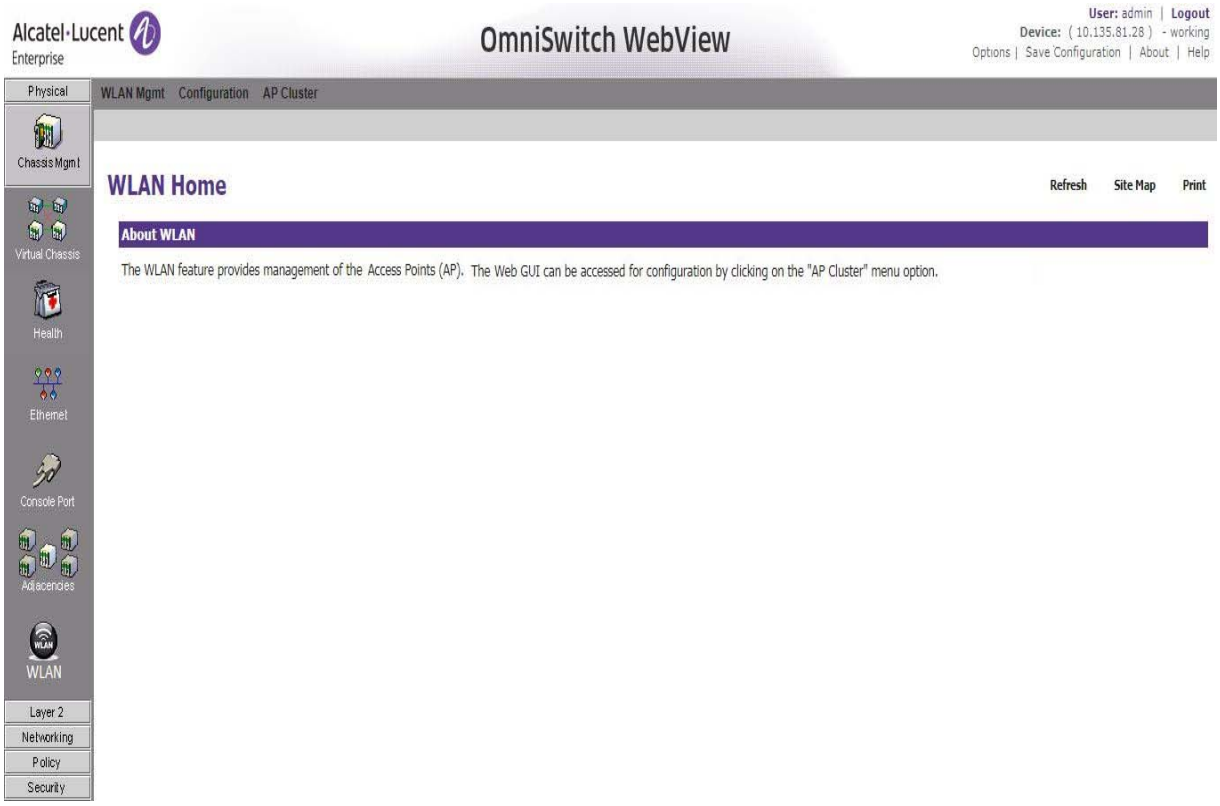
Feature configuration options are displayed as drop-down menus at the top of each feature page.

## 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.

# OAW-AP Web Management Configuration

The OAW-APs can be managed from the OAW-AP web interface. The OAW-AP web interface can be accessed from the WebView page by clicking on the **WLAN** button under the Physical group.




**WLAN WebView Page**

In order to access the OAW-AP web management interface, the switch must be aware of the Virtual Cluster IP of the AP. The WLAN web management can be used to configure and redirect the switch to the URL of the AP (Virtual IP Address) URL from where the OAW-APs can be managed. The Virtual Cluster IP address can be configured using the CLI on the OmniSwitch or from the WebView page.

## Configuring the Virtual Cluster IP address for OAW-AP Web Management using CLI

To configure the AP Virtual Cluster IP address using the CLI, use the **webview wlan cluster-virtual-ip** CLI command. For example:

```
-> webview wlan cluster-virtual-ip 10.25.6.8
```

## Automatic Configuration of Cluster Virtual IP Address

The Cluster Virtual IP address to access the group of APs through OmniSwitch Webview can be automatically configured. The OmniSwitch acquires the Cluster Virtual IP address from the LLDP TLV received from the Access Points (AP).

All AP belonging to the same L2 domain and having the same cluster-ID are grouped into a single cluster. Each of these APs have their own unique IP address and the cluster is associated with a single virtual IP address for management. The cluster can be configured or managed through a Web interface by connecting to the cluster virtual IP address. The cluster virtual IP address is associated with the primary AP of the cluster. The OmniSwitch automatically configures the cluster virtual IP address from the received LLDP packets from the APs.

## Enabling Automatic Configuration of Cluster Virtual IP Address

To automatically configure the cluster virtual IP address the precedence to obtain the cluster IP address from the LLDP packets must be set. To set the precedence for LLDP packets received from the APs, use the **webview wlan cluster-virtual-ip precedence** command. For example, the following command sets the precedence for LLDP packets:

```
-> webview wlan cluster-virtual-ip precedence lldp
```

---

**Note.** By default, the precedence is set for LLDP packets.

---

However, the precedence can be changed to the manually configured cluster virtual IP address. To set the precedence for manually configured virtual IP address, use the **webview wlan cluster-virtual-ip precedence** command. For example, the following command sets the precedence for manually configured IP address:

```
-> webview wlan cluster-virtual-ip precedence configured
```

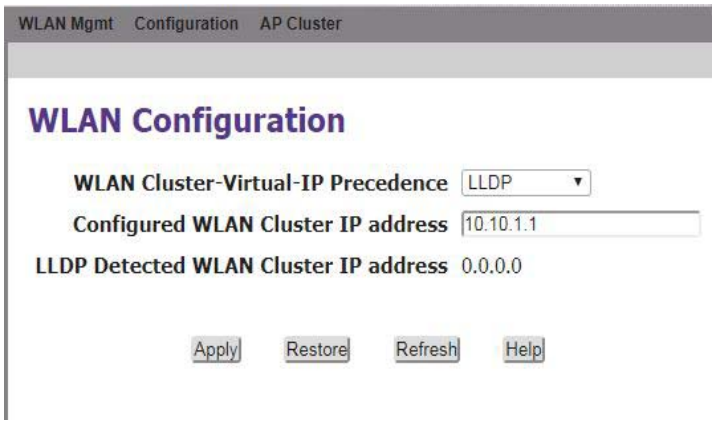
The configuration can be verified using the **show webview wlan config** command.

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

## Configuring the Virtual Cluster IP Address for OAW-AP Web Management Using WebView

The Virtual Cluster IP address of the AP can be configured from the WebView page by clicking on the **WLAN** button under the Physical group. The WLAN WebView page is displayed.

Click on the **Configuration** tab to configure the Virtual Cluster IP address of the AP.




**WLAN Virtual IP Configuration**

Set the precedence to obtain the cluster virtual IP address from the **WLAN Cluster-Virtual-IP Precedence** drop down box. If LLDP is selected, then the precedence to obtain the cluster virtual IP address is set to LLDP packets coming from the APs. If Configured is selected, then the precedence to obtain the cluster virtual IP address is set to the manually configured IP address.

To manually configure the cluster virtual IP address, enter the cluster IP address in the **Configure WLAN Cluster IP address** box.

Click **Apply** to apply the changes. The Virtual Cluster IP address is configured.

Click **Restore** to restore the previous configuration.

Click **Refresh** to refresh the WLAN configuration page.

---

**Note.** By default, the precedence is set to LLDP.

---

## Verifying the WLAN Configuration

The Virtual Cluster IP address configuration can be verified in the WLAN Configuration screen in the WebView or by using the **show webview wlan config** CLI on the OmniSwitch. For example:

```
-> show webview wlan config
WebView WLAN Cluster-Virtual-IP Precedence = LLDP
WebView WLAN Cluster-Virtual-IP configured address = 0.0.0.0
WebView WLAN Cluster-Virtual-IP LLDP address = 1.1.1.1
```

The output displays the precedence set for obtaining the cluster virtual IP address, the configured cluster virtual IP address, and the cluster virtual IP address obtained from LLDP.

# 10 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 10-3
- ‘‘Setting Up Trap Filters’’ on page 10-4
- “Using SNMP For Switch Security” on page 10-10
- “Configure SNMP Engine ID” on page 10-15
- “Working with SNMP Traps” on page 10-16


This chapter also includes lists of Industry Standard and Enterprise (Proprietary) MIBs used to manage the OmniSwitch.

# 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 10-10. 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 OmniSwitch 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 Enterprise'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 Enterprise 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 10-3

### Configuring the Security Modes and User Certificate Identity for Management Stations

To send SNMP traps over TLS connection, the SNMP station needs to be configured with TSM user along with certificate identities.These configurations are supported only for SNMP version 3.

~~Two security models, TSM and USM are available for SNMP stations. By default, the security mode is set to USM~~. Use the **snmp station** command to configure the TSM security mode. For example:

```
-> snmp station 168.22.1.1 joe v3 tsm local-identity aluSubagent.crt remote-
identity manager.crt enable
```

When the TSM security model is enabled, all the v1/v2/v3 USM request and traps are discarded. The SNMP requests are supported only over IPv4 transport.

When TSM security model is disabled, all v1/v2/v3 (USM and TSM) requests and traps are allowed.

The TSM mode requires the users local and remote identity to be configured.

**Note.** If the contents of local or remote certificates are changed, the updated certificates must be manually copied from master or primary to all secondaries and slaves. A reboot is required for the changes to be applied.

To view the configuration details, use the **show snmp station** command. For example:

```
   -> show snmp station details
ipAddress/port:    10.255.24.59/162,
  status:          disable,
  protocol:        v2,
  user:            public,
ipAddress/port:    localhost/10162,
  status:          disable,
  protocol:        v3,
  security model:  tsm,
  user:            joecool,
  local identity:  aluSubagent.crt,
  remote identity: manager.crt,
```

# 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 read-only all
```

**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
```

**Note.** This display also verifies that the community map mode is enabled.

# Configuring TLS encryption for SNMP

TLS encryption can be enabled for the SNMP connections for enhanced security. To enable TLS encryption use the **snmp security tsm** command. For example:

```
-> snmp security tsm enable
```

To view the configuration status of the TLS encryption over SNMP, use the **show snmp security** command.

---

**Note.** The TLS encryption can be enabled only for SNMP version 3.
In Common Criteria mode (CC mode) TLS encryption for SNMP is enabled by default and cannot be disabled.

---

## Mapping the remote certificate for TLS authentication

The user account must be mapped to the remote certificate in TSM mode. To map the remote identity to a user, use the **snmp tsm-map** command. For example:

```
-> snmp tsm-map remote-identity manager.crt user joe
```

If the content of remote certificate is changed, the updated certificate must be manually copied from master or primary to all secondaries and slaves. A reboot is required for the changes to be applied.

The remote identity mapping can be done for only one user at a time. It cannot be mapped to multiple users. Mapping it to a different user will replace the existing user.

---

**Note.** Use the **show snmp tsm-map** command to view the SNMP remote identity mapping for the user.

---

# 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 DES and AES encryption. The **sha**, **md5**, **sha+des**, **md5+des**, **sha+aes** 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
```

**Note.** The user's SNMP authentication is shown as MD5 and SNMP encryption is shown as DES.

---

### Separate Auth Key and Encryption Key for SNMPv3 User Access

The switch supports SNMPv3 users with both hashing and encryption such as SHA+DES, MD5+DES, or SHA+AES. Two different passwords are supported for a SNMPv3 user, one for switch login and another for SNMPv3 frames authentication/encryption using the **priv-password** parameter.

When the user does not specify any separate password for SNMPv3 (privilege password), user login password shall be used for SNMPv3 frame authentication. This privilege SNMP password shall be configured only for users with encryption security level; user without encryption security level will not be able to configure the privilege password. Also when the authentication level of user with separate privilege password is changed, user shall be forced to change the privilege password.

For example,

```
-> user snmpv3user password pass1pass1 priv-password priv1priv1 read-write all
sha+aes
```

The privacy password can be entered in a masked format rather than as clear text format. While creating a user, **prompt-priv-password** option can be used with the 'user' command to configure the privacy password for the user. When this option is selected, a password prompt appears and the password can be provided. Password needs to be re-entered, and only if both the passwords match, command is accepted. Password provided in this mode is not displayed on the CLI as text.

For example,

```
-> user snmpv3user password pass1pass1 prompt-priv-password sha+aes
Enter privacy password:   *********
Re-enter privacy password: *********
```

## 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<br>Non-authenticated v3 Gets and Get-Nexts<br>Authenticated v3 Sets, Gets, and Get-Nexts<br>Encrypted v3 Sets, Gets, and Get-Nexts |
| **authentication all** | Authenticated v3 Sets, Gets, and Get-Nexts<br>Encrypted v3 Sets, Gets, and Get-Nexts |
| **privacy set** | Authenticated v3 Gets and Get-Nexts<br>Encrypted v3 Sets, Gets, and Get-Nexts |
| **privacy all** | Encrypted v3 Sets, Gets, and Get-Nexts |
| **traps only** | All SNMP requests are rejected. |

# Configure SNMP Engine ID

A unique engine ID for the OmniSwitch SNMP agent can be configured.

When the SNMP agent is first initialized, the SNMP engine ID is set to the base MAC address of the switch appended to the enterprise value for OmniSwitch platforms (for example, if the enterprise value is “8000195603” and the switch base MAC address is “2c:fa:a2:13:e4:02”, then the default engine ID is set to “80001956032cfaa213e402”). During the SNMP agent configuration process, if the engine ID is configured and saved in the configuration file, then the SNMP agent engine ID will be reconfigured to match what is in the configuration file and to what the user has set it to.

SNMP agent engine ID must be a valid IPv4 address, IPv6 address, MAC address, or text. SNMP engine ID can be restored to the original value by using the ‘default’ option as shown below. The default value is reserved for MAC Address type only.

To configure a unique engine ID for the OmniSwitch SNMP agent, use the **snmp snmp-engineid-type** command. For example,

```
-> snmp snmp-engineid-type text snmp-engineid "test lab"
-> snmp snmp-engineid-type mac-address snmp-engineid 00:2a:95:01:02:03
-> snmp snmp-engineid-type ipv4-address snmp-engineid 168.22.2.2 111
```

To set the engine ID back to the default value, specify the **mac-address** parameter and the **default** parameter with this command. For example,

```
-> snmp snmp-engineid-type mac-address snmp-engineid default
```

Use the **show snmp snmp-engineid** command to view the current SNMP engine ID value for the switch.

# 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 10-10. These families correspond to switch CLI command families. When read-only privileges for a user account are restricted for a command family, that user account is also restricted from reading traps associated with that family.

Procedures for filtering traps according to command families can be found in the Quick Steps for "Filtering by Trap Families" on page 10-4. For a list of trap names, command families, and their descriptions refer to the "Using SNMP For Switch Security" on page 10-10.

### Filtering By Individual Trap

You can configure the switch to filter out individual traps by using the **snmp-trap filter-ip** command. This command allows you to suppress specified traps from the management station. The following information is needed to suppress specific traps:

- The IP address of the SNMP management station that will receive the traps.
- The ID number of the individual traps to be suppressed.

Procedures for filtering individual traps can be found in the Quick Steps for "Filtering by Individual Traps" on page 10-5. For a list of trap names, ID numbers, and their descriptions refer to the table "Using SNMP For Switch Security" on page 10-10.

# Authentication Trap

The authentication trap is sent when an SNMP authentication failure is detected. This trap is a signal to the management station that the switch received a message from an unauthorized protocol entity. This normally means that a network entity attempted an operation on the switch for which it had insufficient authorization. When the SNMP authentication trap is enabled, the switch will forward a trap to the management station. The following command will enable the authentication trap:

```
-> snmp authentication trap enable
```

The trap will be suppressed if the SNMP authentication trap is disabled.

# Trap Management

Several CLI commands allow you to control trap forwarding from the agent in the switch to the SNMP management station.

## Replaying Traps

The switch normally stores all traps that have been sent out to the SNMP management stations. You can list the last stored traps by using the **show snmp-trap replay-ip** command. This command lists the traps along with their sequence number. The sequence number is a record of the order in which the traps were previously sent out.

You may want to replay traps that have been stored on the switch for testing or troubleshooting purposes. This is useful in the event when any traps are lost in the network. To replay stored traps, use the **snmp trap replay** command followed by the IP address for an SNMP management station. This command replays (or re-sends) all stored traps from the switch to the specified management station on demand.

If you do not want to replay all of the stored traps, you can specify the sequence number from which the trap replay will start. The switch will start the replay with a trap sequence number greater than or equal to the sequence number given in the CLI command. The number of traps replayed depends on the number of traps stored for this station.

## Absorbing Traps

The switch may send the same traps to the management station many, many times. You can suppress the transmission of identical repetitive traps by issuing the **snmp-trap absorption** command. When trap absorption is enabled, traps that are identical to traps previously sent will be suppressed and therefore not forwarded to the SNMP management station. The following command will enable SNMP trap absorption:

```
-> snmp trap absorption enable
```

To view or verify the status of the Trap Absorption service, use the **show snmp-trap config** command.

## Sending Traps to WebView

When WebView forwarding is enabled, all traps sent by switch applications are also forwarded to WebView. The following command allows a WebView session to retrieve the trap history log:

```
-> snmp trap to webview enable
```

# SNMP MIB Information

## MIB Tables

You can display MIB tables and their corresponding command families by using the **show snmp mib-family** command. The MIB table identifies the MIP identification number, the MIB table name and the command family. If a command family is not valid for the entire MIB table, the command family will be displayed on a per-object basis.

For a list and description of system MIBs and Traps refer to Appendix B, “SNMP Trap Information,” in this guide.

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. |
| **snmp security tsm** | 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*.

# 11 Using OmniVista Cirrus

OmniVista Cirrus is a cloud-based network management solution used to deliver zero-touch provisioning using the cloud. The OmniVista Cirrus NMS solution provides reduced costs, ease of device provisioning and a unified wired/wireless management from the cloud. The OmniSwitch cloud management feature is configured using the OmniVista Cloud Agent.

Deployment of OmniVista Cirrus provides easier to use management and monitoring tools in a network and the ability to manage the network using devices ranging from workstations to smartphones.

## In This Chapter

This chapter provides an overview of OmniVista Cirrus and functionality, and includes information about the following procedures:


- "Quick Steps for Configuring OmniVista Cirrus" on page 11-3
- "OmniVista Cirrus Overview" on page 11-5
- "Components of OmniVista Cirrus" on page 11-5
- "DHCP Server Option 43" on page 11-8
- "Interaction with Other Features" on page 11-9
- "OmniVista Cirrus Deployment Scenarios" on page 11-10
- "Verifying the OmniVista Cirrus Configuration" on page 11-10

# OmniVista Cirrus Defaults

When OmniVista Cirrus is configured, the following default parameter values are applied unless otherwise specified:

| Parameter Description | Default Value |
|---|---|
| OmniVista Cirrus Agent Admin Status | Enabled<br>**Note:** OmniVista Cirrus Agent Admin Status is enabled by default only during RCL cases where *(vc)boot.cfg* is not present in the switch.<br>For Switch with *(vc)boot.cfg*, it needs to be enabled using CLI command. |
| OmniVista Cirrus Agent Discovery Interval | 30 minutes |
| Default location of Activation Server downloads | /flash/switch/cloud/ |
| Default URL of the Activation Server | activation.myovcloud.com:443 |

# Quick Steps for Configuring OmniVista Cirrus

The following steps provide a quick tutorial on how to configure and enable OmniVista Cirrus on an OmniSwitch.

**1** The OmniSwitch must have access to the DHCP server in the network with zero configurations on the devices. The DHCP server should be configured for the following:

- IP address
- IP subnet
- Default gateway address
- DNS server address
- Domain name (optional)
- NTP server address (Option 42)
- DHCP Vendor-Specific Options (Option 43 - VSO)

**2** When the OmniSwitch is booted up for the first time, the switch will not have a [*(vc)boot.cfg*] configuration file. Hence, OmniVista Cirrus is enabled by default.

**3** In an existing switch, which has been upgraded from a previous build and has a *vcboot.cfg*, Cloud agent has to be enabled manually. Enable the OmniVista Cirrus functionality on the switch using the **cloud-agent admin-state** command. For example:

```
-> cloud-agent admin-state enable
```

Call home can also be initiated using **cloud agent admin state restart** command and connect to OV Cirrus.

**4** Configure the time interval after which the switch will call-home the activation server, in case of any fatal error. Use the **cloud-agent discovery-interval** command. For example:

```
-> cloud-agent discovery-interval 60
```

**5** The OmniSwitch will now be connected to the OmniVista Cirrus.

---

**Note.** To verify and display the Cloud Agent status and parameters received from the DHCP and activation server, use the **show cloud-agent status** command. For example,

```
-> show cloud-agent status
Admin State                    : Enabled,
Activation Server State        : CompleteOK,
Device State                   : DeviceManaged,
Error State                    : None,
Cloud Group                    : e6a05537-4810-4231-8e3a-f903c5f86374,
DHCP Address                   : 135.254.171.88,
DHCP IP Address Mask           : 255.255.255.0,
Gateway                        : 135.254.171.1,
Activation Server              : activation.myovcloud.com:443,
NTP Server                     : 135.254.171.160,
DNS Server                     : 10.67.0.254,
DNS Domain                     : netaos.in,
Proxy Server                   : 192.168.254.49:8080,
VPN Server                     : f5f86374.tenant.vpn.8xsw.myovcloud.com:443,
Preprovision Server            : e5f86374.tenant.ovd.8xsw.myovcloud.com:80,
OV tenant                      : pingram999.ov.8xsw.myovcloud.com:443,
VPN DPD Time (sec)             : 600,
```

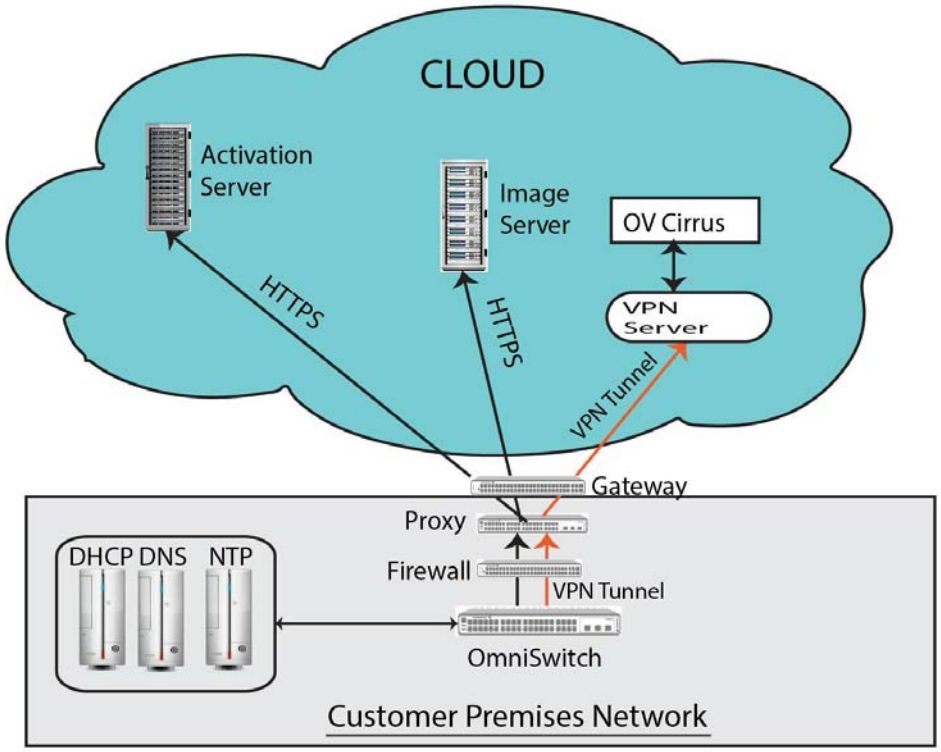
```
Image Server                    : -,
Image Download Retry Count      : -,
Discovery Interval (min)        : 30,
Time to next Call Home (sec)    : 0,
Call Home Timer Status          : Not-Running,
Discovery Retry Count           : 1,
Certificate Status              : Consistent
```

# OmniVista Cirrus Overview

The OmniVista Cloud-based solution is an alternative to the current on-premise version of OmniVista. OmniVista Cirrus Agent is a solution to deliver zero-touch provisioning using OmniVista over the cloud. The solution provides reduced costs, ease of device provisioning and a unified wired/wireless management from the cloud. The solution also provides an ability to identify each device uniquely and provide a freemium/premium solution based on the user policy.

# Components of OmniVista Cirrus

OmniSwitch interacts with the following main components in an OmniVista Cirrus topology.




The above diagram shows the deployment topology of OmniVista Cloud.

OmniVista Cirrus agent configures and enables the DNS resolver service based on the DHCP option received

OmniSwitch interacts with the following main components in an OmniVista Cirrus topology.

### DHCP Server

The DCHP Server is located at the customer network premises. The DHCP server in the network should be configured for the following.

- IP address
- IP subnet
- Default gateway address
- DNS server address
- Domain name (optional).
- NTP server address (Option 42)
- DHCP Vendor-Specific Options (Option 43)

See "DHCP Server Option 43" on page 11-8" for more information on DHCP Vendor-Specific Option 43.

### Activation Server

The Activation Server (AS) placed in the cloud environment and has to be reachable through the secure Internet router with minimal to no special configuration. The default cloud agent configuration file in the OmniSwitch (*cloudagent.cfg*) will have "activation.myovcloud.com" as the default activation server.

### OV Cirrus Instance

This is in the Cloud and is accessible through the Internet router. This connection is secure and OVCloud manages the OmniSwitch using SNMP. A secure VPN connection is used to communicate between the switch and the OV Cirrus instance.

### Proxy Server

All the communication to the Activation site and OmniVista Cirrus connects through this Proxy server. The VPN client and HTTPS client must be able to work through a Proxy in the network. The Proxy server address and port shall be obtained from the DHCP VSO. A secure VPN connection should be used to communicate between the switch and the OVCloud instance.

---

**Note.** It is not mandatory for a proxy to be present. This comes into consideration only if a proxy is present.

---

### NTP Server

Time synchronization between the devices and across the network is critical to ease communication across the network. Time synchronization helps to trace and track security issues, network usage and troubleshoot network issues.

The Network Time Protocol (NTP) helps to obtain the accurate time from a server and synchronize the local time in each network element. Connectivity to a valid NTP server is required to synchronize the OmniSwitch clock to set the correct time. If NTP server is not configured in the network, OmniSwitch reboot may lead to variation in time data.

NTP server is used to synchronize the time of VPN server and OmniVista Cirrus. NTP update is used to set time initially through NTP step mode. This is to shorten the convergence of NTP time and ensures that the device time is within the certificate validity time.

Initially, OmniVista Cirrus agent configures and enables the NTP server based on the DHCP parameters received. It will first run NTP date to set up the time in step mode. Then start the NTP client to keep synchronizing the switch time.

If NTP is not configured or present in [*(vc)boot.cfg*] or the NTP information is not available in the DHCP response, OmniSwitch will configure default NTP pool servers for use after the DNS resolution.

The four available NTP pool servers are "*clock0.ovcirrus.com*","*clock1.ovcirrus.com*", "*clock3.ovcirrus.com*" and "*clock4.ovcirrus.com*". These four NTP pool servers will be configured, if the NTP information is not received in DHCP messages and when NTP configuration is not present in switch. This newly added NTP pool servers is saved in [*(vc)boot.cfg*] in FQDN format. Each configured NTP pool servers can resolve to 2 IP address.

The **show cloud-agent status** command displays all the configured NTP servers under "NTP server".

---

**Note.** Without NTP, devices will not be able to talk to the activation server and join the cloud, unless the user manually sets the correct date.

---

For detailed information on how to configure the NTP server, see the Chapter 16, "Configuring Network Time Protocol (NTP)"

## Image Download Server

OmniSwitch downloads the AOS images from this server. The Activation server provides this URL for this server to the OmniSwitch. The switch uses HTTPS to download the images.

## VPN Server

VPN Server is a full-featured secure network tunneling VPN solution that integrates VPN server capabilities and enterprise management capabilities. This server is in the Cloud. OmniSwitch establishes the VPN connection to this server for secure communication with the OV instance. The Activation server provides VPN configuration to the OmniSwitch.

When trying to connect to the VPN server, if the connection is not established is 90 seconds, the switch will move to an error state and will call home after the expiry of the discovery interval. Once after the VPN connection is established, and if for any reason, the VPN connection is lost, the switch shall keep trying to re-connect with the VPN server. If the VPN connection cannot be re-established for a period of 10 mins, the switch shall terminate the VPN client and call home again.

To displays the Cloud Agent VPN status, use the **show cloud-agent vpn status** command.

```
-> show cloud-agent vpn status
VPN status                      : Connected,
VPN Assigned IP                 : 10.8.0.4,
VPN DPD time (sec)              : 600
```

# DHCP Server Option 43

In an OmniVista Cirrus network, A DHCP server should be configured to send the IP address along with other parameters and options. The Vendor-Specific Option Code (option 43) is one such option to be configured in the DHCP server. This information allows an OmniSwitch to automatically discover the use of Activation server for its configuration and management.

## OmniSwitch DHCP Server

The Vendor-Specific Option Code (option 43) has to be configured for the following sub-options in the *dhcpd.conf* file on the OmniSwitch DHCP server.

| Sub Options | Option Code |
|---|---|
| OXO / OV server | 1 (0x1) |
| Activation server URL | 128 (0x80) |
| Proxy server URL | 129 (0x81) |
| Proxy server Port | 130 (0x82) |
| User Name | 131 (0x83) |
| Password | 132 (0x84) |

An example of the configuration for Option 43 that needs to be added to the DHCP configuration file is:

```
option 43 1 alcatel.nms.ov2500 128 activation.dev.myovcloud.com 129
URL=192.168.254.49 130 8080 131 admin 132 password;
```

For detailed information on configuring an internal DHCP server on the OmniSwitch, see the "Configuring an Internal DHCP Server" chapter in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

**Note.** Unless prompted by customer support, there is no reason to configure an alternate Activation URL using option 43.

## Linux DHCP Server

On a Linux DHCP server, option 43 sub-options cannot be configured similar to an OmniSwitch DHCP server. Instead, the sub-options have to be configured in hexadecimal format. For example:

```
option vendor-specific
[010c616c656e7465727072697365801c61637469766174696f6e2e6465762e6d796f76636c6f756
42e636f6d];
```

- Suboption 1, length 12, value **alenterprise**
  - Suboption hex 01
  - Length hex 0c
  - Value hex 010C616c656e7465727072697365
- Suboption 128, length 28, value **activation.dev.myovcloud.com**
  - Suboption hex 80
  - Length hex 1c
  - Value hex 61637469766174696f6e2e6465762e6d796f76636c6f75642e636f6d

For more information on File Parameters and Syntax, see as "Configuration File Parameters and Syntax" section on page 23-14 in the *OmniSwitch AOS Release 8 Network Configuration Guide*.

# Interaction with Other Features

## Remote Configuration Download (RCL) and Auto Fabric

When the switch first boots up, if it does not have a *(vc)boot.cfg* config file, the switch will launch with identifying and configuring a virtual chassis using auto VC. After this, the switch initiates RCL (Auto Remote config download) to help configure the switch locally using a DHCP discovered TFTP server. After RCL is complete the switch attempts auto-fabric to auto-discover LACP, SPB and then MVRP. Once the auto-fabric discovery window is over, it will trigger OmniVista Cirrus agent.

If the switch boots up with a *(vc)boot.cfg*, the switch will skip auto VC and RCL. If auto-fabric is enabled, it will attempt to auto-discover LACP, SPB and then MVRP. Once the auto-fabric discover window is over, it will trigger OmniVista Cirrus agent.

## Virtual Chassis

Upon VC takeover by OmniSwitch, the new master starts fresh "Call-home" with the original master's synced VC Mac Address and VC serial number, and the previously received client-side certificate bundle.

Upon receipt of the certificate from the Activation server, the switch will first sync the client side certificate bundle, along with the master VC Mac Address and VC serial number, before initiating a new call home using this certificate. This will prevent a scenario where takeover occurs before the certificates are synced.

If a takeover occurs before the certificate is received, the new master will "call home" with the hash method using the VC Mac Address and VC serial number of the newly elected master.

Upon OmniSwitch VC takeover, the VPN connection will be terminated, and the new master will start again with a call-home, fetch the VPN and other parameters and then connect to the VPN server using these credentials.

## HTTP/TLS

HTTP/TLS is the secure protocol that is used for communication between the switch with the activation server and image server. The OmniSwitch first obtains its certificates from the Activation Server. All subsequent communication with the Activation server or OV is secured using this certificate. The VPN client and HTTPS/TLS client will work through a proxy in the network. The proxy address and port are obtained from the DHCP VSO. In this way, a secure VPN connection is established and used to communicate between the switch and the OmniVista Cirrus instance.

# Dependencies

- The switch will initiate a call-home after every reboot if there is no configuration file on the switch.
- If there is a configuration file on the switch, the switch will initiate a call-home only if the cloud agent enabled explicitly using the **cloud-agent admin-state** command in the configuration. Enabling cloud agent using this command will immediately initiate a call-home sequence with the activation server.

- If the call-home sequence is already in progress or in a connected state, the CLI will display a warning "*Switch is already connected/connecting to OV Cloud. Please `write memory` to save the configuration*". Use the **write memory** command ` to save the configuration.

# OmniVista Cirrus Deployment Scenarios

The deployment scenarios of ALE devices are as follows:

**Greenfield deployments:** In this scenario, ALE switches/APs that at are deployed for the first time with Freemium or Non-Freemium OVCloud service.

**Brownfield deployments 1:** In this scenario, the network consists of an existing operational network of third-party devices, ALE switches, and APs. To this operational network, the customer adds ALE switches/APs with Freemium or Non-Freemium OVCloud service. Only the newly added devices are using the OVCloud service.

**Brownfield deployments 2**: In this scenario, the network consists of an existing operational network of third-party devices, ALE switches, and AP. To this operational network, the customer adds the OVCloud management service to manage the existing network. The existing configuration of the customer should not be overwritten when moving to the cloud unless explicitly changed from the cloud.

# Verifying the OmniVista Cirrus Configuration

To display information about OmniVista Cirrus on the switch, use the show commands listed below:

| | |
|---|---|
| **show cloud-agent status** | Displays the Cloud Agent status and parameters received from the DHCP and activation server. |
| **show cloud-agent vpn status** | Displays the Cloud Agent VPN status. |

# 12 Web Services, CLI Scripting, OpenFlow, and AOS Micro Services (AMS)

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.

## In This Chapter

This chapter contains the following information:


- "Web Services Overview" on page 12-2
- "Web Services REST Examples" on page 12-5
- "Using Python" on page 12-15
- "CLI Scripting" on page 12-20
- "Embedded Python Scripting" on page 12-25
- "AOS Micro Services (AMS)" on page 12-27
- "OpenFlow Agent Overview" on page 12-31
- "Quick Steps to Configure OpenFlow Agent" on page 12-33

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

**Domain—**This 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 dependent 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 an HTTP standard diagnostic code:<br>• A 2xx value if the command was successful; in most cases '200' will be used.<br>• A 3xx value if a resources was moved (not implemented).<br>• A 4xx value if the request contained an error; e.g. '400' in case of failed authentication.<br>• 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 |
|---|---|
| {"result":{<br>"domain":"auth (login)",<br>"diag":200,<br>"error":"",<br>"output":"",<br>"data":[]}} | <?xml version="1.0" encoding="UTF-8" ?><br><nodes><br><result><br><domain>auth (login)</domain><br><diag>200</diag><br><error></error><br><output></output><br><data></data><br></result><br></nodes> |

**Example Error Response**

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

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

## 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?<br>mibObject0=vlanNumber:2&mibObject1=vlanDescription:VLAN-2 |

**Example Success Response**

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

**Example Error Response**

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

## 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?<br>mibObject1=alaIpItfConfigName:my_new_interface2&mibObject0=alaIpItfConfigIfIndex:0<br><br>POST Request: [https://192.168.1.1/mib/alaIpInterfaceTable?]<br>mibObject1=alaIpInterfaceAddress:2.1.1.1&mibObject0=ifIndex:13600002&mibObject3=alaIpInterfaceVlanID:1&mibObject2=alaIpInterfaceMask:255.255.255.0 |

**Example Success Response**

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

**Example Error Response**

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

## 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?<br>mibObject1=vlanNumber:2&mibObject0=vlanDescription:vlan-Two |

**Example Success Response**

**JSON**

```
{"result":{
   "domain":"mib:vlanTable",
   "diag":200,
   "output":"",
   "error":[
      "Set operation finished
successfully!"],
   "data":[]}}
```

**XML**

```
<?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>
```

## 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?<br>mibObject0=esmPortCfgSpeed:1000&mibObject1=ifIndex:1001&mibObject2=esmPortCfgDuplexMode1 |

**Example Success Response**

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

## 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?<br>mibObject1=vlanNumber:2 |

**Example Success Response**

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

**Example Error Response**

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

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

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

**Example Error Response**

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

# 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())
```

# PYTON 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** | Invoke this method to perform a "show" query or run a CLI command.<br><br>**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';<br><br>**urn** - represents the "address" of the entity being accessd: 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'<br><br>**args** - is a dictionary of key->value pairs where each respective key's name if '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. |
| **put(domain, urn, args)** | Invoke this method to create a new object. It is not a valid command when using the 'cli' domain.<br><br>**domain** - is the same as described in the 'query()' section, except 'cli' is not supported.<br><br>**urn** - is the same as described in the 'query()' section, except no cli-related value is supported.<br><br>**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; |

| | |
|---|---|
| **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.<br>default value: True |
| **obeyproxy** | When True, system proxy settings will be followed.<br>default value: True |
| **prettylinks** | When True, use semantically correct links as opposed to '?a=b&c=d…'<br>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.<br>default value: -1 |
| **headers** | An object used to inject additional headers in the request if necessary.<br>default value: None |
| **debug** | When True, low-level GET, POST, PUT and DELETE commands will be displayed in the current terminal.<br>default value: False |

## AOSHeaders (Dict config)

| | |
|---|---|
| **config** | A a dictionary that contains the current configuration:<br><br>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;<br><br>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 behaviour, 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 repetively 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 successfuly 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
```

# Embedded Python Scripting

The OmniSwitch includes many standard Python packages to access AOS and system functions. This feature allows administrators to create Python scripts and associate these scripts with specific traps. When the traps are generated by the switch, the pre-configured scripts will be run on the switch. This provides the capability to adapt to a dynamically changing network and customize how the switch should react to those changes. There are multiple ways to execute Python on the switch:

- Automatically, as an event-action when a trap occurs
- Interactively, from the console
- In a script file executed by command from the console

AOS Python includes many standard Python packages for:

- OS access and issuing AOS commands
- Sending email and database access.

## Guidelines

- Scripts can only be created by administrators with write privileges to the partition management family AAA.
- Event-based scripts must be stored in the **/flash/python** directory.
- The **show snmp-trap config** command can be used to see list of traps on the switch.
- An event can have only one script assigned to it, but a script can be assigned to multiple events.

## Assigning Events

To assign a switch event to a script use the **event-action** command, for example:

```
-> event-action trap linkDown script /flash/python/link_event.py

-> event-action trap stpNewRoot script stp_event.py
```

**Note.** Use the **interfaces link-trap** command to enable traps for link up/down events.

## View the Events

To view statistics such as how many times a script has been run use the **show event-action** command, for example:

```
-> show event-action

type              name                    script  (/flash/python/…)
------+----------------------------------+-----------------------------
trap    linkDown                            link_event.py
trap    stpNewRoot                          stp_event.py
trap    sessionAuthenticationTrap           catchAll.py
```

```
-> show event-action statistics
                                                  Script             Launch
Type                    Name                      Last Launched      Count
------+-----------------------------------------+-------------------+----------
trap   linkDown                                   2014-10-23 13:45:34      2
```

# Python Examples in AOS

To following is a simple interactive example of how AOS can be used to execute Python commands.

```
-> python3
Python 3.2.2 (default, Dec 10 2014, 02:41:47)
[GCC 4.8.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("Hello, World from AOS-python\n")
Hello, World from AOS-python
>>> quit()
->
```

To following is an example of how AOS can be used to execute a Python script named **sample.py** that is stored on the switch.

```
-> cat sample.py
# sample python command file
for i in [1, 2, 3, 4]:
        print("Hello, World " + str(i))
-> python3 sample.py
Hello, World 1
Hello, World 2
Hello, World 3
Hello, World 4
->
```

To following is an example of how AOS can be used to execute a Python script named **import_sample.py** that is stored on the switch and uses the imported subprocess and os libraries.

```
-> cat import_sample.py
#!/bin/python3
import os
import subprocess
result = subprocess.check_output(["show","microcode"],universal_newlines=True)
print ("----Subprocess Output----")
print (result)
print ("----OS Output----")
os.system("show microcode")
->
-> python3 import_sample.py
----Subprocess Output----
   /flash/working
   Package           Release                 Size     Description
-----------------+-------------------------+--------+--------------------------
Tos.img           7.3.4.314.R01             210517932 Alcatel-Lucent OS

----OS Output----
   /flash/working
   Package           Release                 Size     Description
-----------------+-------------------------+--------+--------------------------
Tos.img           7.3.4.314.R01             210517932 Alcatel-Lucent OS
```

# AOS Micro Services (AMS)

AOS Micro services (AMS) is a network programmability application ecosystem which provides an asynchronous mechanism for communication and information synchronization across a community of OmniSwitches. The mechanism requires the presence of a broker to relay the messages across OmniSwitches. The role of broker is played by OmniVista or by an OmniSwitch if OmniVista is not present in a network.

AMS uses publish-subscribe messaging as the underlying protocol for communication among switches. All OmniSwitches can act as subscribers or clients. A client firstly needs to establish a connection with the broker (another OmniSwitch) and subscribe to a topic to which it wishes to listen or advertise to. Any logical grouping can be used to create a topic of interest, for example, all access switches, all OmniSwitch 6560 switches in a network or all switches in a particular location.

As an example, AMS provides an application of synchronizing the OS6465 power supply configuration information. In this example all OS6465 switches will subscribe to a topic pertaining to the OS6465 power supply configuration and other non-OS6465s would not. Once a power supply is configured on a OS6465, the same information can be shared with the community of all OS6465 switches using AMS. The broker switch can be any of the OmniSwitches, although it's recommended to use an OS6860, OS6900 or OS9900 as a broker. Any number of topics can be created for communication across switches

AMS provides an environment for ALE bundled or 3rd party (customer developed) applications to be loaded on the switches outside of AOS and which can utilize this mechanism for sharing information across switches. IoT device profiling agent and OS6465 power supply configuration synchronization are some of the applications that have been developed and bundled with AOS to showcase the AMS application environment. The bundled applications can be installed, uninstalled, started and stopped without impacting AOS in any way. AMS also provides a "configuration replay" functionality (when the broker is an OmniSwitch) where any new switch joining the community gets a replay of the previous messages on the topics (for live configuration) to which the new switch is subscribing to. The AMS application infrastructure also provides an environment for hosting user developed applications and the mechanisms for synchronization of information across the community of switches.

## AMS Components

**Broker** - AMS uses a publish / subscribe mechanism which requires a Broker to relay all the messages in the network. A Broker IP needs to be configured in all the switches for the AM infrastructure to work.

**Topics** - The relation of publisher and subscriber is associated with TOPIC names. When a client registers with Broker, it can provide set of Topics it wants to register with the Broker. Subsequently when any client publishes a message on one of these Topics, all the other clients would get the message if they have registered for this specific Topic. Topic is a string name and hierarchical.

**Community** - A group of switches participating in AMS. Community is part of the Topic hierarchical definition of COMMUNITY_NAME/APPLICATION/APPLICATION_SUB_CONFIG. For example, ALE_ACCESS/DEVICE_PROFILING/DEVICE_PROFILING_SNMP_TABLE_NAME. If a group of switches subscribes to the Topic COMMUNITY_NAME/# then all Topics under that community will be sent to those subscribers. If a group of switches subscribes to Topic COMMUNITY_NAME/ APPLICATION/# then all Topics associated with that application will be sent to the subscribers.

**Config DB** - A component residing on OmniSwitch Broker. Config-DB maintains a record of all configuration between the subscriber switch and Broker switch. When a new switch joins the community it's responsible for replaying all the earlier for a topic to that new switch. The corresponding application needs to be started only on Broker

**Config-sync** - An application running on all switches responsible for formatting messages received from Broker for consumption on local switch. Corresponding application needs to run on all switches.

# AMS Applications

| | |
|---|---|
| **Device Profiling Agent** | A bundled application with AMS, this application provides the reporting & synchronization of IoT device signatures in a network. When used with OmniVista, this application provides the endpoint information on any new IoT device connecting on that switch to OmniVista. In non-OmniVista environment, this application helps to synchronize across the network, device signatures of any new IoT device connecting to the switch. |
| **OS6465 Power Supply Configuration Synchronization** | A bundled application with AMS, this application helps to synchronize the power supply configuration for OS6465-P6 & OS6465-P12 switches in a network in cases where the same power supplies are being used across the network. The power supply needs to be configured on one switch first before initiating the synchronization. |

# AMS Configuration

The following configuration files are used by AMS. All configuration files are available in **/flash/running-directory/pkg/ams** directory.

| | |
|---|---|
| **ams-broker.cfg** | The configuration file for clients to connect to the broker. The broker IP can be updated as part of "-h" option in the file.<br><br>For example:<br>-h 10.135.82.43 |
| **config-sync.cfg** | The config-sync configuration file provides configuration options to start the config-sync. Some of the user configurable parameters are as follows:<br>- **community**: The community that the client belongs to.<br>- **topics**: The list of topics and SNMP table mapping that the client subscribes to synchronize configuration in.<br><br>For example:<br>"topics": {<br>"alaDpDevicesTable": "DP_DEVICES_TABLE",<br>"alaDpGlobalConfig":"DP_GLOBAL_CONFIG"<br>}<br><br>The above example provides the configuration for Device Profiling signatures (i.e. SNMP table alaDpDevicesTable) on a switch to be synchronized to other switches in the community. |

## Use Case Example - Device Profiling Signature Synchronization

A currently supported use case for the AMS framework is to synchronize device profiling signatures and configuration between switches. The following device profiling tasks are supported:

- Device Profiling Global Configuration Synchronization
- Device Profiling Signature Synchronization

As an example, there will be cases where an unknown device is reclassified as known by providing the device type and device name along with a signature. Once a single device instance is reclassified, all the subsequent instances of similar devices would be automatically identified and the UNP profile updated.

To replicate this behavior on other switches of the network an administrator would need to configure the device signature on each individual switch. But by leveraging AMS, the device signature can be automatically synchronized throughout the network. By updating the device signature on one switch on the network, the same signature is updated on all the switches which are part of AMS framework.

### Use Case Example - OmniSwitch 6465 Power Supply Configuration Synchronization

Power supply configuration commands for the OmniSwitch 6465 can be synchronized. For example:

```
-> powersupply 1 name ps1 type ale hi-ac chassis-id 1
```

## Enabling AMS

Before enabling AMS, the broker configuration file must be edited on all switches and the IP address of the broker modified to be that of the broker switch. The *ams-broker.cfg* file can be found in the **/flash/*running-directory*/pkg/ams** folder. Change the default IP address to that of the broker switch.

1. Start the broker switch and required agents:

```
-> appmgr start ams broker - (Starts the broker)
-> appmgr start ams config-dbase - (Starts the config-dbase application)
-> appmgr start ams config-sync - (Starts the config-sync application)
-> appmgr commit - (Saves the appmgr settings across reboots)
```

2. Start all the subscriber switches:

```
-> appmgr start ams config-sync - (Starts the config-sync applicaton)
-> appmgr commit - (Saves the appmgr settings between reboots)
```

3. Display the running modules on the broker switch:

```
-> appmgr list
Legend: (+) indicates application is not saved across reboot

  Application    Status    Package Name        User/Group            Status Time Stamp
---------------+---------+-------------------+---------------------+---------------------
  broker        started   ams                 admin/user            May 01, 2019: 09:18:25
  config-dbase  started   ams                 admin/user            May 01, 2019: 15:16:32
  config-sync   started   ams                 admin/user            May 01, 2019: 15:16:42
```

4. Display the running modules on the subscriber switches:

```
-> appmgr list
Legend: (+) indicates application is not saved across reboot

  Application    Status    Package Name        User/Group            Status Time Stamp
---------------+---------+-------------------+---------------------+---------------------
  config-sync   started   ams                 admin/user            Jun 22, 2014: 18:52:19
```

### Device Profile Example

**1** Create a new device profile on either the broker or a subscriber switch:

```
Switch1-> device-profile admin-state enable
Switch1-> device-profile device-type new-device-type device-name new-device-name
from dhcp-option-55 1,3,6
```

**2** Display the device profile configuration on the other switches. Device profiling will be enabled and the new device profile will be automatically created by AMS:

```
-> show device-profile signatures

Device Type                      Device Name                    DHCP Option 55
---------------------+------------------------------+------------------------
ip-cam                  netcam                    1,3,6,15
SmartPhone/PDA/Tablets  Apple iPad                1,3,6,15,119,252
IP-Phone                Gigaset A580 VoIP         1,3,6,120,125,114
Printer                 Kyocera Network Printer   1,3,12,23,6,15,44,47
SmartPhone/PDA/Tablets  Motorola                  1,121,33,3,6,28,51,58,59
Windows                 Windows XP                1,15,3,6,44,46,47,31,33,249,43
Printer                 SAMSUNG Network  1,3,6,7,12,15,18,23,26,44,46,51,54,58,59,78,79,81
*new-device-type        new-device-name           1,3,6

---------------------------------
Number of Signatures: 8 grp1
```

## OmniVista Device Profiling Interface

An additional use case of this new model is between device profiling and OmniVista. This model can be used to communicate between AOS switches and OmniVista using the client/broker relationship to provide various information related to endpoints attached to the switches. The switch can provide end-point MAC detection and deletion events, DHCP/DNS/HTTP user-agent information and packet-type information that is collected for the end-point to the device profiling application running on OmniVista.

---

**Note:** This capability will be available in a future version of OmniVista **.**

---

# 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 switch. Traditionally, switches 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 switch itself, and all control decisions are communicated to the switch 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 firewall, pass the packets to a specific port to perform load balancing, prioritize packets, etc).

The OmniSwitch 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 OmniSwitch for OpenFlow support.

## OpenFlow Logical

An OpenFlow logical switch consists of a portion of the switch's resources that are managed by an OpenFlow Controller (or set of Controllers) via the OpenFlow Agent. Logical switches can be configured on an OmniSwitch, with each logical switch supporting separate controllers. A logical switch 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 switch operates as per the OpenFlow standards.

### OpenFlow Hybrid (API) Mode

In Hybrid mode, the logical switch 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 switch operates on all ports, link aggregates, and VLANs not assigned to other OpenFlow logical switches.

### Support OpenFlow Parameters

In following OpenFlow tables, match fields, groups and actions are supported.

#### Flow Definitions

- Exact Match
- Wildcard
- MAC Table

#### Match Fields

- Ingress Port
- Ethernet Destination Address

- Ethernet Source Address
- VLAN Tag / VLAN Priority
- Ethernet Type
- IPv4 or IPv6 Protocol Number
- IPv4 Source Address / IPv4 Destination Address
- TCP / UDP Source & Destination Ports
- ICMP Type / Code
- ARP Operation

## Groups

Groups are a way of combining a set of activities into one action. For example, a Group could be used to represent an IP next hop with all of the associated activities (MAC change, VLAN update, etc.). The collection of actions is stored in a bucket. Each group includes a collection of buckets and the different types identify policies on how to select which bucket(s) to use.

- ALL - The actions of all buckets are executed. This will be used to implement broadcast or multicast activities. Packet modification actions are not supported by this type of group.
- INDIRECT - This is an ALL type group with a single bucket. Packet modification actions are supported by this type of group.

## Actions Fields

- Output - To physical, reserved or linkagg port
- Drop - Drop the packet
- Group - Process packets according to specified group
- Set Field - Set fields in the packet (only for single egress port). VLAN priority can only be set for tagged packets.

# 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 OmniSwitch. A logical switch in Hybrid mode does not have a VLAN or interface configured.

**1** Create the logical switch and configure the mode:

```
-> openflow logical-switch vswitch1 mode normal version 1.3.1 vlan 5
-> openflow logical-switch vswitch2 mode api
```

**2** Assign a controller to the logical switch:

```
-> openflow logical-switch vswitch1 controller 1.1.1.1
-> openflow logical-switch vswitch2 controller 2.2.2.2
```

**3** Assign interfaces to the logical switch:

```
-> openflow logical-switch vswitch1 interfaces port 1/1/3
```

**4** Verify the configuration

```
-> show openflow logical-switch
                          Admin
Logical Switch            State  Mode    Versions    VLAN   Ctrlrs   Intf   Flows
--------------------------+-----+------+------------+------+--------+------+-------
vswitch1                   Ena   Norm   1.3.1            5        1      1     5
vswitch2                   Ena   API    1.0 1.3.1       N/A       1     56     0

-> show openflow logical-switch controllers
                                                                    Admin    Oper
Logical Switch                  Controller               Role       State    State
----------------------------+----------------------+----------+-----+-----------
vswitch1                      1.1.1.1:6633           Equal      Ena   Connect
vswitch2                      2.2.2.2:6633           Equal      Ena   Backoff

-> show openflow logical-switch interfaces
         Logical Switch          Interface     Mode
-------------------------------+--------------+------
vswitch1                        1/1/3            Norm
vswitch2                        1/1/1            API
vswitch2                        1/1/2            API
vswitch2                        1/1/4            API
vswitch2                        1/1/5            API
vswitch2                        1/1/6            API
vswitch2                        1/1/7            API
(output truncated)
```

# 13 Configuring Virtual Chassis

A Virtual Chassis is a group of switches managed through a single management IP address that operates as a single bridge and router. It provides both node level and link level redundancy for layer 2 and layer 3 services and protocols acting as a single device. The use of a virtual chassis provides node level redundancy without the need to use redundancy protocols such as STP and VRRP between the edge and the aggregation/core layer.

The following are some key points regarding a virtual chassis configuration:

- With the introduction of the Virtual Chassis feature a switch can now operate in two modes; Virtual Chassis or Standalone.
- When a switch operates in Virtual Chassis this will cause a change to the CLI requiring a chassis identifier to be used and displayed for some commands such as interfaces or ports.
- A Virtual Chassis provides a single management IP address for a group of switches that are acting as a single bridge or router.
- The switches participating in a Virtual Chassis are created by inter-connecting them via standard single or aggregated interfaces.

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 them 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 Default Values" on page 13-3
- "Quick Steps for Configuring A Virtual Chassis" on page 13-4
- "Virtual Chassis Overview" on page 13-6
- "Virtual Chassis Topologies" on page 13-14
- "Interaction with Other Features" on page 13-16
- "Virtual Chassis Configuration Example" on page 13-23
- "Automatically Setting up a Virtual Chassis" on page 13-28
- "Displaying Virtual Chassis Configuration and Status" on page 13-38
- "Automatic Virtual Chassis Flow" on page 13-33
- "Virtual Chassis Split Protection (VCSP)" on page 13-35


See Chapter 1, "Getting Started and Upgrading AOS," for licensing information and getting started with this feature.

# Virtual Chassis Default Values

The table below lists default values for Virtual Chassis.

| Parameter Description | Command | Default Value/Comments |
|---|---|---|
| Chassis Identifier | **virtual-chassis configured-chassis-id** | 0 |
| Chassis group identifier | **virtual-chassis chassis-group** | Derived from last byte of Master chassis MAC address |
| Chassis priority | **virtual-chassis configured-chassis-priority** | OS6900-Q32/X72 - 120<br>All Others - 100 |
| Hello-interval | **virtual-chassis hello-interval** | 10 seconds |
| Control VLAN | **virtual-chassis configured-control-vlan** | 4094 |
| Default VLAN virtual-fabric link | N/A | 1 |
| VFL Mode | **virtual-chassis vf-link-mode** | Auto |
| Default auto-VFL ports | **virtual-chassis auto-vf-link-port** | OS6900-X and T models - The last 5 ports of each chassis, including expansion slots. Ports without a transceiver present are included when determining VFL port eligibility.<br><br>OS6900-Q32 - The last 5 ports (28-32). A port that has a 40G-to-10G splitter cable will be counted as four ports.<br><br>OS6900-X72 - The last 5 ports (50-54). A port that has a 40G-to-10G splitter cable will be counted as four ports.<br><br>OS6900-V72/C32 - Static VFL only.<br><br>OS6860 - Dedicated VFL ports.<br><br>OS6865 - None.<br><br>OS6560 - Dedicated VFL ports and last two 10G SFP+ ports on (P)24X4/(P)48X4.<br><br>OS9900 - Static VFL only.<br><br>OS6465-P6/P12 - None.<br>OS6465-P28 - Ports 27/28. |

# Quick Steps for Configuring A Virtual Chassis

Follow the steps below for a quick tutorial on configuring two switches to operate as a Virtual Chassis. Additional information on how to configure a Virtual Chassis is provided in the section "Configuring Virtual Chassis" on page 13-17.

A switch running in standalone mode can be converted to a Virtual Chassis participant by using the CLI to create the required ***vcsetup.cfg*** and ***vcboot.cfg*** files. The virtual chassis configuration can be kept in any directory, including the ***/flash/working*** directory. The example below uses a new directory named ***vc_dir***.

The minimum configuration requirements for a switch to operate in virtual chassis mode are:

- A vcsetup.cfg file must be present in the running directory and it must have, at least, a valid chassis identifier and virtual-fabric link configured. Additionally for the switch to join a virtual chassis topology, the virtual-fabric link must have member ports assigned to it and the underlying interfaces administratively enabled.
- A vcboot.cfg file containing the generic virtual chassis configuration present must be present in the running directory.

**Note**: Multi-Chassis Link Aggregation cannot be configured on a switch prior to converting it to virtual chassis mode. The MC-LAG configuration must first be removed.

## Virtual Chassis Conversion

**1** Enter the following on each switch using the appropriate port numbers based on the chassis physical connections. Note that ***vc_dir*** is the name of the directory where the virtual chassis configuration will be stored, it will be automatically created if it does not exist.

```
Chassis_1-> virtual-chassis configured-chassis-id 1
Chassis_1-> virtual-chassis vf-link 0 create
Chassis_1-> virtual-chassis vf-link 0 member-port 1/1
Chassis_1-> virtual-chassis vf-link 0 member-port 1/24
Chassis_1-> write memory
Chassis_1-> convert-configuration to vc_dir

Chassis_2-> virtual-chassis configured-chassis-id 2
Chassis_2-> virtual-chassis vf-link 0 create
Chassis_2-> virtual-chassis vf-link 0 member-port 1/1
Chassis_2-> virtual-chassis vf-link 0 member-port 1/24
Chassis_2-> write memory
Chassis_2-> convert-configuration to vc_dir
```

**2** Reload both chassis from the newly created ***vc_dir*** directory:

```
Chassis_1-> reload from vc_dir no rollback-timeout
Chassis_2-> reload from vc_dir no rollback-timeout
```

## 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
                                        Config
 Chas  Role         Status              Chas ID  Pri   Group  MAC-Address
-----+------------+-------------------+--------+-----+------+-----------------
 1     Master       Running              1        100   0     00:e0:b1:e7:09:a3
 2     Slave        Running              2        100   0     00:e0:b1:e7:09:a4
```

**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

       Config                          Oper     Config   Oper    Config
       Chas   Chas             Chas    Control  Control  Hello   Hello
 Chas* ID     Type    License* Group*  Vlan*    Vlan     Interv* Interv  Status
------+------+-------+--------+------+--------+--------+-------+-------+--------
 1     1       OS6900  0x3      0       4094     4094      5       10       OK
 2     2       OS6900  0x3      0       4094     4094      5       10       OK
```

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

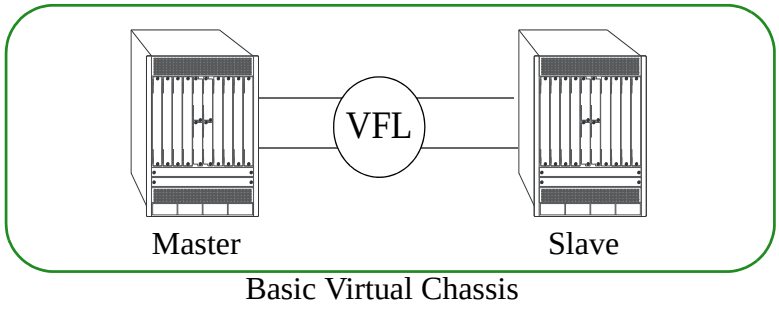
```
-> show virtual-chassis vf-link member-port
 Chassis/VFLink ID  Chassis/Slot/Port  Oper       Is Primary
-------------------+------------------+----------+-------------
 1/0                1/1/1              Up         Yes
 1/0                1/1/24             Up         No
 2/0                2/1/1              Up         Yes
 2/0                2/1/24             Up         No
```

# Virtual Chassis Overview

Virtual Chassis is a group of switches 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
- Active-Active multi-homed link aggregation
- Provides predictable and consistent convergence with redundant links to the two switches
- 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 Basic Topology**

## 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 switches 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.

**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 via the virtual-fabric links to determine its configuration.

**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 in a switch operating in virtual chassis mode. It is recommended to have both the EMP-VC IP address and the Chassis EMP IP address configured.

**Chassis EMP Address—**The local chassis management IP address (EMP-CHAS1 or EMP-CHAS2). 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 switch specific *vcsetup.cfg* configuration file in a switch operating in virtual chassis mode.

**Virtual Fabric Link (VFL)**—A single or aggregated group of ports that connects the switches 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.

**Remote Chassis Detection (RCD) protocol**—Provides a back up mechanism for helping to detect a split-chassis scenario.

**IS-IS VC**—Proprietary protocol for managing a Virtual Chassis mesh 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, chassis EMP IP addresses and VFL links.

**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. Similar to the boot.cfg file used in standalone mode. The vcboot.cfg file is only used when a switch operates in virtual chassis mode.

## Converting to Virtual Chassis Mode

In order for a switch to become part of a virtual chassis it must first be converted from a standalone switch. Virtual chassis operation requires the two files below to be created. They can be created manually or automatically using the **convert-configuration** command.

- **vcsetup.cfg**—Virtual chassis setup file used to incorporate the physical chassis into the virtual chassis topology.
- **vcboot.cfg**—Virtual chassis configuration file.

Before converting a standalone switch's configuration keep the following in mind:

- The switches to be converted cannot have multi-chassis link aggregation configured. A switch operating in multi-chassis link aggregation mode must be reconfigured to operate in standalone mode and rebooted before the conversion to a virtual chassis can be automatically accomplished via the steps described here. An alternative conversion from multi-chassis link aggregation mode to virtual chassis is always possible manually. This can be achieved by manually creating both *vcsetup.cfg* and *vcboot.cfg* files offline in the appropriate running directory and rebooting the switches.

## Converting Chassis Mode Using the CLI

The following shows an example of how to convert two switches that are in standalone mode to virtual chassis mode.

- The VFL member ports configuration should reflect the switch's current physical connections.
- The directory ***vc_dir*** can be any directory, including the ***working*** directory. By creating a separate directory specifically for virtual chassis operation the existing ***working*** directory is not affected.

```
Chassis #1
Chassis-> virtual-chassis configured-chassis-id 1
Chassis-> virtual-chassis vf-link 0 create
Chassis-> virtual-chassis vf-link 0 member-port 1/1
Chassis-> virtual-chassis vf-link 0 member-port 1/24
Chassis-> write memory
Chassis-> convert-configuration to vc_dir
Chassis-> reload from vc_dir no rollback-timeout

Chassis #2
Chassis-> virtual-chassis configured-chassis-id 2
Chassis-> virtual-chassis vf-link 0 create
Chassis-> virtual-chassis vf-link 0 member-port 1/1
Chassis-> virtual-chassis vf-link 0 member-port 1/24
Chassis-> write memory
Chassis-> convert-configuration to vc_dir
Chassis-> reload from vc_dir no rollback-timeout
```

**Note.** It is recommended that the switches be rebooted at approximately the same time.

### Conversion Process

**1** A directory with the name ***vc_dir*** will be created if it does not exist.

**2** If a current standalone configuration (e.g. boot.cfg) exists it will not be affected.

**3** The ***vcsetup.cfg*** and ***vcboot.cfg*** files will be automatically created within ***vc_dir*** directory.

**4** The images from the current running directory will be automatically copied to the ***vc_dir*** directory. If different image files are to be used they should be manually copied after the convert configuration command has been executed and prior to the reload command.

### Reboot Process

When the switches come up after the reload command, here is what will happen:

**1** The ***vcboot.cfg*** and image files must be the same on all switches running in virtual chassis mode. As a result, if there is a mismatch between the Master and Slave ***vcboot.cfg*** or images files, the Master will overwrite the files on the Slave chassis and the Slave will automatically reboot.

**2** The original configuration of the Slaves will be overwritten and must be reapplied if necessary once the Virtual Chassis is up and stabilized.

### New "chassis/slot/port" Syntax

Once the switches are operating in virtual chassis mode all commands that relate to specific ports or NI modules must have a leading chassis identifier to differentiate between the physical ports on each switch as seen in the example below.

| Standalone Mode | Virtual Chassis Mode |
|---|---|
| interfaces 1/1 admin-state enabled | interfaces 1/1/1 admin-state enabled (chassis 1) |
| interfaces 1/1 admin-state enabled | interfaces 2/1/1 admin-state enabled (chassis 2) |

## 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 initial connection over a VFL to all the other neighboring switches.

**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 moves to start up error mode if the *vcsetup.cfg* file is corrupted or edited in such a way that it is unable to read a valid chassis identifier in the appropriate range.

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 (e.g. out of range) chassis identifier.
- **Invalid-License (no longer applicable)**: The chassis is not operational in virtual chassis mode because no valid Advanced license has been found.

### License Behavior

The following table describes the behavior of a Slave chassis when attempting to join a VC based on the installed licenses. A Slave chassis must have the proper license(s) when attempting to join an existing VC. Based on the type of license installed on the Master, the Slave chassis may fail to join the existing VC or may inherit the existing licenses.

| Existing Master | New Slave | Join? | Inherit License | Note |
|---|---|---|---|---|
| DC | No license | Yes | DC | Slave inherits DC license from Master |
| DC | DC | Yes | None | |

**License Table Matrix**

### 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 the Slave chassis will takeover the Master role and all traffic flows that are based on the multi-homed physical connections will reconverge on the new Master.
- If the Slave chassis goes down the Master chassis will retain its Master role and all traffic flows that are based on multi-homed physical connections will reconverge on the existing Master.
- If the VFL goes down, the Master chassis will retain its Master role. The Slave chassis will transition to assume the Master role as well. At this point the virtual chassis topology has been split and there will be two Masters in the network. If a management EMP network has been configured the Remote Chassis Detection (RCD) protocol will detect this split topology. In response to this event, the former Slave chassis will shutdown all its front-panel user ports to prevent duplicate IP and chassis MAC addresses in the network. The Slave's chassis status will be modified from *Running* to *Split-Topology* to indicate this second pseudo-master chassis is not operational at this point. If the VFL comes back up, the former Slave chassis will reboot and rejoin the virtual chassis topology assuming its Slave role again.
- If the primary CMM on the Master chassis fails the secondary CMM, if available, will takeover and the chassis will remain the Master chassis.

- If all CMMs on the Master chassis fail the chassis will reboot and the first-in-line Slave chassis will take over becoming the new Master chassis. The first-in-line is derived from the same election criteria that were used to select the original Master.

## Split Chassis Detection

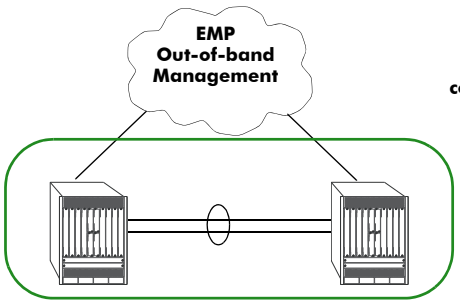
Split chassis detection is implemented using a proprietary protocol called RCD (Remote Chassis Detection) protocol. The goal of the split-chassis detection mechanism is to provide information in a virtual chassis environment which can be used to determine whether a VFL has failed. A split chassis can occur when the VFL connection is broken but each of the switches remains operational. This scenario must be detected so that one of the switches remains the Master and continues using the same IP and MAC address in the network.

**Note.** RCD is only enabled once the virtual chassis is operational. If a switch is unable to join a virtual for any reason, the RCD protocol will not be enabled.

To help detect this scenario each switch in the Virtual Chassis topology periodically sends information via its local EMP port. All of the switches participating in a Virtual Chassis should be able to communicate via the local EMP port using an out-of-band network. When a VFL goes down, each switch can still communicate with the others via the EMP port, this acts as a backup mechanism to help detect the split chassis scenario. RCD will use the following IP addresses in order of preference:

1. CMM IP address stored in NVRAM (if configured)
2. Chassis EMP IP address

See the "Configuring EMP IP Addresses" on page 13-21 for information on configuring the EMP IP addresses. Also, see the "Split Chassis Detection - Chassis-based CMMs" on page 13-12 for information on EMP communication between CMMs.




**Each chassis in the Virtual Chassis sends periodic updates via the EMP port. Each chassis should be able to communicate with the other via the EMP out-of-band network.**

Having both switches with all the front-panel ports up while the VFL is down can cause layer 2 and layer 3 connectivity issues. In order to avoid this scenario an out-of-band management protocol has been implemented on the EMP port that detects the status of each chassis. If all VFL links go down then this protocol will detect and shutdown all user ports on the former Slave chassis to prevent the duplicate IP and MAC addresses from being used on the network. The user ports will automatically come up when the VFL connectivity is re-established.

**Note. If more than one Virtual Chassis is part of the same EMP out-of-band management network then each Virtual Chassis MUST have a unique chassis-group ID. Otherwise the RCD protocol cannot differentiate between the two Virtual Chassis and will not operate correctly.**

### Split Chassis Detection - Chassis-based CMMs

Directly connecting the EMP ports of the CMMs on the Slave and Master switches is not a recommended method for detecting a split chassis scenario. Using directly connected CMM EMP ports could result in a scenario where the Primary CMM on one switch is directly connected to the Secondary CMM on the other switch if a local CMM takeover occurred on one of the switches. Since the RCD protocol is only active on the Primary CMM, this would result in a loss of RCD communication.

## Virtual Chassis Split Protection (VCSP)

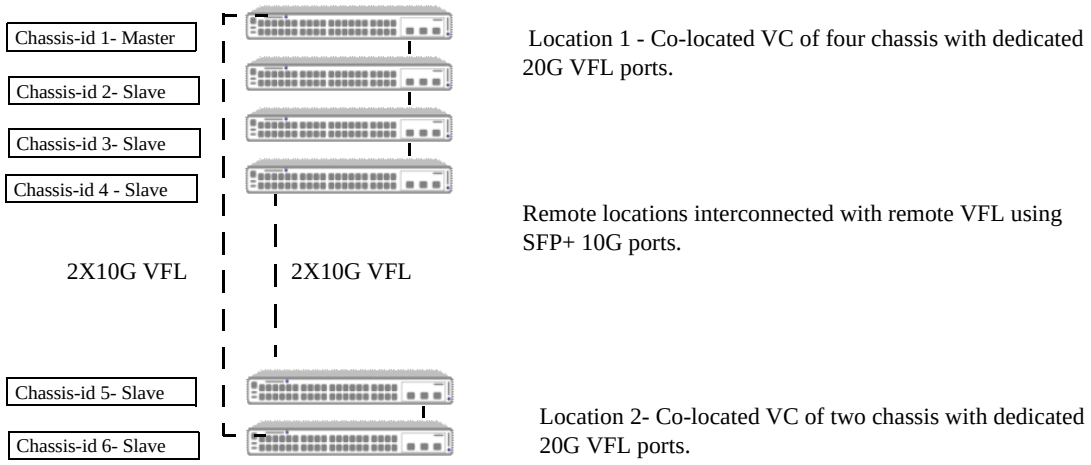
Virtual chassis split protection is implemented using the proprietary VCSP protocol. The goal of the VCSP mechanism is to provide information in a virtual chassis environment which can be used to determine whether a VFL has failed and resulted in a split VC. A split VC can occur when one or multiple 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 13-35 for information on configuring VCSP.

## Remote Virtual Chassis (Remote Stacking)

Long distance VFL connections can be configured on the 10G SFP+ ports to extend the capability of the virtual chassis to remote locations. This is achieved by configuring the 10G SFP+ user ports as auto-VFL ports in addition to the 20G dedicated VFL ports.

See the "Automatic VFL" on page 13-29 for information on configuring a port as an auto VFL port.




**Remote VC Example**

## Virtual Chassis Topology Change Notification

The VC topology is saved and synchronized across the VC when the **write memory** command is issued. During a reboot or takeover scenario the Master will compare the current VC topology with the previous saved topology. An SNMP trap will be sent if an element of the VC has been removed or added. Additionally, a trap will be sent if a VC element is added or removed during runtime after the **write memory** command is issued.

| Scenario | Description | Trap |
|---|---|---|
| VC Takeover | Any element including the previous Master does not join after takeover. | Trap will not be sent after the new Master reaches ready state. |
| | Any element including the previous Master rejoins after takeover. | Trap will be sent by the new Master. |
| Element added | New element added to the VC. | Trap will be sent. |
| Element removed | Element removed from the VC. | Trap will be sent. |

### VC Topology Change Notification - Confirmation

After executing the **write memory** command, if any one of the VC elements is down the configuration for that element will be lost. When the **write memory** command is issued the current VC topology will be compared against the saved VC topology and if there is any difference then a warning will be issued about possible configuration purge and ask for confirmation from the user to proceed. If the user confirms, the existing configuration for the element which is down will not be saved. If the user does not confirm then the write memory operation will not proceed. This will ensure that configurations will not be lost without notification to the user.

# Virtual Chassis - Upgrading

See “Upgrading the Software” on page 1-5.

# Virtual Chassis Topologies

This section describes the building blocks that are used to construct more flexible network topology using virtual chassis feature. Some example topologies for virtual chassis are given below. For more information on virtual chassis topologies, refer to the following sections.

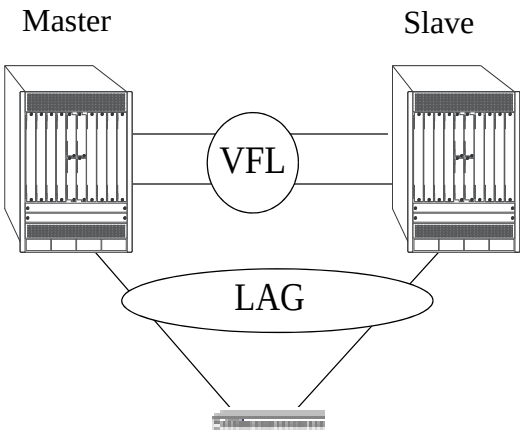
• "Basic Virtual Chassis Building Block" on page 13-14

• "Recommended Topologies" on page 13-14

• "Interaction with Other Features" on page 13-16

## Basic Virtual Chassis Building Block

The building block below can be used to connect to the edge or core devices in the network and is comprised of two switches connected with a virtual fabric link (VFL).




**Virtual Chassis Building Block**

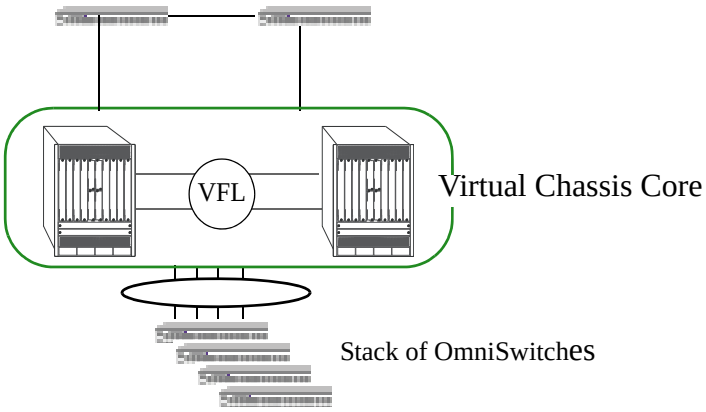
## Recommended Topologies

The following topologies are recommended to support the virtual chassis functionality:

- Virtual Chassis in a Campus Core
- Virtual Chassis in a Data Center

### Campus Core

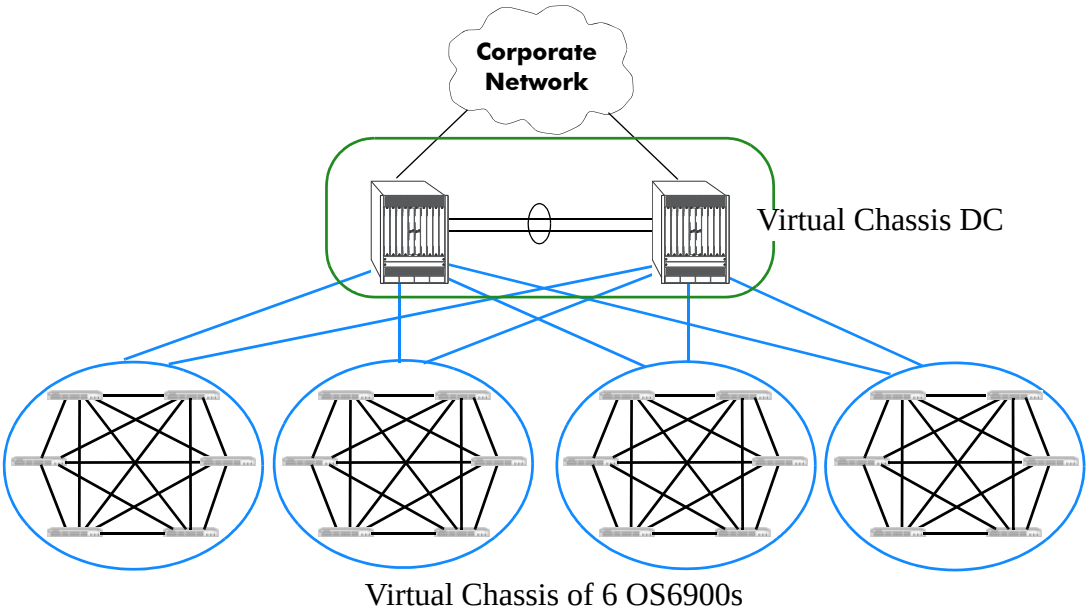
In the topology shown below, all edge devices are attached to both virtual chassis peers at the core. Spanning Tree is not needed in this network because there are no loops. In this topology, the physical loop around the virtual chassis ports and Virtual Fabric Link is prevented.




**Virtual Chassis at the Core**

## Data Center VC

In the topology shown below, edge switches are connected through virtual chassis and core switches are dual attached.




**Data Center VC**

# 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.

## Multicast Load Balancing

IP Multicast traffic load balancing may not be optimized on VFL member ports that belong to the same port group as listed in the table above. To ensure IP Multicast traffic load balancing is optimized over the VFL, only one VFL member port should be included per port group.

## QoS

It is recommended to use only QSP1 (strict priority) if configuring QSP on a VFL.

## VCSP

If a VC is split, configuration changes on the split switch will not take affect until the switch is rebooted.

# Configuring Virtual Chassis

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


- "Virtual Chassis Configuration Guidelines" on page 13-17
- "Configuring the Chassis Identifier" on page 13-19
- "Configuring the Virtual Chassis Group Identifier" on page 13-20
- "Creating the Virtual Fabric Link (VFL)" on page 13-20
- "Configuring the Hello Interval" on page 13-20
- "Configuring the Control VLAN" on page 13-21
- "Configuring EMP IP Addresses" on page 13-21


**Note.** See "Quick Steps for Configuring A Virtual Chassis" on page 13-4" 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 vary depending on whether they are executed while a switch is operating in standalone mode (conversion process) or virtual chassis mode (runtime configuration). The following guidelines focus on the initial configuration, when a switch is still operating in standalone mode. For a thorough description of the configuration process while a switch is already operating in virtual chassis mode, please refer to the CLI guide.

### General

- Virtual chassis functionality is only active for switches on which a valid chassis identifier is configured.
- One of the chassis will become the Master chassis while the remaining switches will become Slaves.
- A virtual chassis cannot contain a mix of different families of switches (i.e OS6900 and OS6860).
- 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.
- When a new chassis is added to an existing virtual chassis the new chassis will reboot two times under any of the following conditions:
  - The new chassis has a different running configuration directory name than the existing VC.
  - The new chassis has different images than the existing VC.
  - The new chassis has a different vcboot.cfg file than the existing VC.

## Chassis Identifier

- Each switch requires a chassis identifier that is unique within the virtual chassis group of topology.
- 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 should be corrected by accessing the switch directly via the local EMP port.
- The chassis identifier is used to generate globally unique values for the module identifiers as well as allowing inter-chassis communication.
- A switch 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 13-19.

## Virtual Chassis Group Identifier

- Each switch also requires a virtual chassis group identifier to identify the switch as belonging to that specific virtual chassis topology.
- When determining the chassis group ID the last byte of the Master chassis MAC address is used. For example, if the Master's MAC address is xx:xx:xx:xx:xx:7e, the chassis group will be 126 (the decimal equivalent to hexadecimal 7e).
- The same group identifier must be assigned to each switch in the virtual chassis topology. Switches belonging to other virtual chassis groups must use a different group identifier.
- If two or more switches 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 should be corrected by accessing the switch directly via local EMP port.
- 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. This configuration may cause problems for the RCD (Remote Chassis Detection) protocol used to detect virtual chassis topology splits as well as other unpredictable issues.
- When communicating between VCs the Master chassis MAC address is used.

For information about configuring the chassis group identifier, see "Configuring the Virtual Chassis Group Identifier" on page 13-20.

## Virtual Fabric Link (VFL)

- Individual protocols such as SFlow, ERP, UDLD and LLDP are not supported on VFLs and must not be configured on ports belonging to a VFL. This situation may occur if a previous configuration, such as MC-LAG, is converted to a Virtual Chassis configuration. It is highly recommended to review the configuration carefully and make the necessary changes particularly when converting from MC-LAG to VC.
- An operational VFL is a basic requirement to support a fully functional virtual chassis.
- The Link Aggregation Control Protocol (LACP) is used to mange and monitor the state of the VFL.
- Explicitly configuring the VFL and the physical port members is required. It's recommended to configure the VFL during network maintenance time or when the virtual chassis is first configured.

Changing the VFL configuration at runtime is supported but should be performed with caution as an incorrect VFL configuration can cause undesirable disruption to traffic flows.

- It is recommended to configure the VFL at the same time as the chassis identifier. This ensures that the switch reboots with the correct VFL configuration.
- For increased resiliency, there should be a minimum of two member ports and they should be distributed across different port groups and NI modules.
- Only physical ports operating at 10-Gbps (not including 10GBaseT), 40-Gbps, or 100-Gbps can be members of a VFL. Additionally, 10-Gbps and 40-Gbps links cannot be mixed in the same VFL. Any type of 10Gbps or 40-Gbps transceiver or direct-attached cable can be used for creating the VFL. 10GBase-T ports cannot be members of a VFL.
- The member ports configured as part of the VFL are bundled to form a single logical link. All the member ports must operate at the same speed.
- To help avoid a split chassis scenario the last active VFL member port cannot be deleted or disabled. Additionally, the last NI module hosting the last active member port cannot be administratively powered down or reloaded.
- The VFL automatically becomes a member of all VLANs configured on the switch.
- VFL member ports can only be configured on interfaces that are fixed ports, network ports or PFC enabled ports. For instance, interfaces configured as Q-tag ports or ERP ports cannot be configured as VFL member ports.
- The hello interval parameter must match between switches. The hello protocol runs across the VFL between the switches.
- Some user-data traffic loss may be seen on VFL link when sending at wire rate. Since all packets that traverse the VFL have an additional 16 byte header prepended to the packet this reduces the effective bandwidth of a given VFL port.

For more information on Virtual Fabric Link, see “Creating the Virtual Fabric Link (VFL)” on page 13-20.

### Control VLAN

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

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 configured-chassis-id 1
```

By default, the chassis identifier is set to “0”. This indicates the switch is running in standalone mode, which means that no virtual chassis functionality is available.

### Duplicate Chassis Identifier

In the event two switches 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 switches to be automatically renumbered to fall into the range (101-102). This range is reserved to represent switches 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 via EMP port access.

## 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 switch 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.

## Creating the Virtual Fabric Link (VFL)

The VFL is an aggregate of high-speed ports used for inter-chassis user traffic and control data. For a virtual chassis group to become operational, a VFL must be configured and brought to an operational state.

To configure a VFL and its member ports, use the **virtual-chassis vf-link create** and **virtual-chassis vf-link member-port** commands. For example:

```
-> virtual-chassis vf-link 0 create
-> virtual-chassis vf-link 0 member-port 1/1
-> virtual-chassis vf-link 0 member-port 1/24
```

## 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.

It is recommended that the same hello interval be used for all switches that will participate on the same virtual chassis topology. Failure to adhere to this recommendation will lead the switches whose values depart from the master chassis' settings to assume the *Inconsistent* role and *Misconfigured-Hello-Interval* status.

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 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 primary CMM of 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 or EMP-CHAS2). This address can be used for remote access to each switch comprising the virtual chassis. This address is automatically assigned to the primary CMM of 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.
- It is recommended to have both the EMP-VC IP address and the Chassis EMP IP address configured.

## Configuring the Chassis EMP IP Address - Standalone Mode

Use the **ip interface** command to modify the Chassis EMP IP address as shown below. These commands would be issued prior to the execution of the **convert configuration** command.

```
Chassis1-> ip interface local emp address 10.255.100.1 mask 255.255.255.0
Chassis2-> ip interface local emp address 10.255.100.2 mask 255.255.255.0
```

## Configuring the Chassis EMP IP Address - Virtual Chassis Mode

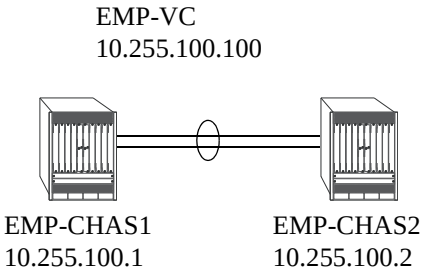
Use the **ip interface** command to modify the Chassis EMP IP address as shown below. These commands would be issued after the virtual chassis is operational:

```
-> 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 - Virtual Chassis Mode

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

The Primary or Secondary's CMM's 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 a chassis-based switch the IP addresses are named as follows and are associated to each CMM on each chassis.

- EMP-CMMA-CHAS1
- EMP-CMMB-CHAS1
- EMP-CMMA-CHAS2
- EMP-CMMB-CHAS2

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:

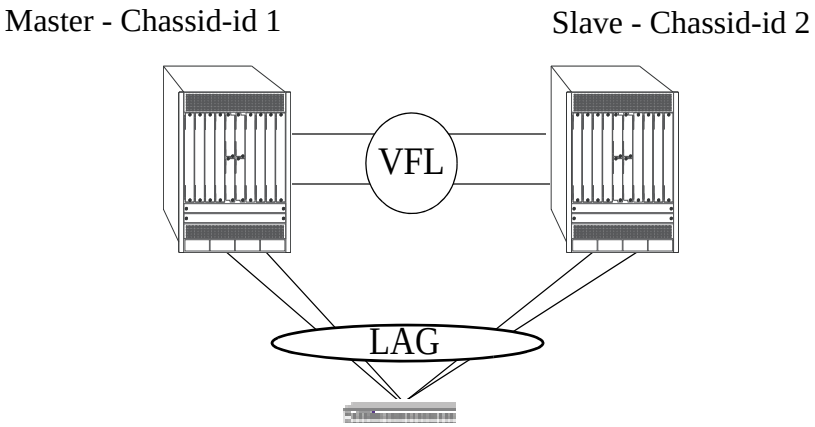
```
-> modify boot parameters
Boot > boot empipaddr 255.255.100.50
Boot > boot empmasklength 16
Boot > commit system
Boot > commit
```

## 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.

# Virtual Chassis Configuration Example

This section provides an example of virtual chassis configuration in a network.




**VC Example**

## Virtual Chassis Configuration

```
Chassis_1-> virtual-chassis configured-chassis-id 1
Chassis_1-> virtual-chassis vf-link 0 create
Chassis_1-> virtual-chassis vf-link 0 member-port 1/24-25
Chassis_1-> ip interface local emp address 10.255.100.1 mask 255.255.255.0
Chassis_1-> write memory
Chassis_1-> convert-configuration to vc_dir

Chassis_2-> virtual-chassis configured-chassis-id 2
Chassis_2-> virtual-chassis vf-link 0 create
Chassis_2-> virtual-chassis vf-link 0 member-port 1/24-25
Chassis_1-> ip interface local emp address 10.255.100.2 mask 255.255.255.0
Chassis_2-> write memory
Chassis_2-> convert-configuration to vc_dir

Chassis_1-> reload from vc_dir no rollback-timeout
Chassis_2-> reload from vc_dir no rollback-timeout
```

## Virtual Chassis EMP IP Address Configuration

Once the virtual chassis group is operational, the rest of the configuration is carried out on the Master Chassis. The step below is critical because it defines an IP address that will be used to manage the entire virtual chassis.

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

## VLAN Configuration

Now that the virtual chassis group is operational, the rest of the configuration is carried out on the Master chassis.

```
VC_Core-> vlan 100
VC_Core-> vlan 200
VC_Core-> ip interface vlan-100 address 100.100.100.1/24 vlan 100
VC_Core-> ip interface vlan-200 address 200.200.200.1/24 vlan 200
```

### Link Aggregation Configuration

```
VC_Core-> linkagg lacp agg 1 size 4 admin-state enable
VC_Core-> linkagg lacp agg 1 actor admin-key 1
VC_Core-> linkagg lacp port 1/1/10 actor admin-key 1
VC_Core-> linkagg lacp port 1/1/11 actor admin-key 1
VC_Core-> linkagg lacp port 2/1/10 actor admin-key 1
VC_Core-> linkagg lacp port 2/1/11 actor admin-key 1
VC_Core-> vlan 100 members linkagg 1 untagged
VC_Core-> vlan 200 members linkagg 1 tagged
```

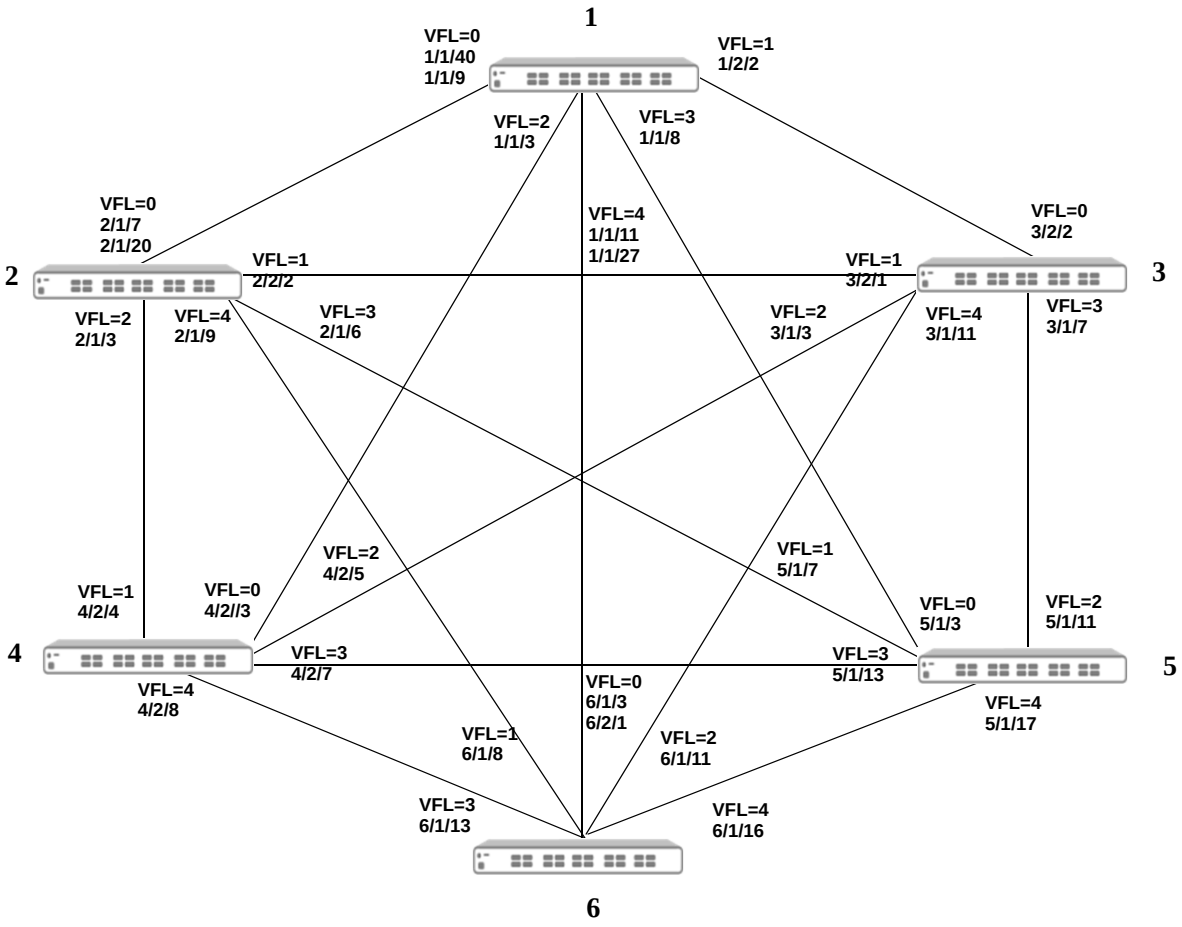
### Verify VC Configuration

```
VC_Core-> show virtual-chassis topology
VC_Core-> show virtual-chassis consistency
VC_Core-> show virtual-chassis vf-link member-port
```

### SW1 Configuration

```
SW1-> linkagg lacp agg 1 size 4 admin-state enable
SW1-> linkagg lacp agg 1 actor admin-key 1
SW1-> linkagg lacp port 1/1-4 actor admin-key 1
SW1-> vlan 100 members linkagg 1 untagged
SW1-> vlan 200 members linkagg 1 tagged
```

# Virtual Chassis Mesh VFL Configuration Example




**Virtual Chassis Mesh**

## Virtual Chassis of 6 VFL Configuration Example

```
Chassis_1-> virtual-chassis configured-chassis-id 1
Chassis_1-> virtual-chassis vf-link 0 create
Chassis_1-> virtual-chassis vf-link 0 member-port 1/40, 1/9
Chassis_1-> virtual-chassis vf-link 1 create
Chassis_1-> virtual-chassis vf-link 1 member-port 2/2
Chassis_1-> virtual-chassis vf-link 2 create
Chassis_1-> virtual-chassis vf-link 2 member-port 1/3
Chassis_1-> virtual-chassis vf-link 3 create
Chassis_1-> virtual-chassis vf-link 3 member-port 1/8
Chassis_1-> virtual-chassis vf-link 4 create
Chassis_1-> virtual-chassis vf-link 4 member-port 1/11, 1/27
Chassis_1-> ip interface local emp address 10.255.100.1 mask 255.255.255.0
Chassis_1-> write memory
Chassis_1-> convert-configuration to vc_dir

Chassis_2-> virtual-chassis configured-chassis-id 2
Chassis_2-> virtual-chassis vf-link 0 create
Chassis_2-> virtual-chassis vf-link 0 member-port 1/7, 1/20
```

```
Chassis_2-> virtual-chassis vf-link 1 create
Chassis_2-> virtual-chassis vf-link 1 member-port 2/2
Chassis_2-> virtual-chassis vf-link 2 create
Chassis_2-> virtual-chassis vf-link 2 member-port 1/3
Chassis_2-> virtual-chassis vf-link 3 create
Chassis_2-> virtual-chassis vf-link 3 member-port 1/6
Chassis_2-> virtual-chassis vf-link 4 create
Chassis_2-> virtual-chassis vf-link 4 member-port 1/9
Chassis_2-> ip interface local emp address 10.255.100.2 mask 255.255.255.0
Chassis_2-> write memory
Chassis_2-> convert-configuration to vc_dir

Chassis_3-> virtual-chassis configured-chassis-id 3
Chassis_3-> virtual-chassis vf-link 0 create
Chassis_3-> virtual-chassis vf-link 0 member-port 2/2
Chassis_3-> virtual-chassis vf-link 1 create
Chassis_3-> virtual-chassis vf-link 1 member-port 2/1
Chassis_3-> virtual-chassis vf-link 2 create
Chassis_3-> virtual-chassis vf-link 2 member-port 1/3
Chassis_3-> virtual-chassis vf-link 3 create
Chassis_3-> virtual-chassis vf-link 3 member-port 1/7
Chassis_3-> virtual-chassis vf-link 4 create
Chassis_3-> virtual-chassis vf-link 4 member-port 1/11
Chassis_3-> ip interface local emp address 10.255.100.3 mask 255.255.255.0
Chassis_3-> write memory
Chassis_3-> convert-configuration to vc_dir

Chassis_4-> virtual-chassis configured-chassis-id 4
Chassis_4-> virtual-chassis vf-link 0 create
Chassis_4-> virtual-chassis vf-link 0 member-port 2/3
Chassis_4-> virtual-chassis vf-link 1 create
Chassis_4-> virtual-chassis vf-link 1 member-port 2/4
Chassis_4-> virtual-chassis vf-link 2 create
Chassis_4-> virtual-chassis vf-link 2 member-port 2/5
Chassis_4-> virtual-chassis vf-link 3 create
Chassis_4-> virtual-chassis vf-link 3 member-port 2/7
Chassis_4-> virtual-chassis vf-link 4 create
Chassis_4-> virtual-chassis vf-link 4 member-port 2/8
Chassis_4-> ip interface local emp address 10.255.100.4 mask 255.255.255.0
Chassis_4-> write memory
Chassis_4-> convert-configuration to vc_dir

Chassis_5-> virtual-chassis configured-chassis-id 5
Chassis_5-> virtual-chassis vf-link 0 create
Chassis_5-> virtual-chassis vf-link 0 member-port 1/3
Chassis_5-> virtual-chassis vf-link 1 create
Chassis_5-> virtual-chassis vf-link 1 member-port 1/7
Chassis_5-> virtual-chassis vf-link 2 create
Chassis_5-> virtual-chassis vf-link 2 member-port 1/11
Chassis_5-> virtual-chassis vf-link 3 create
Chassis_5-> virtual-chassis vf-link 3 member-port 1/13
Chassis_5-> virtual-chassis vf-link 4 create
Chassis_5-> virtual-chassis vf-link 4 member-port 1/17
Chassis_5-> ip interface local emp address 10.255.100.5 mask 255.255.255.0
Chassis_5-> write memory
Chassis_5-> convert-configuration to vc_dir

Chassis_6-> virtual-chassis configured-chassis-id 6
Chassis_6-> virtual-chassis vf-link 0 create
```

```
Chassis_6-> virtual-chassis vf-link 0 member-port 1/3, 2/1
Chassis_6-> virtual-chassis vf-link 1 create
Chassis_6-> virtual-chassis vf-link 1 member-port 1/8
Chassis_6-> virtual-chassis vf-link 2 create
Chassis_6-> virtual-chassis vf-link 2 member-port 1/11
Chassis_6-> virtual-chassis vf-link 3 create
Chassis_6-> virtual-chassis vf-link 3 member-port 1/13
Chassis_6-> virtual-chassis vf-link 4 create
Chassis_6-> virtual-chassis vf-link 4 member-port 1/16
Chassis_6-> ip interface local emp address 10.255.100.6 mask 255.255.255.0
Chassis_6-> write memory
Chassis_6-> convert-configuration to vc_dir
```

# Automatically Setting up a Virtual Chassis

Automatic Virtual Chassis can be used to ease the required manual configuration for a VC. The automatic VC feature will allow a brand new chassis shipped from the factory or a chassis with no configuration to be setup as a VC without user configuration.

There are two main components with the automatic Virtual Chassis feature:

- Automatic configuration of VFL IDs and ports
- Automatic chassis ID assignment

Benefits of automatic Virtual Chassis.

- Existing switches configured in standalone mode will be unchanged and remain in standalone mode.
- Existing switches configured as part of a VC will be unchanged and remain as part of an existing VC.
- Newly shipped switches or switches with no configuration will default to automatic VC mode and the automatic VC feature will run.

## Automatic Virtual Chassis Concepts and Components

Automatic VC can be used to ease a VC setup. The automatic VC feature will allow a brand new chassis shipped from the factory or a chassis without a configuration to be setup as part of a VC without user configuration.

**VFL Mode—**A chassis can operate in either automatic VFL mode or static VFL mode but not both at the same time. Static VFL mode is what has been supported in previous releases where the VFL is configured by explicitly creating VFL IDs and specifying its member ports. In automatic VFL mode the user specifies ports that are designated as automatic VFL ports, or uses the default set of automatic VFL ports, and the software will automatically assign VFL IDs.

**Automatic VFL port**—A port that is eligible to participate in the automatic VFL process.

**Automatic Chassis ID Assignment—**Automatic chassis ID assignment is used to automatically configure a chassis ID.

## VFL Mode

A VC can operate in either automatic or static VFL mode. In static VFL mode VFLs are configured by explicitly creating VFL IDs and specifying member ports. In automatic VFL mode ports only have to be designated as automatic VFL ports and the system will automatically assign VFL IDs, chassis IDs, and aggregate the VFL member ports if possible.

The **virtual-chassis vf-link-mode** command is used to modify the VFL mode. It is a global configuration that applies to all chassis in the VC.

- If the chassis boots without **vcsetup.cfg** file, by default the chassis is in automatic VFL mode.
- If the **vcsetup.cfg** file exists but the VFL mode configuration **virtual-chassis vf-link-mode {static | auto}** is not in the vcsetup.cfg file, the chassis boots up in static VFL mode. This scenario would apply to a chassis that is being upgraded from a previous release that doesn't support automatic VFL. release).
- Chassis must have the same VFL mode to form a VC.

- An “out-of-the-box” chassis or a chassis with no configuration file will default to automatic VFL mode. For this chassis to automatically join an existing VC, the existing VC must be in VFL automatic VFL mode. If the existing VC is not in automatic VFL mode it can be converted to automatic mode or the new chassis can be changed to static mode in order to join the existing VC.
- The VFL mode of a VC can be changed at runtime without a reboot. This is global configuration change and applies to all the chassis in the VC.

# Automatic VFL

Automatic VFL detection process will run to automatically configure the VFL ports on a VC. The process is only run on ports that are eligible to be an automatic VFL port. Automatic VFL has the following guidelines:

- Automatic VFL ports must be a 10Gbps or 40Gbps port.
- Automatically detects whether an automatic VFL port should become a VFL member port.
- Dynamically assigns a VFL ID to an automatic VFL port which becomes a VFL member port.
- Aggregates multiple VFL member ports that are connected to the same remote chassis.
- Uses a default set of ports that are eligible to be automatic VFL ports. See “Virtual Chassis Default Values” on page 13-3.

The automatic VFL process is run under the following conditions:

- Chassis boots without a **vcsetup.cfg** or **vcboot.cfg** file. Since there is no configuration, the default set of ports will be used to run the automatic VFL process.
  - If the speed of the transceiver in the set of the default ports is not 10G or 40G that port cannot become a VFL member port.
  - If the media type of the port in the set of default ports is copper, that port cannot become a VFL member port.
  - Once the automatic VFL discover period ends, any ports that have not been configured as VFL member ports will become regular front panel ports.
- The chassis boots with a **vcsetup.cfg** file. Then the automatic VFL process is run only on those ports explicitly configured as auto VFL ports.

## Configuring Automatic VFL Ports

To configure a port to become an automatic VFL port use the **virtual-chassis auto-vf-link-port** command. This allows a port to participate in the automatic VFL process.

## Converting the VFL Mode

The VFL mode of a VC can be changed at runtime using the **virtual-chassis vf-link-mode** command. This is a global configuration change that applies to all chassis in the VC. This change does not require a reboot.

### Converting Static to Automatic

After issuing the **virtual-chassis vf-link-mode auto** command the VFL mode is converted from static to automatic. All existing VFLs will be converted to automatic VFL ports regardless of whether the links are active or not.

For example, below is the current configuration with VFL mode as static:

```
virtual-chassis chassis-id 1 configured-chassis-id 1
virtual-chassis vf-link-mode static
virtual-chassis chassis-id 1 vf-link 0 create
virtual-chassis chassis-id 1 vf-link 0 member-port 1/1/21
virtual-chassis chassis-id 1 vf-link 0 member-port 1/1/22
virtual-chassis chassis-id 1 vf-link 1 create
virtual-chassis chassis-id 1 vf-link 1 member-port 1/1/23
virtual-chassis chassis-id 1 vf-link 1 member-port 1/1/24 (assume link is down
on this port)
```

After issuing the **virtual-chassis vf-link-mode auto** command the VFL mode is converted from static to automatic. All the existing VFLs are converted to automatic VFL ports regardless of whether the links are active or not.

For example, below is the new configuration after the mode is converted to automatic:

```
virtual-chassis chassis-id 1 configured-chassis-id 1
virtual-chassis vf-link-mode auto
virtual-chassis auto-vf-link-port 1/1/21
virtual-chassis auto-vf-link-port 1/1/22
virtual-chassis auto-vf-link-port 1/1/23
virtual-chassis auto-vf-link-port 1/1/24
```

Please note the following:

- Although port 1/1/24 is down, that port is still converted to an automatic VFL port.
- Although it is not shown in the new configuration ports 1/1/21 and 1/1/22 are still member ports of VFL 0 and ports 1/1/23 and 1/1/24 are still member ports of VFL 1 (as long as there is no topology change).
- Use the **write memory** command to save the new configuration.

## Converting Automatic to Static

After issuing the **virtual-chassis vf-link-mode static** command the VFL mode is converted from automatic to static. All existing VFLs will be converted to static VFL ports if they are active at the time of conversion.

Please note the following assumptions:

- Ports 1/1/21 and 1/1/22 have become VFL ports and belong to VFL ID 1
- Ports 1/1/23 and 1/1/24 have become VFL ports and belong to VFL ID 0
- Ports 1/1/25 has not become a VFL port yet (i.e. link is down)

For example, below is the current configuration with VFL mode as auto:

```
virtual-chassis chassis-id 1 configured-chassis-id 1
virtual-chassis vf-link-mode auto
virtual-chassis auto-vf-link-port 1/1/21
virtual-chassis auto-vf-link-port 1/1/22
virtual-chassis auto-vf-link-port 1/1/23
virtual-chassis auto-vf-link-port 1/1/24
virtual-chassis auto-vf-link-port 1/1/25 (this port has not become VFL, i.e.
link is down)
```

After issuing the **virtual-chassis vf-link-mode static** command the VFL mode is converted from auto to static. All auto VFL ports that have become VFL member ports are converted to static VFL with their current VFL IDs.

For example, below is the new configuration after the mode is converted to static:

```
virtual-chassis chassis-id 1 configured-chassis-id 1
virtual-chassis vf-link-mode static
virtual-chassis chassis-id 1 vf-link 0 create
virtual-chassis chassis-id 1 vf-link 0 member-port 1/1/23
virtual-chassis chassis-id 1 vf-link 0 member-port 1/1/24
virtual-chassis chassis-id 1 vf-link 1 create
virtual-chassis chassis-id 1 vf-link 1 member-port 1/1/21
virtual-chassis chassis-id 1 vf-link 1 member-port 1/1/22
```

Please note the following:

- There is no entry for port 1/1/25 in the new configuration because at the time of the conversion, port 1/1/25 was not part of a VFL and there is no way to know which VFL ID this port belongs to.
- Use the **write memory** command to save the new configuration.

## Automatic Chassis ID Assignment

As part of the automatic VC feature, each chassis will also automatically be assigned a chassis ID. Automatic chassis ID assignment happens when a chassis boots up without **vcsetup.cfg.** This is the case for a new 'out-of-the-box' chassis. On its first bootup, a **vcsetup.cfg** file will be created and a chassis ID will be assigned and stored in the newly created **vcsetup.cfg** file**.** On subsequent reboots, this chassis will use the chassis ID that was configured in **vcsetup.cfg**. As long as the **vcsetup.cfg** file exists, automatic chassis ID assignment will not be attempted.

When an out-of-the-box chassis boots up, it will have temporary chassis ID of 1 and a special flag indicating that this chassis needs a chassis ID assigned. After VC discovery process is completed, master election process will take place and a master chassis will be elected based on the master election parameters (chassis priority, uptime, chassis-id, and chassis MAC address).

Master will assign each chassis in the VC (including itself) a unique chassis ID based on the chassis ID assignment algorithm. Each chassis will store its newly assigned chassis ID by writing it in vcsetup.cfg file, so this chassis id can be used in subsequent reboots. After receiving its newly assigned chassis id, each slave chassis will reboot for the new chassis id to take effect. Master will not reboot.

For Master election, chassis with configured chassis id will always win over chassis with temporary chassis id (has no vcsetup.cfg).

# Automatic Virtual Chassis Scenarios

## Boot up with no vcsetup.cfg file

**1** Since the chassis has no configuration it will begin the automatic VFL process by default.

**2** The chassis will create a new vcsetup.cfg file and temporarily use chassis ID 1 while running the discovery protocol on the default set of automatic VFL ports.

**3** The chassis will communicate with its peers that are also running the VFL protocol to determine which ports will become VFL member ports and what the VFL IDs will be. Multiple ports connected to the same peer chassis will be aggregated and assigned the same VFL ID. Ports connected to different chassis will be assigned different VFL IDs.

**4** Once the VFLs are configured a Master chassis will be elected using the Master chassis election criteria.

**5** Once a Master chassis is identified the Slave chassis will be assigned unique chassis IDs using the automatic chassis ID assignment procedure.

**6** The chassis ID, automatic VFL mode, and automatic VFL port information will be written to the vcsetup.cfg file. For the default set of automatic VFL ports that did not become VFL member ports, no configuration information will be saved and those ports will no longer be automatic VFL ports.

**7** All Slave chassis will be rebooted and rejoin the VC.

## Bootup with vcsetup.cfg file and automatic VFL Mode Enabled

**1** Since the chassis has a configuration and automatic VFL is enabled it will begin the automatic VFL process.

**2** The chassis will use the configured chassis ID while running the discovery protocol on the configured set of automatic VFL ports.

**3** The chassis will communicate with its peers that are also running the VFL protocol to determine which ports will become VFL member ports and what the VFL IDs will be. Multiple ports connected to the same peer chassis will be aggregated and assigned the same VFL ID. Ports connected to different chassis will be assigned different VFL IDs.

**4** Once the VFLs are configured a Master chassis will be elected using the Master chassis election criteria and the VC will become active.

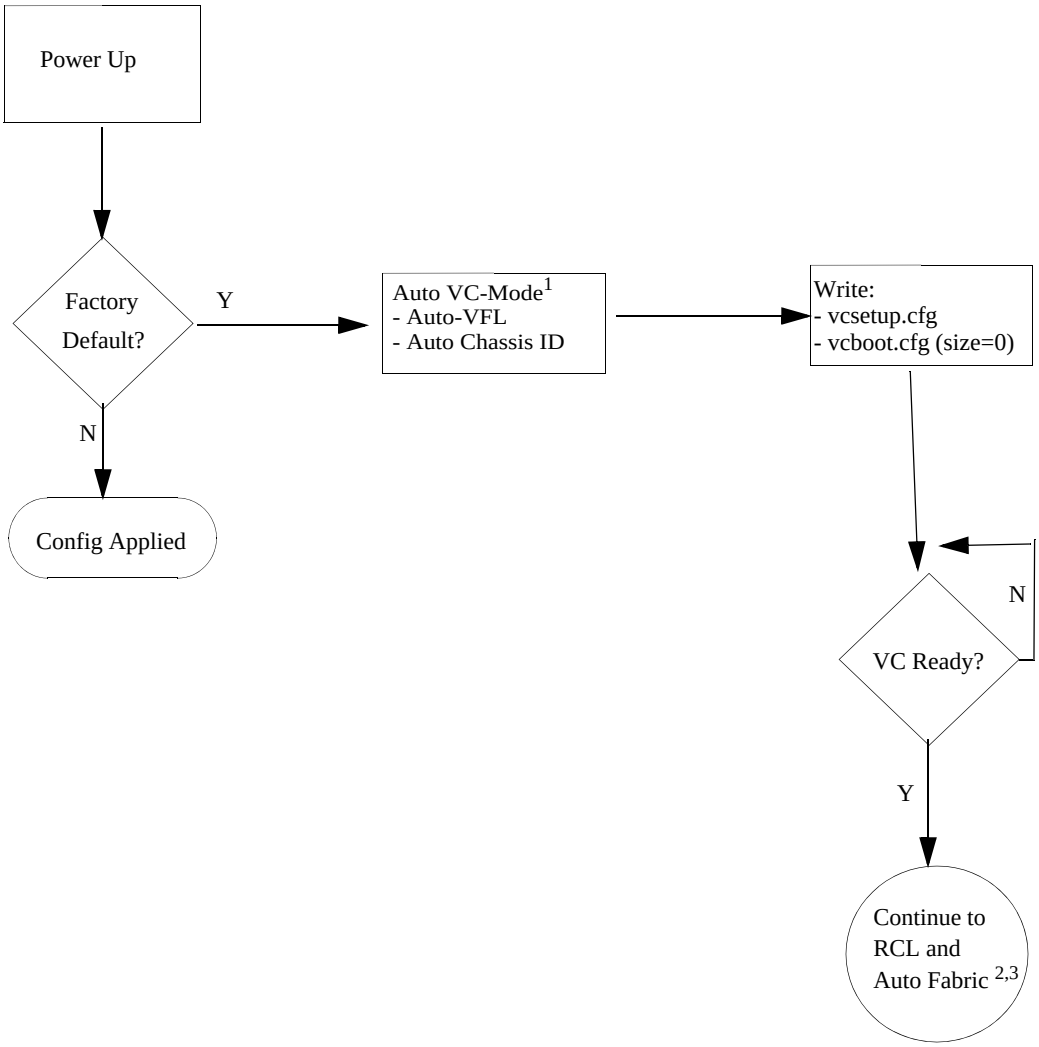
## Runtime Automatic VFL Configuration

**1** The chassis will begin the automatic VFL process on the newly configured automatic VFL ports.

**2** The chassis will communicate with its peers that are also running the VFL protocol to determine which ports will become VFL member ports and what the VFL IDs will be. Multiple ports connected to the same peer chassis will be aggregated and assigned the same VFL ID. Ports connected to different chassis will be assigned different VFL IDs.

# Automatic Virtual Chassis Flow

This following provides a general flow of the Automatic VC setup.




**Automatic VC Flow**

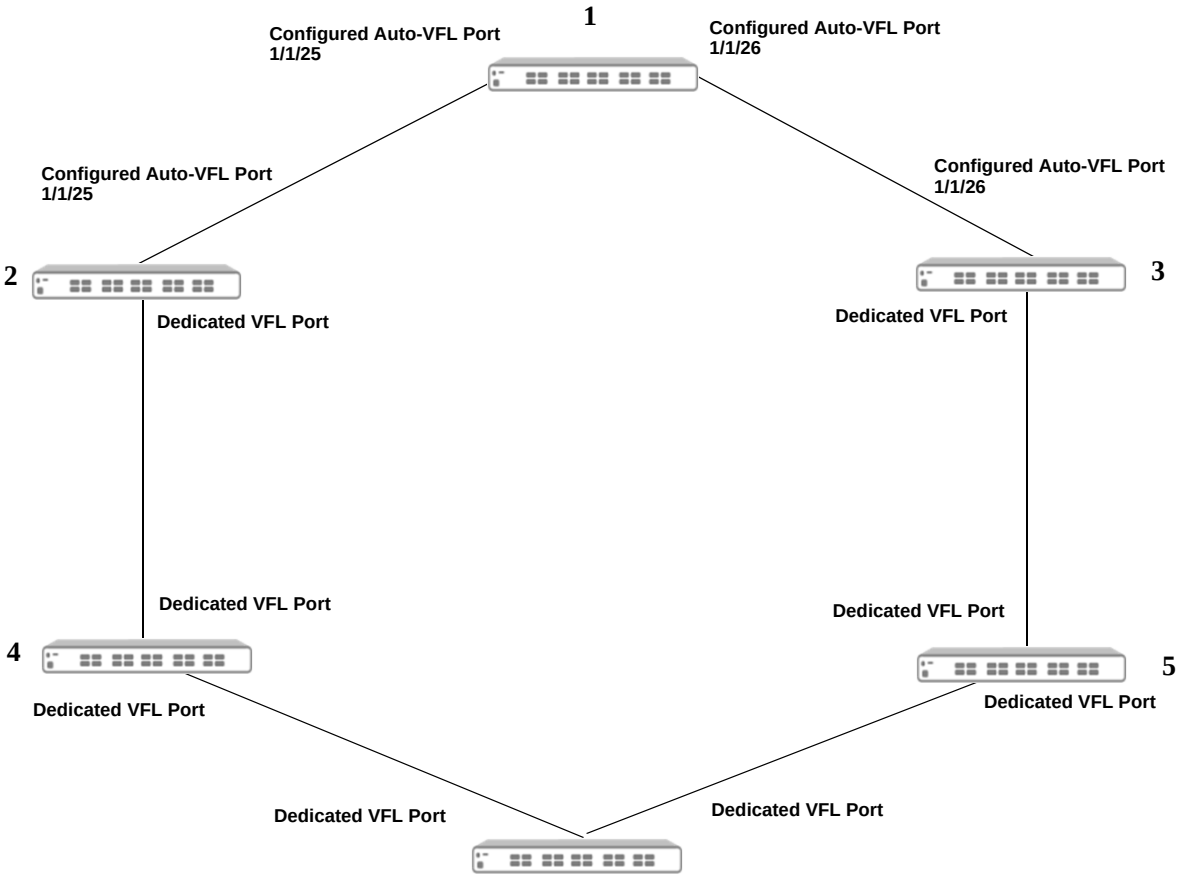
1. See “Automatically Setting up a Virtual Chassis” on page 13-28 for detailed information.

2. See Chapter 14, “Managing Automatic Remote Configuration Download” for additional information on Automatic Remote Configuration Download.

3. See Chapter 15, “Configuring Automatic Fabric” for additional information on Automatic Fabric.

# Virtual Chassis with Auto-VFL and Remote VC

The following example describes how to configure a VC using OS6860s with dedicated VFL ports along with configuring SFP+ ports to be auto-VFL ports. The dedicated VFL ports are always auto-VFL ports, only the SFP+ 10G ports need to be configured as auto-VFL ports. There is no need to configure Chassis-IDs or VFL links, they will be automatically configured.




**Virtual Chassis - Auto-VFL Configuration**

## Virtual Chassis with Dedicated Auto-VFL and Configured Auto-VFL Ports

```
Chassis_1-> virtual-chassis vf-link-mode auto
Chassis_1-> virtual-chassis auto-vf-link-port 1/1/25
Chassis_1-> virtual-chassis auto-vf-link-port 1/1/26

Chassis_2-> virtual-chassis vf-link-mode auto
Chassis_2-> virtual-chassis auto-vf-link-port 1/1/25

Chassis_3-> virtual-chassis vf-link-mode auto
Chassis_3-> virtual-chassis auto-vf-link-port 1/1/26

Chassis_4-> virtual-chassis vf-link-mode auto

Chassis_5-> virtual-chassis vf-link-mode auto

Chassis_6-> virtual-chassis vf-link-mode auto
```

# 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 **virtual-chassis split-protection admin-state** and **virtual-chassis split-protection linkagg** 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

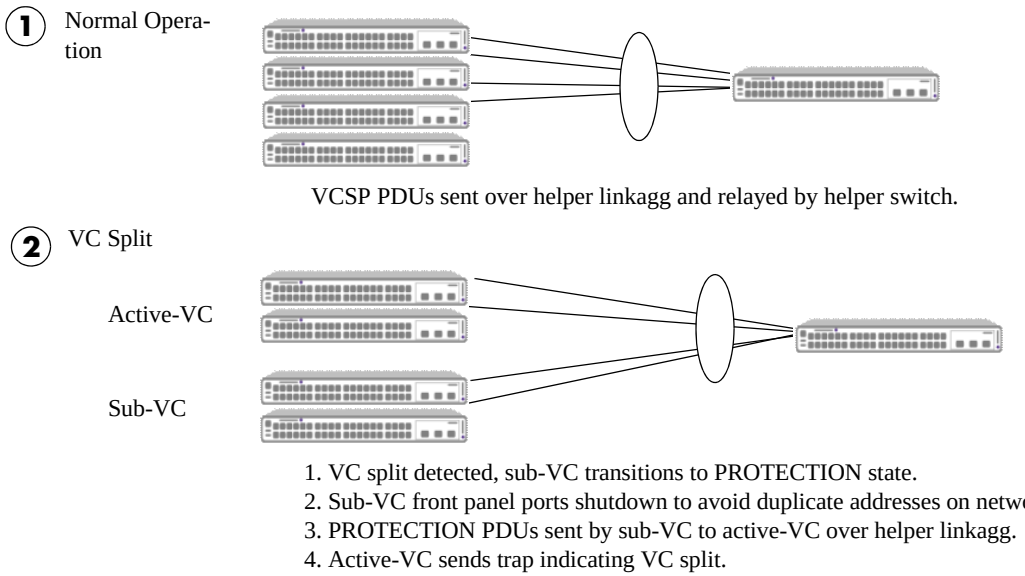
If the guard timer is 0, automatic recovery is disabled and the sub-VC stays in protection mode until the unit is reloaded by the administrator. After the re-boot the administrator has to manually recover the switch by first disabling VCSP and then re-enabling VCSP. This clears the protection state variables

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**

# 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. |
| **show virtual-chassis auto-vf-link-port** | Displays a summary of the auto VFL ports. |

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

# 14 Managing Automatic Remote Configuration Download

The Automatic Remote Configuration capability automates and simplifies the deployment of large network installations eliminating the need for manual configuration of each switch. It also ensures that each switch is compliant with the centrally controlled switch configuration policies and firmware revisions.The Automatic Remote Configuration feature enables:

- the automatic upgrade of firmware and/or configuration of a standalone switch without user intervention.
- the automatic upgrade of firmware and/or configuration of a Virtual Chassis 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 Automatic Remote Configuration on the OmniSwitch. The sections in this chapter are:


- "Automatic Remote Configuration Defaults" on page 14-2
- "Quick Steps for Automatic Remote Configuration" on page 14-4
- "Overview" on page 14-5
- "Interaction With Other Features" on page 14-8
- "Automatic Remote Configuration Download Process" on page 14-9
- "Download Component Files" on page 14-11
- "DHCP Client Auto-Configuration Process" on page 14-15
- "Nearest-Edge Mode Operation" on page 14-17
- "LACP Auto Detection and Automatic Link Aggregate Association" on page 14-19
- "RCL Process Illustration Flow - Chart A" on page 14-23


See Chapter 1, "Getting Started and Upgrading AOS," for licensing information and getting started with this feature.

# Automatic Remote Configuration Defaults

| Description | Default |
|---|---|
| Management VLAN<br>Untagged Management VLAN | VLAN 1 |
| DHCP broadcast VLAN<br>802.1q tagged VLAN | VLAN 127 |
| Default Auto Link Aggregate Creation | VLAN 1 (untagged) and VLAN 127 (tagged) |
| Nearest-edge MAC Address | 01:20: DA: 02:01:73 |
| Instruction file | Location: TFTP Server<br><br>File name: ***.alu**<br>(* represents any instruction filename)<br><br>Download location: **/flash** directory<br>Downloaded as a temporary file. |
| Configuration file | File name: **Any name**<br><br>Location: FTP/SFTP/TFTP Server<br><br>Download location: **/flash/working** directory |
| Debug configuration file | File name: **AlcatelDebug.cfg**<br><br>Location: FTP/SFTP/TFTP Server<br><br>Download location: **/flash/working** directory |
| Script file | File name: **Any name**<br><br>Location: FTP/SFTP/TFTP Server<br><br>Download location: **/flash/working** directory |
| Firmware version | **OS_*_*_R01**<br>(*_* represents version number) |
| Firmware or image files | File name extension: ***.img**<br>(* represents image filename)<br><br>Location: FTP/SFTP/TFTP Server<br><br>Download location: **/flash/working** directory |
| File download server | Primary FTP/SFTP/TFTP Server |
| Backup server for file download | Secondary FTP/SFTP/TFTP Server |
| License file | File name: **swlicense.dat**<br><br>Location: FTP/SFTP/TFTP Server<br><br>Download location: **/flash** directory |

| Description | Default |
|---|---|
| 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 the automatic remote configuration process is initiated.

**5** A DHCP client is automatically configured on the OmniSwitch (see "DHCP Client Auto-Configuration Process" on page 14-15) . 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.

**Notes.**

- 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 on 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:

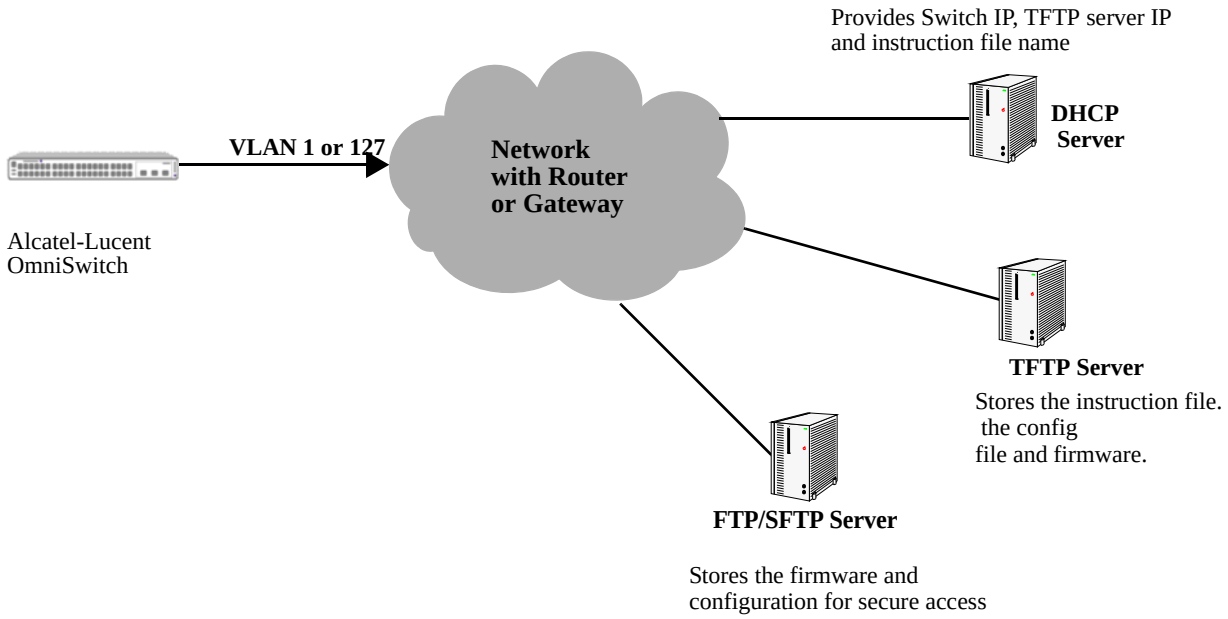
- There is no bootup configuration file (**vcboot.cfg**) on the switch.
- During a takeover or reboot on the new Primary unit or CMM.
- The initialization process of the switch is complete and the network interfaces or ports are ready.
- There is connectivity with a DHCP server through the default VLAN 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 on 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 14-15. 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.
- License 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 14-11.

## 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.

---

**Notes.** 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.

---

**Notes.** 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 14-17.

---

**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 14-9

## 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.
- Automatic Remote Configuration in process: OK1 LED is flashing amber.

# 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 14-15

## 802.1Q

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

## LLDP

In Nearest-Edge mode operation LLDP packets carry and provide the advertised VLAN ID to the 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 Virtual Chassis and Automatic Fabric

- Automatic Remote Configuration will run after a Master is chosen and the VC is established.
- Automatic Remote Configuration will run before any automatic fabric protocols (LACP, SPB, MVRP, Loopback, IP).

# 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 "Troubleshooting" on page 14-20

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 14-19.

**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 14-15

**4** The DHCP client looks for the OV Cirrus DHCP server response to provide preference to the desired OV Cirrus DHCP server. For details, see the following section "DHCP Server Preference" on page 14-16

**5** The DHCP client obtains the switch IP address information from the DHCP server.

**6** The DHCP client obtains the TFTP server IP address from the DHCP server using Option (66).

**7** The DHCP client obtains the instruction file name and location from the DHCP server using Option (67).

**8** SSH access is automatically enabled to allow remote access in case the automatic configuration process fails.

**9** The instruction file with the **.alu** extension is downloaded from the TFTP server to the **/flash/working** directory of the OmniSwitch.

**10** 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.

**11** 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.

**12** 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).

**13** If available, commands in the script file are run and the DHCP client configuration is automatically removed.

## Process Illustration

For a detailed flow chart on the RCL process, see "RCL Process Illustration Flow - Chart A" on page 14-23.

## 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.

---

**Notes.** 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.

**3** The Automatic Remote Configuration process can be stopped using the **auto-config-abort** command.

# 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 14-11
- **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 14-13
- **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 14-13
- **Debug Configuration file**—The debug configuration file stores the default debug configuration information. For further details, see "Debug Configuration File" on page 14-14
- **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 14-14

## 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.

**Notes.**

- 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. |
| License file name and location | The license file containing the licensing information. |
| 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 OS6900 – Instruction file version 1.2.1
! Firmware version
Firmware version:OS_8.3.1_R01
Firmware location:/home/ftpboot/firmware
! Configuration file
Config filename:boot_OS6900.cfg
Config location:/home/ftpboot/config
! Debug file
Debug filename:AlcatelDebug.cfg
Debug location:/home/ftpboot/debug
! Script File
Script filename:OS6900_script.txt
Script location:/home/ftpboot/scripts
! License File
License filename:swlicense.dat
License location:/home/ftpboot/license
! 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 3-1.

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 file (**vcboot.cfg**) is not present during the initial bootup process when a new OmniSwitch 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

- It is recommended to create the script file with a Unix / Linux type text editor. Creating the script file in a Windows environment can result in hidden control characters that may cause issues with script file parsing.
- 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.

# License File

License file (swlicense.dat ) contains the licensing information and will be downloaded as any other file to /**flash** directory of the Master in Virtual Chassis.

# 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:

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

The 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 "Nearest-Edge Mode Operation" on page 14-17 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.

**5** If during this process the switch receives an LACP PDUs it will attempt to automatically create a link aggregate with the peer device. The link aggregate will become part of VLAN 1 (untagged) and VLAN 127 (tagged).

# DHCP Server Preference

When RCL is running and the DHCP client is created, the following steps are followed in order to provide preference to different DHCP servers. When server-preference is enabled, the following precedence order is followed for the VLAN 1 DHCP client.

1.OVCirrus Server:"alenterprise"

2.OVClient Server:"alcatel.nms.ov2500"

3.OXO DHCP Server:"alcatel.a4400.0"

4.Others / Undesired : Identified by absence of VSI string

The following describes the DHCP client preference operation:

**1** If a DHCP response is received on the VLAN 1 DHCP client from a non-preferred DHCP server it will be stored during the 30 second window allowing time for a DHCP response from a higher preference server. Subsequent responses from non-preferred DHCP servers will be dropped.

**2** If a DHCP response is received on the VLAN 1 DHCP client from an OXO DHCP server it will overwrite any non-preferred DHCP response. The response will be stored during the 30 second window allowing time for a DHCP response from an high preference server. Subsequent responses from any OXO DHCP servers or non-preferred DHCP servers will be dropped.

**3** If a DHCP response is received on the VLAN 1 DHCP client from an OmniVista DHCP server it will overwrite any non-preferred DHCP response. The response will be stored during the 30 second window allow ing time for a DHCP response from an OVCloud server. Subsequent responses from any OmniVista /OXO DHCP servers/non-preferred DHCP servers will be dropped.

**4** If a DHCP response is received on the VLAN 1 DHCP client from an OVCloud DHCP server it will overwrite any existing DHCP responses and be applied immediately.

---

**Note**:
- A DHCP server should be configured and have connectivity to the switch during the initial boot-up.

- The RCL process may be delayed while waiting for a preferred server.

---

For more information on configuring DHCP Client and Server preference, See Chapter 22, “Configuring DHCP Relay,” in the *OmniSwitch AOS Release 8 Network Configuration Guide* for more information.

# 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 Automatic Remote Configuration Switch

The Automatic Remote Configuration 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 a 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 switch.

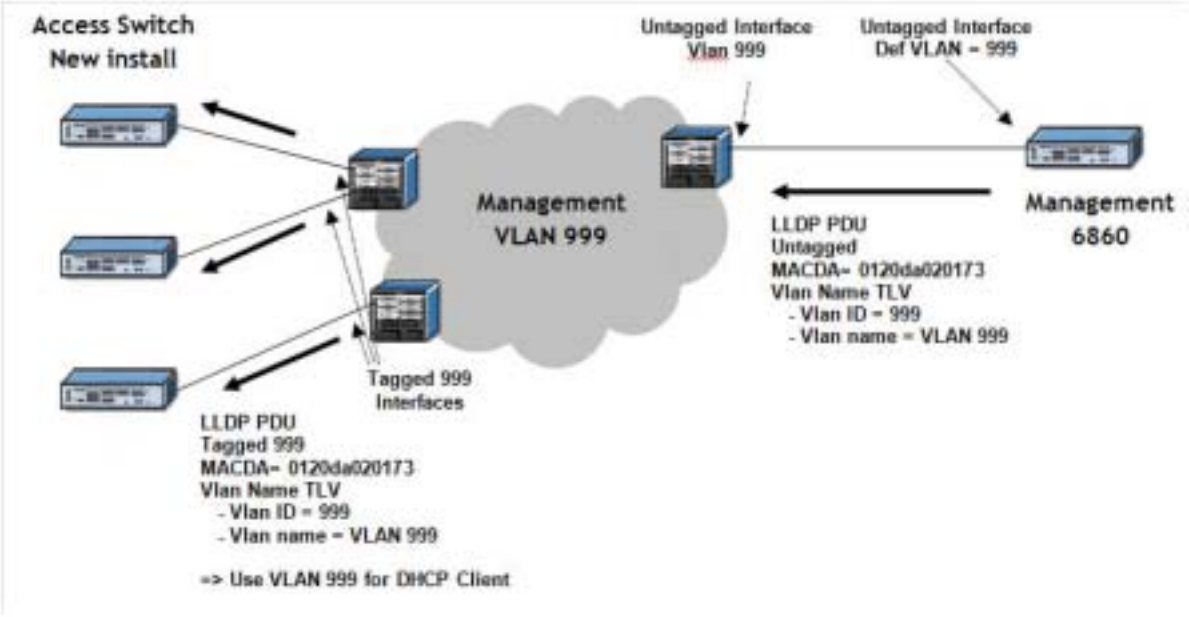
The Management Switch is connected to the network using an untagged interface and is configured to use the Nearest-edge Mode MAC address. LLDP is configured on the untagged port of the Management Switch so that the LLDP PDUs are sent with the Management VLAN information. The LLDP PDUs are sent on the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

For example:

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

## Newly Installed Switch

When used in conjunction with the Automatic Remote Configuraton feature no configuration is necessary on the newly installed switches. 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 Managment VLAN. This auto-configuration allows the DHCP client interface on the OmniSwitch to receive an IP address in the proper IP subnet.

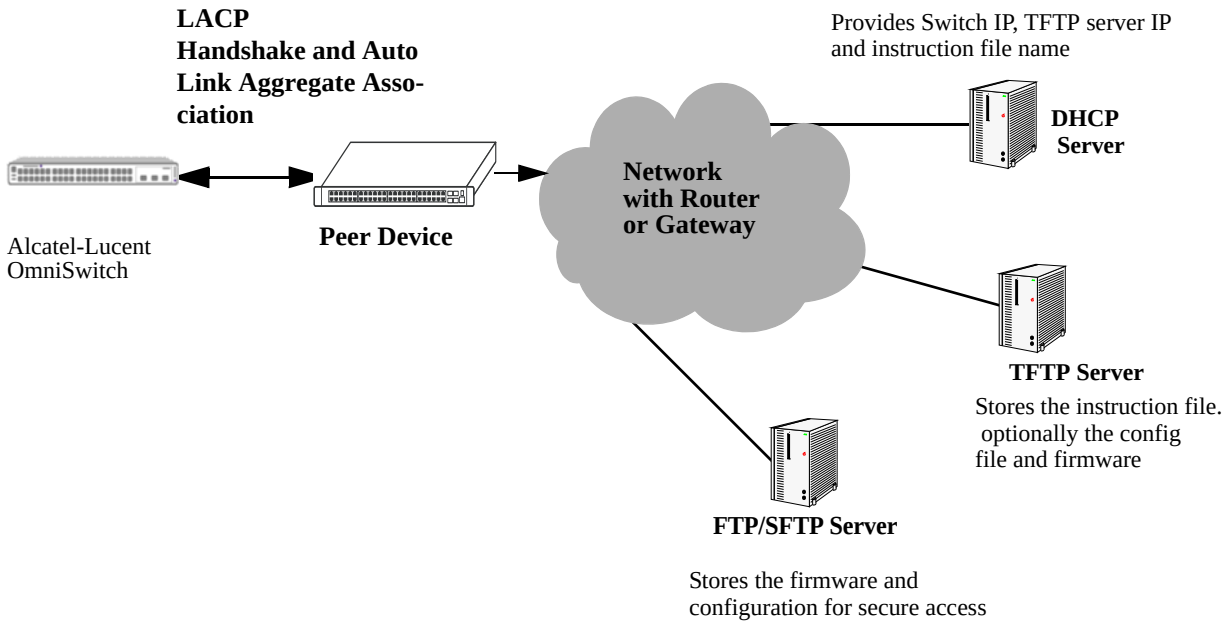



**Example Nearest-Edge Configuration**

# 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 any 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 on all ports on the OmniSwitch during the Automatic Remote Configuration stage.

**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.

---

**Notes.** 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 Auto-Config Abort** | `Automatic Remote Config Abort received.` | User manually aborted the process using the auto-config-abort command |
| **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:

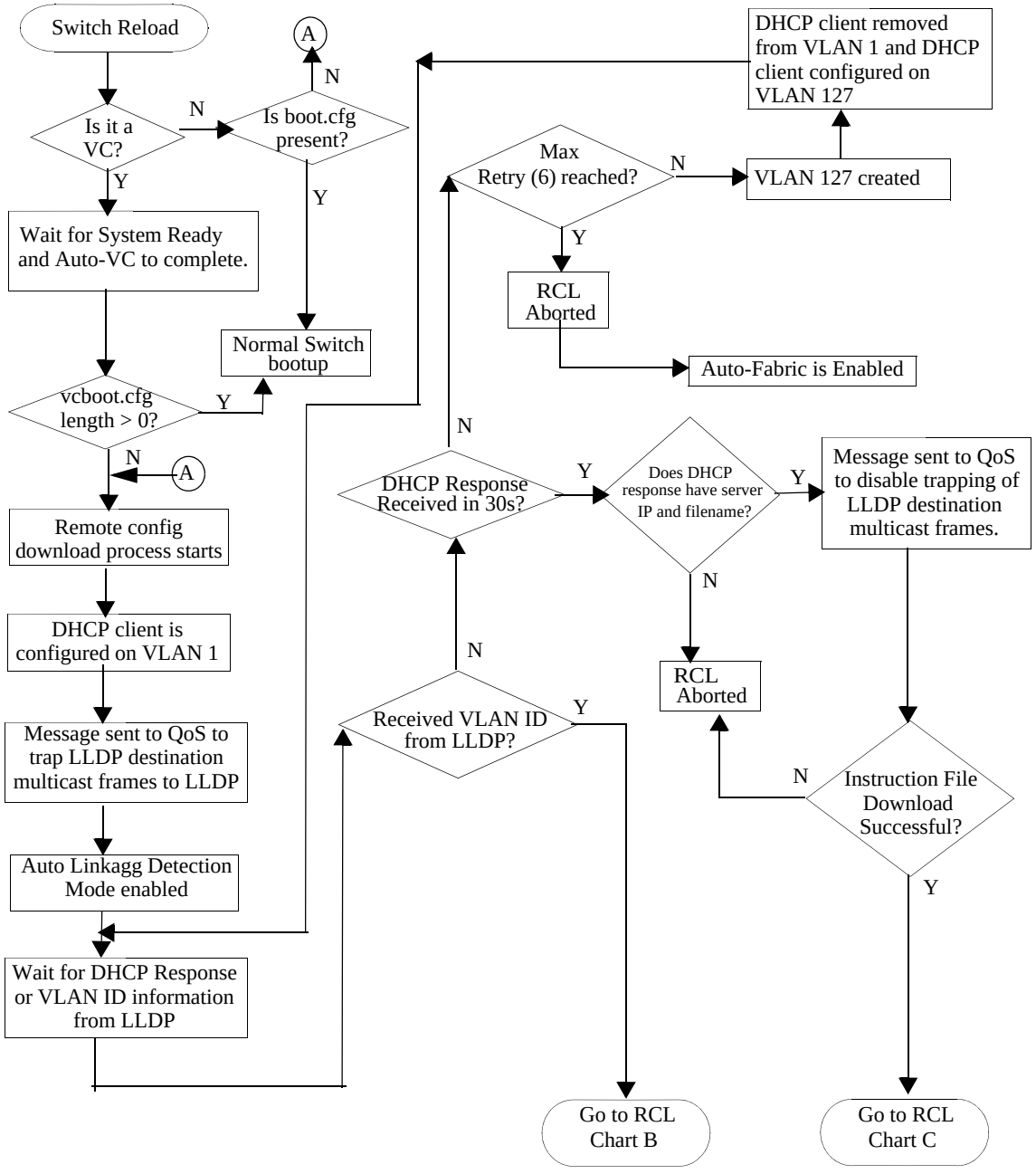
- If any script file command fails, it is logged in to a file ***.err** (* is the script file name) in the **/flash** directory and the remaining commands are implemented. In such an instance, check the ***.err** file. The script file commands can be manually implemented and debugged in the order specified 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:

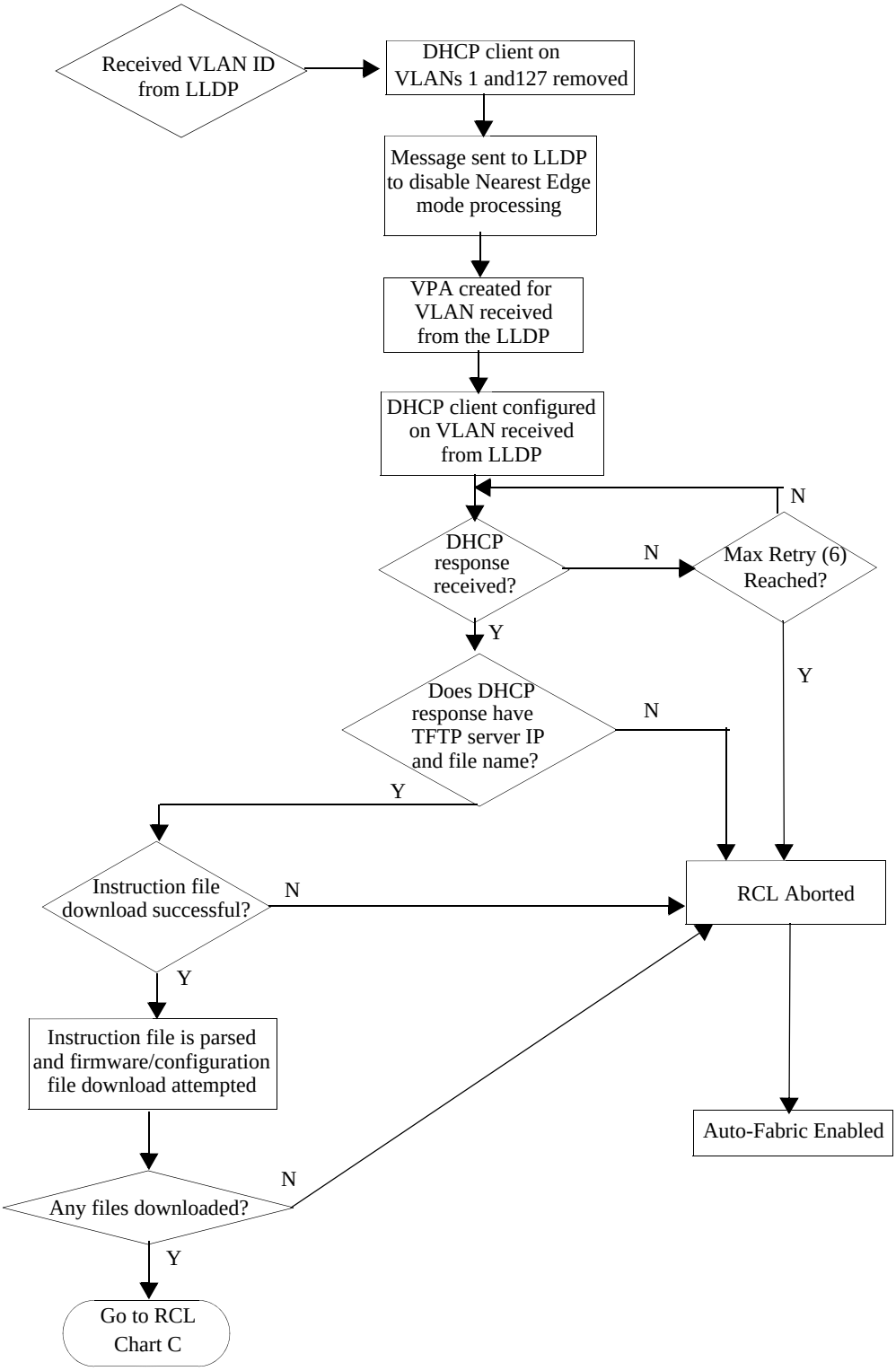
| Error Type | Error | Description |
|---|---|---|
| **Script File Download** | `Download of Script file from Primary Server Failed` | Script file cannot be downloaded from the primary server. |
| | `Starting download of Script file: <File name and pathname> from Secondary Server` | |
| | `Download failed - <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. |

# RCL Process Illustration Flow - Chart A



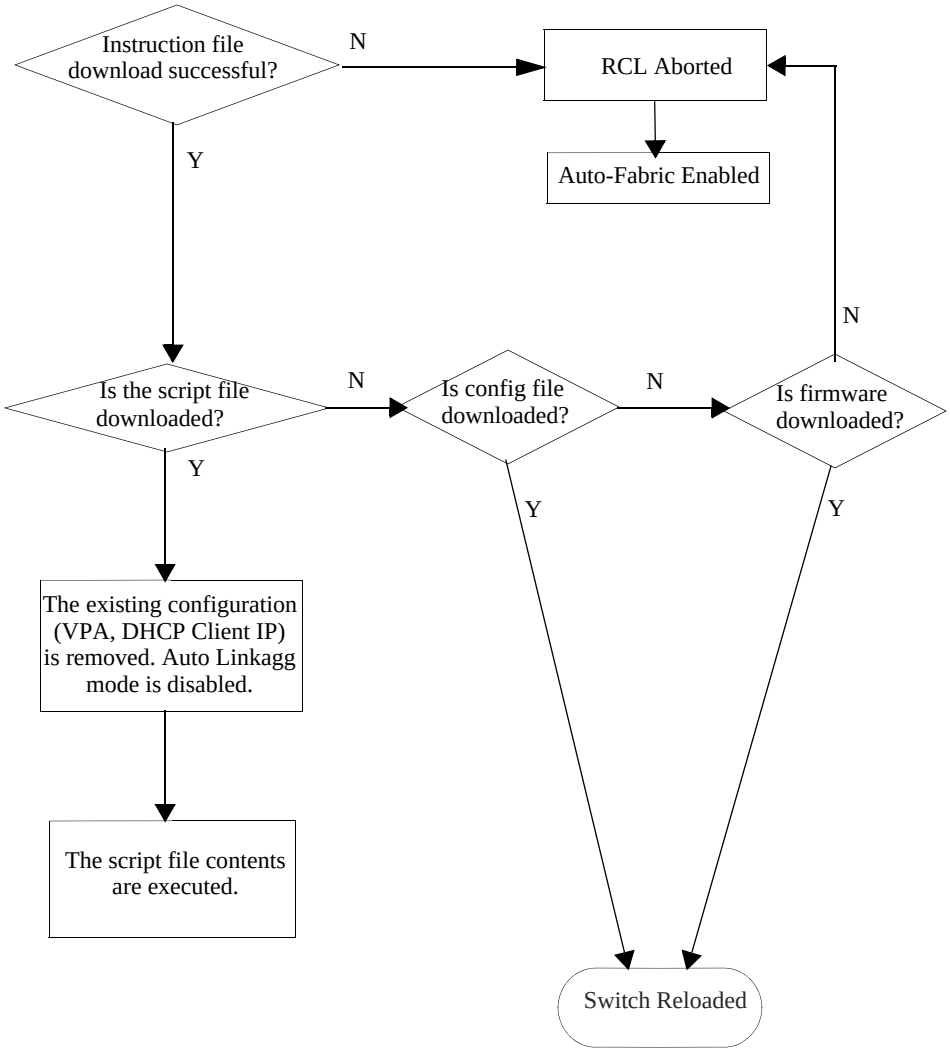

**RCL Flowchart - Graphic A**

# RCL Flow - Chart B




**RCL Flowchart - Graphic B**

# RCL Flow - Chart C




**RCL Flowchart - Graphic C**

# 15 Configuring Automatic Fabric

The Automatic Fabric feature can be used to bring up an OmniSwitch by automating some of the tedious and error prone steps, such as link aggregate formation and Shortest Path Bridging (SPB) neighbor adjacency formation. Dynamic recognition of the neighboring elements allows for a quick, out-of-the-box configuration of the switch. The focus area for this feature is in the data center, but Automatic Fabric is also applicable in a campus LAN environment to help reduce administrative overhead.

This feature is supported in both standalone or virtual chassis mode. Automatic Fabric discovery will not operate until after the Virtual Chassis (VC) setup is completed and normal configuration commands are applied from the configuration file, if present. If enabled, the switch will then attempt automatic discovery and configuration for LACP, SPB, and MVRP. In addition, automatic discovery and configuration for IP protocols is performed in parallel with the LACP, SPB, and MVRP discovery phases.

The Automatic Fabric feature allows a true fabric to be built when a device is plugged into the network and automates the edge port configuration with profiles.

For more information about Automatic Fabric, see “Automatic Fabric Overview” on page 15-7.

# In This Chapter

This chapter describes the basic components of Automatic Fabric and its operation and configuration 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:


- "Automatic Fabric Default Values" on page 15-3
- "Quick Steps for Configuring Automatic Fabric" on page 15-4
- "Automatic Fabric Overview" on page 15-7
- "Automatic Fabric Discovery Examples" on page 15-17
- "Interaction with Other Features" on page 15-21
- "Configuring Automatic Fabric" on page 15-25.
- "Displaying the Automatic Fabric Configuration" on page 15-29


See Chapter 1, "Getting Started and Upgrading AOS," for licensing information and getting started with this feature.

# Automatic Fabric Default Values

The following default settings are applied for the Automatic Fabric feature:

| Parameter Description | Command | Default Value/Comments |
|---|---|---|
| Automatic Fabric administrative state | **auto-fabric admin-state** | enabled (if no configuration file exists)<br>disabled (OS9900) |
| Automatic Fabric protocols state | **auto-fabric protocols** | enabled<br>disabled (OS9900) |
| Automatic Fabric configuration save administrative state | **auto-fabric config-save admin-state** | disabled |
| Automatic Fabric configuration save interval | **auto-fabric config-save interval** | 300 seconds (if automatic configuration save is enabled). |
| Automatic Fabric discovery interval | **auto-fabric discovery-interval** | 0 (discovery window timer is disabled) |
| Automatic Fabric SPB default SAP profile | **auto-fabric protocols spb default-profile** | auto-vlan (SAPs are created based on VLAN tag) |

# Quick Steps for Configuring Automatic Fabric

The following steps provide a quick tutorial for setting up a basic Automatic Fabric configuration. This scenario applies to the default operation of a switch without a configuration file, as well as configuring a switch with an existing configuration file. Additional information about how to configure Automatic Fabric is provided in the section "Configuring Automatic Fabric" on page 15-25.

## Automatic Fabric Operation with No Configuration File

When the switch boots up and there is no configuration file, the Automatic Fabric operation is automatically enabled and triggers the following discovery process:

**1** The switch will attempt to discover and automatically set up an LACP configuration.

**2** After the LACP discovery process completes, the switch will attempt to discover and automatically set up a Shortest Path Bridging (SPB) configuration. This includes discovering and configuring SPB adjacencies, UNP SPB access ports, and UNP SPB Service Access Points (SAPs).

**3** After the SPB discovery process completes and if MVRP is enabled, the switch will attempt to discover and automatically set up an MVRP configuration. As part of the MVRP discovery process, the Spanning Tree mode for the switch is changed to the flat Spanning Tree mode.

**4** The automatic discovery process for IP protocols takes place in parallel with the discovery process for the other supported Automatic Fabric protocols (as described in Steps 1, 2, and 3).

Once the switch boots up and the Automatic Fabric process has completed, the default settings for the Automatic Fabric parameters can be configured.

## Configuring Automatic Fabric Parameters

When a switch is already up and running with an existing configuration file, it is possible to change default parameter settings (see "Automatic Fabric Default Values" on page 15-3) to fine tune the Automatic Fabric operation going forward.

**1** To change the global administrative status of Automatic Fabric for the switch, use the **auto-fabric admin-state** command. For example:

```
-> auto-fabric admin-state disable
```

**2** To change the status of Automatic Fabric on specific ports, use the **auto-fabric admin-state** command with the **interface** parameter. For example:

```
-> auto-fabric interface 1/1/1-4 admin-state disable
```

The Automatic Fabric status configured for a port takes precedence when the global status is enabled for the switch. For example, if Automatic Fabric is disabled on a port but globally enabled for the switch, Automatic Fabric will not run the discovery process on that port.

**3** To change the status of Automatic Fabric discovery for specific protocols, use the **auto-fabric protocols** command. For example:

```
-> auto-fabric protocols lacp admin-state enable
-> auto-fabric protocols mvrp admin-state enable
-> auto-fabric protocols spb interface 1/1/3 admin-state disable
-> auto-fabric protocols ip ospfv2 admin-state enable
-> auto-fabric protocols ip isis admin-state disable
```

**4** To change the status of Loopback Detection on UNP SPB access ports, use the **auto-fabric protocols** command with the **loopback-detection** parameter. For example:

```
-> auto-fabric protocols loopback-detection admin-state disable
```

**5** To change the Automatic Fabric discovery window time interval, use the **auto-fabric discovery-interval** command. For example:

```
-> auto-fabric discovery-interval 30
```

This value specifies the number of minutes the switch will wait between each attempt to discover a configuration for the switch. When the discovery window time interval is set to zero (the default), the discovery interval is disabled.

**6** By default, the Automatic Fabric configuration save operation is disabled for the switch. To enable this function, use the **auto-fabric config-save admin-state** command. For example:

```
-> auto-fabric config-save admin-state enable
```

When this function is enabled, the configuration discovered through the Automatic Fabric process is automatically saved to the switch configuration file at a specified time interval.

**7** When the Automatic Fabric configuration save operation is enabled, the switch will save the discovered configuration to the switch configuration file every 300 seconds (5 minutes) by default. To change this time interval, use the **auto-fabric config-save interval** command. For example:

```
-> auto-fabric config-save interval 600
```

**8** To change the default profile used to dynamically create a Service Access Point (SAP) on UNP SPB access ports, use the **auto-fabric protocols spb default-profile** command. For example:

```
-> auto-fabric protocols spb default-profile single-service
```

A single service profile specifies attributes for untagged traffic; an auto-VLAN profile (used by default) specifies attributes for tagged traffic.

**9** To change the default SAP profile applied to a specific port, use the **auto-fabric protocols spb set-profile** command. For example:

```
-> auto-fabric protocols spb set-profile single-service interface 1/1/1
```

The default SAP profile configured for a port takes precedence over the default SAP profile configured globally for all UNP SPB access ports. For example, if a single service profile is specified for a port but the default SAP profile for the switch is auto VLAN, the single service profile is used to create the SAP on that port.

---

**Note.** To enable Automatic Fabric after the switch has booted, the protocol must also be enabled.
**(**i.e. **ip load ospf** and **ip ospf admin-state enable**)

---

## Verifying the Automatic Fabric Configuration

Use the **show auto-fabric config** command to check the global configuration for the Automatic Fabric feature. For example:

```
-> show auto-fabric config
Auto-fabric Status          : Disabled,
Config Save Timer Status    : Enabled,
Config Save Timer Interval  : 600 seconds,
Default UNP SAP Profile     : Auto-vlan,
Discovery Interval          : 30 minute(s),
Discovery Status            : Idle,
LACP Discovery Status       : Enabled,
LBD Discovery Status        : Disabled,
MVRP Discovery Status       : Enabled,
OSPFv2 Discovery Status     : Enabled,
OSPFv3 Discovery Status     : Disabled,
ISIS Discovery Status       : Disabled,
SPB Discovery Status        : Enabled
```

Use the **show auto-fabric config interface** command to check the Automatic Fabric configuration for a specific interface. For example:

```
-> show auto-fabric config interface 1/1/1
Auto-Fabric Interface Config:
    Port 1/1/1 :
      Operational Status : Disabled
        Admin-Status
          Global : Disabled,   Port : Disabled
        LACP
          Global : Enabled,    Port : Enabled
        SPB-M
          Global : Disabled,   Port : Disabled
        MVRP
          Global : Enabled,    Port : Enabled
        SAP Profile
          Global : Auto-vlan   Port : Single-service
```

# Automatic Fabric Overview

The Automatic Fabric feature reduces the burden of configuration on the administrator. Dynamic recognition of the neighboring elements will allow for quick, out-of-the-box configuration and reduced administrative overhead. Automatic Fabric is used to dynamically discover and configure a switch for the LACP, SPB, MVRP, and IP protocols and is supported when the switch is operating in standalone or Virtual Chassis (VC) mode.

Some of the key benefits provided by Automatic Fabric include the following:

- Automatic discovery reduces administrative overhead.
- Automatic discovery supports the discovery of the LACP, SPB, MVRP, and IP protocols.
- The automatically discovered configuration for LACP and SPB (not MVRP) can be permanently saved to the switch configuration file so that the configuration is not lost on the next switch reboot.

All switches that ship from the factory default to running in the VC mode and attempt to run the automatic VC protocol, Automatic Remote Configuration, and then Automatic Fabric. Some of these automatic features can be disabled during the switch reboot or after the switch has finished booting if desired.

When a switch boots with no configuration file or with a configuration file with a size of 0 bytes, the following boot processes occur:

**1** The switch will run the automatic VC protocol and try to automatically configure the Virtual Fabric Links (VFLs) and setup a VC.

**2** Once the automatic VC process completes, the automatic remote configuration download process starts.

**3** Once the automatic remote configuration download process completes, the Automatic Fabric discovery process starts.

For more information about the boot sequence of these automatic management features, see Chapter 1, “Getting Started and Upgrading AOS.”

## Automatic Fabric Discovery Process

The Automatic Fabric discovery process starts when one of the following occurs:

- The switch boots up without a configuration file and the automatic VC and Automatic Remote Configuration processes have completed.
- The switch boots up with an existing configuration file that enables Automatic Fabric for the switch.
- The Automatic Fabric discovery time interval expires. For example, if the time interval is set for 30 minutes, every 30 minutes the discovery process will start again.
- The administrator manually starts the discovery process on the switch.

Once the Automatic Fabric discovery process starts, the following events are triggered for ports on which the Automatic Fabric feature is enabled:

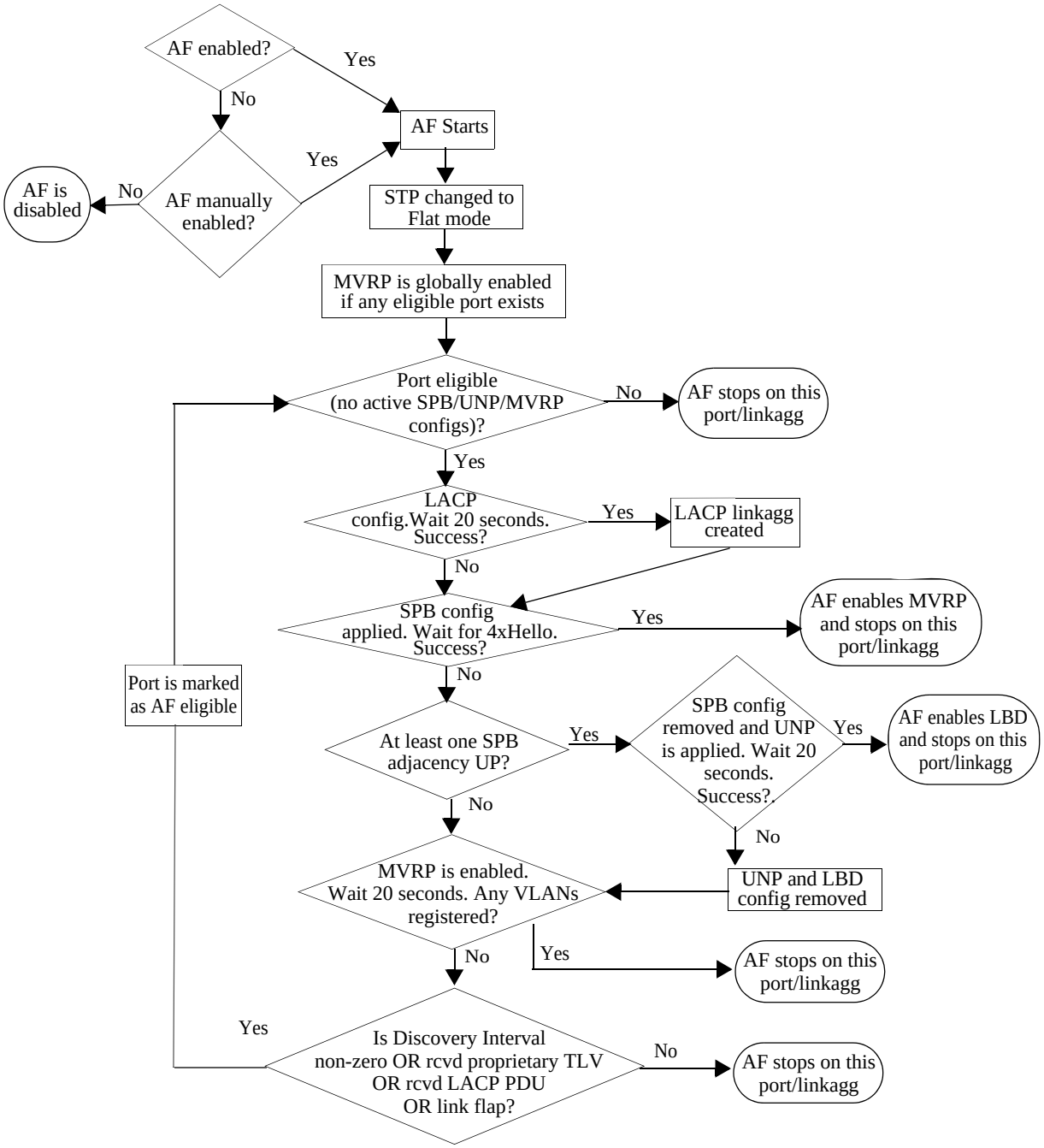
**1** The switch will start the LACP discovery process.

**2** After the LACP discovery process is complete, the SPB automatic discovery process will start.

**3** After the SPB and UNP SPB SAP discovery process is complete, the MVRP automatic discovery process will start.

The automatic IP protocols discovery process runs at the same time as the discovery processes for LACP, SPB, and MVRP. See "IP Protocol Discovery" on page 15-13 for more information.

The following diagram illustrates the Automatic Fabric (AF) discovery and configuration process:



## Automatic Fabric Port Eligibility

The following conditions determine whether a switch port is eligible to participate in the Automatic Fabric discovery process:

- The port has no previous configuration that would prevent the port from joining a link aggregate, forming an SPB adjacency, serving as a UNP SPB access port, and enabling MVRP to run on the port. In other words, there is no switch configuration that is applied to the port that would prevent the port from participating in any of the protocols managed by the Automatic Fabric feature. If a port meets these requirements, the port is considered to be in a default port state.
- If MVRP discovery is enabled and the Spanning Tree mode is set to per-VLAN (1x1), Automatic Fabric will automatically change the Spanning Tree mode to flat. This will allow switch ports to participate in the MVRP discovery process.
- For a port that has MVRP enabled through Automatic Fabric but has no VLAN registrations, if removing MVRP would result in the port returning to its default state, then Automatic Fabric will be enabled on that port.

Note that Automatic Fabric discovery and configuration of IP protocols is only performed on existing IP interfaces. As a result, port eligibility is not considered in this case. See “IP Protocol Discovery” on page 15-13 for more information.

## Automatic Fabric Discovery Window

The Automatic Fabric discovery process initiates a time period during which Automatic Fabric ports are examined to detect any configuration for LACP, SPB, and MVRP. This time period is referred to as the Automatic Fabric discovery window.

- Once a configuration is detected and written to the switch configuration file, the port state for Automatic Fabric is disabled so that the port will not participate in the next discovery window. This does not globally disable Automatic Fabric for the switch.
- Whenever a port is brought up, the discovery window will be started provided no LACP is discovered, no SPB adjacencies are formed, and, if MVRP is enabled, there are no VLAN registrations.
- If an LACP frame is received on a non-aggregate port with Automatic Fabric enabled, the Automatic Fabric discovery window is started, provided no SPB adjacencies are formed and there are no MVRP registrations on that same port.
- When the discovery cycle ends, the following occurs:
  - The MVRP configuration for any port or link aggregate that does not have any VLAN registrations is removed.
  - The configuration for UNP SAPs dynamically created on UNP access ports is removed only if there is no traffic active on the access ports.

The discovery and configuration process for IP protocols is done in parallel, on a per-IP interface basis, and is not tied to a discovery window time period. For more information, see “IP Protocol Discovery” on page 15-13.

## LACP Discovery

The LACP discovery phase analyzes any LACP PDUs or automatic discovery LLDP PDUs received on an Automatic Fabric port. This is done to determine if there is an existing link aggregate the port should join or if creating a new link aggregate with a peer is necessary.

- LACP automatic discovery will work between a configured switch and an automatic discovery enabled switch. The automatic discovery switch analyzes the LACP PDUs received from the configured switch. In this scenario, an automatic discovery switch will place all of the ports from the same switch with the same remote admin key into the same link aggregate.
- LACP link aggregates are configurable between any two automatic discovery switches by exchanging custom LLDP PDUs with TLVs specific to the OmniSwitch. This exchange is necessary to determine an admin key that both devices will use later for actual LACP communication. This exchange will also determine the possible ports that can be part of a link aggregate. This is used only when LACP discovery fails on the port.
- By default, LACP link aggregates created as a result of the LACP discovery process are automatically configured to use the tunnel protocol hashing algorithm.

See the “LLDP” and “LACP” sections of “Interaction with Other Features” on page 15-21 for more information.

## SPB Discovery

After the LACP discovery phase has completed, the SPB discovery phase starts on the Automatic Fabric ports. The main purpose of the SPB discovery phase is to configure the switch with the ability to participate in an SPB backbone configuration. In addition to discovering SPB adjacencies and configuring UNP access ports, the following SPB elements are configured on the switch:

- BVLANs 4000-4015 are created and mapped to Equal Cost Tree (ECT) IDs 1-16, respectively.
- BVLAN 4000 will serve as the control BVLAN on which ISIS-SPB Hello packets are sent.
- Bridge priority is set to 0x8000.

During this phase, all the Automatic Fabric ports are treated as network ports (SPB interfaces) on which the discovery of SPB adjacencies is attempted. If at least one SPB adjacency is established on the switch, UNP access port configuration is attempted on ports or link aggregates that were not used to form SPB adjacencies. Once configuration is finalized and traffic is received on the UNP access ports, the access port configuration is retained even if an adjacency goes down.

In addition to the following subsections, see the “Shortest Path Bridging” section of “Interaction with Other Features” on page 15-21 for more information.

### Dynamic Service Access Points (SAPs)

A SAP is a logical service entity that is configured on a switch to bind a service access port and traffic received on that port to an SPB service ID. During Automatic Fabric discovery of SPB, ports may get converted to UNP access ports. This is done because UNP supports dynamically creating SPB service profiles and corresponding SAPs for traffic received on UNP access ports.

The UNP feature supports SPB service profiles. This type of profile triggers the dynamic creation of a SAP when traffic received on a UNP access port is classified and assigned to that profile. A user-defined SPB service profile specifies the following attributes that are used to dynamically create the SAP:

- The VLAN tag combined with the local UNP access port specifies the encapsulation value for the SAP. For example, “1/1/2:50” specifies that traffic received on access port 1/1/2 tagged with VLAN 50 is mapped to the SAP for encapsulation and tunneling through the SPB backbone.
- The SPB service instance identifier (I-SID) and BVLAN ID specify the SPB service for the SAP that will forward the encapsulated traffic through the SPB backbone.

See the “UNP Dynamic SAPs” section of “Interaction with Other Features” on page 15-21 for more information.

## System Default Profile

To further automate this process, UNP also supports dynamically creating a “System Default” SPB service profile for traffic received on UNP access ports that is *not* classified into a user-defined UNP service profile. This is the case with traffic received on ports converted to UNP access ports during the SPB discovery process. The attribute values that a “System Default” service profile uses to dynamically create a SAP are derived as follows:

- The VLAN tag value is based on the Automatic Fabric setting for an SPB SAP profile. There are two types of SPB SAP profiles available: single service and auto-VLAN.
  - The single service profile is used to create a SAP for untagged traffic received on a UNP access port.
  - The auto-VLAN profile is used to create a SAP for each VLAN ID tag received on the UNP access port.
- The SPB I-SID and BVLAN ID value for the SAP is based on an internal calculation performed by the switch.

In this scenario, traffic arrives on the UNP access port and triggers the switch to dynamically create a “System Default” service profile. Then, based on the Automatic Fabric default SPB SAP profile setting (single service or auto-VLAN), the traffic received is examined to define the SAP that is dynamically created to bind the traffic to an SPB service. The SPB service associated with the dynamic SAP is identified through the I-SID and BVLAN values derived.

## Loopback Detection

A provider network with a set of multiple switches interconnected together can be logically viewed as a large single switch. The large single switch provides service access points to customer networks. Configuration faults in customer networks can result in loops spanning both provider and customer networks. This can result in broadcast storms. In order to protect a provider network from broadcast storms, loops that involve SAP ports need to be detected and broken.

Loopback Detection (LBD) can detect and break loops created on SAP interfaces. For a SAP, the LBD can be enabled for a specific port or link aggregate that is assigned to the SAP. LBD for SAPs allows shutting down only the specific interface (port or link aggregate) of the link involved in the loop.

Automatic Fabric supports LBD on SAP interfaces. Dual-homed connections can be done through link aggregate connections to two or more devices that are part of the same VC. If a switch is connected to multiple devices that are not part of the same VC, the port is converted to an access port and an LBD protocol will be run on these ports.

The status of LBD is configurable through Automatic Fabric commands and applied to the dynamically created SAPs resulting from SPB discovery and configuration.

## MVRP Discovery

MVRP is enabled globally after link aggregates are formed and SPB configuration exchange is completed between peer devices.

**Note.** MVRP is supported only when the switch is operating in the flat Spanning Tree mode. If the switch is running in the per-VLAN (1x1) mode when Automatic Fabric discovery is started for MVRP, the Spanning Tree mode is automatically changed to the flat mode.

- MVRP will operate and accept VLAN registrations on all Automatic Fabric ports and link aggregates that are up. However, if LACP is stopped on a port for any reason, MVRP will not operate on that port.
- If a port or link aggregate goes down, any MVRP configuration is removed from the port or link aggregate.
- MVRP is not enabled on ports that were configured as UNP access ports during the SPB discovery process.
- MVRP configuration learned through the Automatic Fabric process is not written to the switch configuration file. This means that dynamically learned MVRP VLANs are not saved to the switch configuration file. To retain these VLANs so that they are not lost when the switch reboots, manually convert them to static VLANs.
- All VLANs are eligible for MVRP registration, except for SPB BVLANs. There is no reason to share BVLANs through MVRP as the BVLAN topology is already created through Automatic Fabric discovery or manual configuration.
- There are no default MVRP VLANs.
- If no VLAN registrations are found when MVRP is enabled, then the port property is removed and set to its default state.
- MVRP is not tied to a discovery window time period. The MVRP operation is continuous until the administrator makes changes.

It is important to note that the global status of Automatic Fabric discovery for the MVRP protocol is automatically changed when the following conditions occur:

- When the switch boots up without a configuration file, Automatic Fabric enables MVRP discovery and changes the Spanning Tree mode to the flat mode for the switch. The global setting for MVRP on the switch is also set to enabled.
- When the switch boots up with an existing configuration file, Automatic Fabric globally disables MVRP discovery by default. However, if the MVRP discovery setting was user-configured, that value is retained and not automatically changed when the switch boots up. The Spanning Tree and MVRP status for the switch is not changed.

See the “MVRP and Spanning Tree” section of “Interaction with Other Features” on page 15-21 for more information.

## IP Protocol Discovery

The Automatic Fabric discovery and configuration functionality is also extended to IP protocols. However, the discovery and configuration process is not based on physical switch ports. Instead, existing IP interfaces listen for protocol messages to discover if any neighbors are running OSPFv2, OSPFv3, IS-IS IPv4, or IS-IS IPv6. When a response from a neighboring switch is received for any of these protocols, the automatic configuration of the protocol is triggered on the local switch.

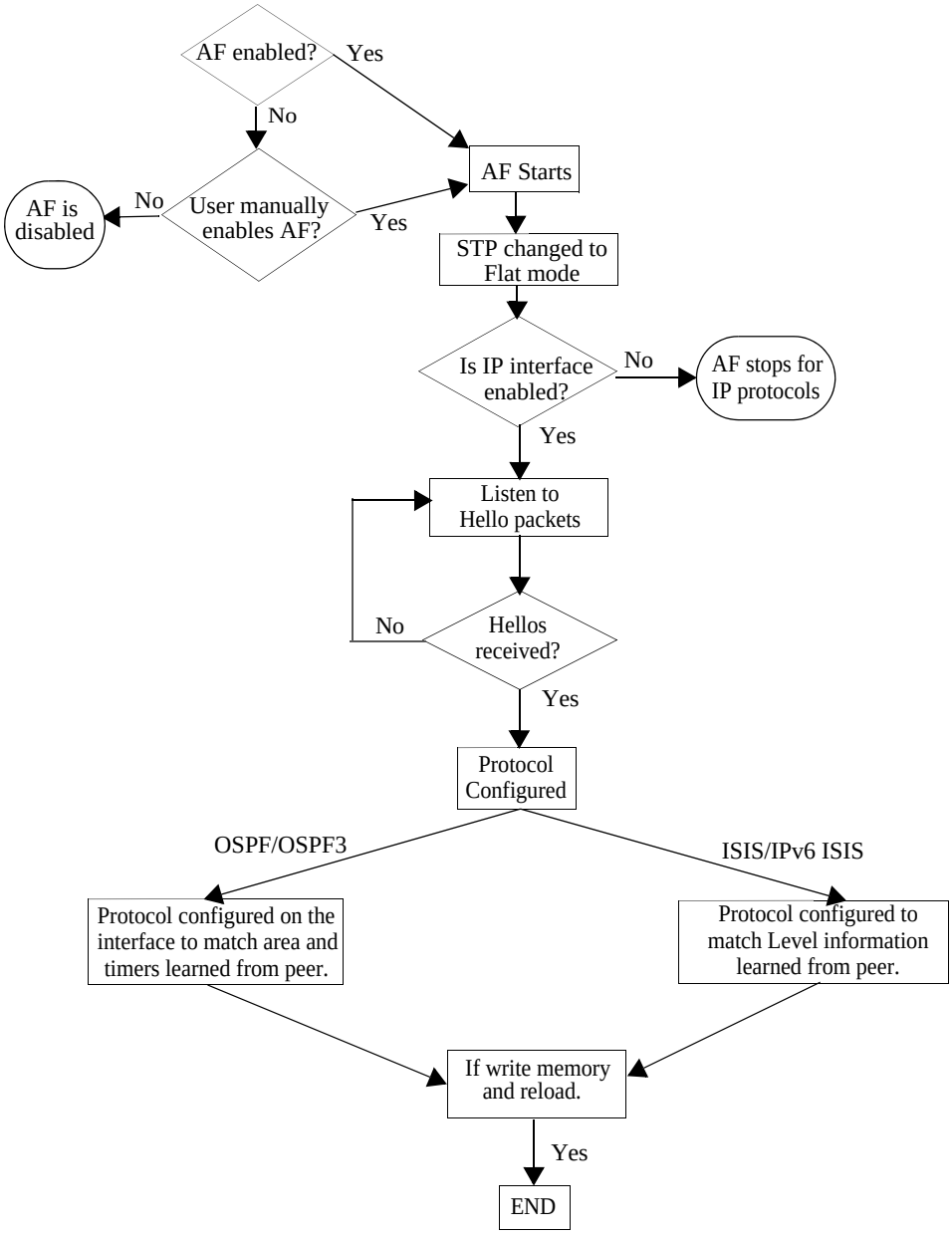
Although automatic IP configuration is triggered when Automatic Fabric starts, the IP discovery and configuration process runs in parallel with the LACP, SPB, and MVRP discovery processes. However, if an IP interface comes up as a result of one of these other discovery processes, automatic IP configuration is triggered on that interface.

Automatic IP runs only when an active IP interface exists on the switch, the interface is not already configured for the routing protocol, and discovery for Automatic Fabric IP protocols is enabled. Once an IP interface is created, the interface will listen for hello packets from the neighboring devices and automatically configure the basic routing parameters based on the information received in the hello packets.

The IP protocol configuration discovered and configured through this process is saved as part of the Automatic Fabric configuration. For more information, see “Saving the Configuration Discovered by Automatic Fabric” on page 15-15.

**Note**: Automatic IP discovery is designed for use in more simplistic networks. It is not recommended to be used for complex networks such as those with multiple OSPF areas.

The following diagram illustrates the Automatic Fabric (AF) discovery and configuration process for the OSPF and IS-IS routing protocols:




## Automatic IP Protocol Configuration

The following switch configuration requirements support the automatic IP protocol configuration process:

- At least one existing IP interface that does not already have a protocol configuration.
- Automatic configuration for the IP protocols (OSPFv2, OSPFv3, IS-IS IPv4, IS-IS IPv6) is enabled (the default). The automatic configuration status for these protocols is configured through an Automatic Fabric CLI command (see “Configuring the Discovery Status for Specific Protocols” on page 15-26).
- The Automatic Fabric feature is enabled for the switch (see “Enabling or Disabling Automatic Fabric” on page 15-25).

- A neighbor is detected on at least one IP interface within a VRF instance.

The following events will trigger the automatic IP protocol configuration process on an IP interface:

- When an IP interface comes up and Automatic Fabric is enabled for protocol PDUs received on the interface.
- If an IP interface is already up and Automatic Fabric is enabled for protocol PDUs received on the interface.

The automatic IP protocol configuration process listens on active IP interfaces for protocol Hello packets received from neighboring switches. This is done to detect and learn the network protocol configuration.

The following events will disable the automatic IP protocol configuration process:

- Automatic Fabric is globally disabled for the switch.
- Automatic configuration for a specific IP protocol is disabled.
- A switch reboots with no Automatic Fabric commands in the configuration file.
- A Hello packet is received on the IP interface.

If an IP interface is toggled or a routing protocol is disabled and re-enabled, automatic configuration resumes listening for Hello packets.

See “Automatic Fabric Process for Automatic IP Configuration” on page 15-19 for general examples of automatic IP protocol configuration.

## Saving the Configuration Discovered by Automatic Fabric

The discovered configuration remains in switch memory until one of the following occurs:

- The discovered configuration is automatically saved to the switch configuration file after a configurable amount of time. This automatic save functionality can be enabled or disabled.
- The administrator does a **write memory** command to save the discovered configuration to the switch configuration file.

If the discovered configuration is not saved to the switch configuration file, then the learned configuration is lost on the next switch reboot. However, when the switch boots up again without any saved configuration, Automatic Fabric is automatically started again.

When the configuration is saved, the Automatic Fabric global and per-port settings are also saved. For example, if Automatic Fabric is globally enabled for the switch, then the parameter for this setting is also saved to the switch configuration file. Then on the next switch reboot, Automatic Fabric is started again even though there is already an existing switch configuration file.

Consider the following when managing the discovered configuration:

- To stop the discovery process and retain what has been learned so far, use CLI commands to disable the global Automatic Fabric process. or specific options of the discovery process.
- Do not save the learned configuration to have the switch perform the Automatic Fabric discovery and configuration each time the switch is rebooted.
- Manual configuration takes precedence over automatic discovery and configuration. For example, when the automatic IP protocol configuration is removed from an IP interface, the interface becomes eligible for automatic configuration again. However, if the IP protocol configuration was manually

applied to the interface, the interface does not become eligible for automatic IP configuration when the manual configuration is removed.

- The UNP SPB access port configuration resulting from the SPB discovery process is saved to the configuration file unless traffic is active on the port.

For more information, see “Saving the Discovered Configuration” on page 15-27.

# Automatic Fabric Discovery Examples

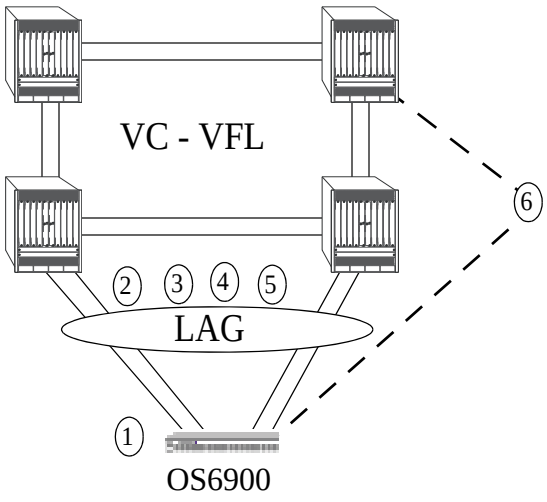
This section contains the following Automatic Fabric discovery examples:


- "Automatic Fabric Configured in the Network Core" on page 15-17.
- "Manual Configuration of the Network Core for LACP, SPB, and MVRP" on page 15-18.
- "Automatic Fabric Process for Automatic IP Configuration" on page 15-19.


## Automatic Fabric Configured in the Network Core

In this example, the network core is manually configured to be in Automatic Fabric mode on a subset of ports. When user enables the discovery window, auto discovery is triggered for discovery time window.

Virtual Chassis in core with auto fabric enabled on a subset of ports.




**Automatic Fabric in the Core**

**1** OS6900 with no configuration file, Automatic Fabric enabled by default. The switch has multiple connections to the core (which has Automatic Fabric enabled on the connected ports).

**2** LLDP exchanges port properties and automatically discovers LACP ports.

**3** LACP with the same admin key is exchanged. Multiple ports with the same admin key are detected and a link aggregate is formed and configured on both the core and edge switches.

**4** After the LACP discovery window expires, the SPB discovery starts. SPB BVLANs and control BVLANs are exchanged and adjacencies are saved.

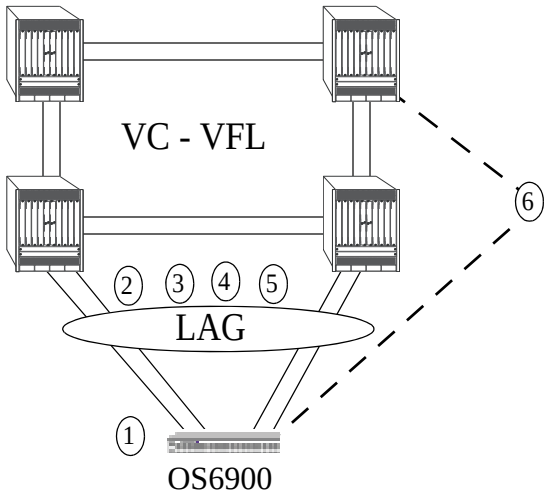
**5** MVRP control frames are exchanged for all non-BVLANs. VLANs received through MVRP frames are associated with the ports on which MVRP frames are received.

**6** The automatically discovered configuration remains in switch memory on the OS6900 as well as on the core switches, but a manual **write memory** command must be entered to make it permanent in the **vcboot.cfg** file and saved across switch reboots if the **auto-fabric config-save admin-state** is not enabled. Ports which already have a configuration are not eligible for automatic discovery on the next reboot.

# Manual Configuration of the Network Core for LACP, SPB, and MVRP

In this example, the network core is not configured for Automatic Fabric. The LACP, SPB, and MVRP protocols have been manually configured on the core.

Virtual Chassis in core with LACP, SPB, and MVRP protocols manually configured.




**No Automatic Fabric in the Core**

**1** OS6900 with no configuration file, Automatic Fabric is enabled by default. The switch has multiple connections to the core and the core has LACP, MVRP, and SPB manually configured, no Automatic Fabric is enabled on the core switches.

**2** LLDP runs on the edge switch, the core may or may not have LLDP enabled.

**3** LACP is manually configured on the core and sends LACP frames to the OS6900. The OS6900 honors the LACP frames since it is running Automatic Fabric and forms a link aggregate of ports with the same admin key. There could be multiple or single link aggregate groups based on the admin key advertised.

**4** After the LACP discovery window expires, the SPB discovery starts. SPB BVLANs and control BVLANs are exchanged and adjacencies are saved.

**5** MVRP control frames are exchanged for all non-BVLANs. VLANs received through MVRP frames are associated with the ports on which MVRP frames are received.

**6** The automatically discovered configuration remains in switch memory on the OS6900, but a manual **write memory** command must be entered to make it permanent in the **vcboot.cfg** file and saved across switch reboots if the **auto-fabric config-save admin-state** is not enabled. Ports which already have a configuration are not eligible for automatic discovery on the next reboot.

# Automatic Fabric Process for Automatic IP Configuration

When an IP interface is automatically configured for OSPF or IS-IS routing, the interface initially operates in passive mode. This means that the interface listens for Hello PDUs from neighbor switches to detect and configure OSPF neighbors or IS-IS adjacencies. The interface does not initially transmit Hello PDUs.

## Automatic OSPF Configuration

OSPF neighbors are detected through Hello packets received from neighbor switches. From these packets, the Area ID, Hello interval, and Dead interval values are learned and used to configure the OSPF interface. Both OSPFv2 and OSPFv3 learn areas and neighbors in a similar way.

The following scenarios are general examples of the automatic IP configuration process for the OSPF protocol.

**Two Automatic Configuration Routers**

- Both routers listen for Hello packets.
- Since neither router sends Hello packets in this scenario, no neighbors are learned.

**One Configured Router and One Automatic Configuration Router**

- The configured router sends the default Hello packets.
- The automatic configuration router receives the Hello packets and sends Hello packets with the learned information.
- The configured router receives Hello packets from the automatic configuration router and both routers become neighbors.
- The interface and area information is synchronized on the automatic configuration router.

## Automatic IS-IS Configuration

The IS-IS automatic configuration process is similar to the OSPF process in that adjacencies are learned from Hello packets received from neighbor switches. In addition, the following items apply specifically to the building of IS-IS adjacencies:

- An Area ID of 0.0.0.0 is used to help learn L2 adjacencies.
- Areas and level (L1, L2, L1L2) are learned from the received Hellos.
- The Hello time, Hello interval, and multiplier values are not learned from the received Hello packets. Instead, the automatic configuration process uses the default IS-IS holding time (27 seconds for non-DIS and 9 seconds for DIS) to derive the needed values.

Both IS-IS IPv4 and IS-IS IPv6 learn areas and neighbors in a similar way.

The following scenarios are general examples of the automatic IP configuration process for the OSPF protocol.

**Two Automatic Configuration Routers**

- Both routers listen for IS-IS Hello packets.
- Since neither router sends IS-IS Hello packets in this scenario, no neighbors are learned.

**One Configured Level 1 Router and One Automatic Configuration Router**

- The configured router transmits default Level 1 IS-IS Hello packets.
- The automatic configuration router receives IS-IS Hello packets and sends IS-IS Hello packets with the learned information.
- The configured router receives the IS-IS Hello packets and the routers become Level 1 adjacent.
- The interface, area, and level information is retained on the automatic configuration router.

# Interaction with Other Features

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

## System

When the Automatic Fabric feature is enabled there may be periodic changes to the switch configuration. This will cause the Running Configuration to display as “NOT SYNCHRONIZED” even after manually synchronizing CMMs.

## LLDP

- The 802.1AB Link Layer Discovery Protocol (LLDP) has been enhanced to detect peer device ports connected on boot up using a proprietary TLV and LLDP PDU exchanges. LLDP discovery will help to detect a set of ports connected to a neighbor device so that a link aggregation can be formed on the detected set of ports if LACP negotiation succeeds.
- If a port is brought up after Automatic Fabric has run and Automatic Fabric is enabled on that port, LLDP exchanges are used to determine if the port is connected to the same device so that the already connected port and subsequent new ports can form an aggregate.

## LACP

An Automatic Fabric discovery switch will be able to learn and configure Link Aggregation Control Protocol (LACP) link aggregates from any 802.3AD compliant and already configured switch.

- If an LACP frame is received on a non-aggregate port that has Automatic Fabric enabled, the Automatic Fabric discovery window is started provided there are no SPB adjacencies and no MVRP registrations on the port.
- If a neighbor device is manually configured for LACP with lesser ports than the number of connected ports between devices, then the rest of the ports in an automatic discovery enabled device will join/ form a back up LACP configuration.

During the LACP discovery and configuration process, the following scenarios are handled:

- **Neighbor is already configured with LACP**—Linkagg will detect LACP PDUs on the ports and map them to different neighbor devices based on admin key, system ID and priority received in the PDUs. This allows an Automatic Fabric enabled port to join an already formed aggregate or a new aggregate ID.
- **Neighbor device is also booting up with this device (max aggregate size not exceeded)**—If there are fewer ports than the maximum possible size of an aggregate then all ports are chosen and an aggregate is formed.
- **Neighbor device is also booting up with this device (max aggregate size exceeded)—**Connecting more ports than is supported for a link aggregate is not supported. The number of physical connections should not be greater than the maximum number of link aggregate ports supported for the OmniSwitch.
- **Device is already up and new port comes up—**In this case the device might already have a port/ aggregate which is connected to the same neighbor device, then the port joins the aggregate which is already formed or both ports will form a new aggregate.

Upon writing the automatically discovered configuration to the configuration file and rebooting, the automatically discovered link aggregate will become a manually configured link aggregate.

## MVRP and Spanning Tree

The Spanning Tree (STP) mode for the switch is automatically changed based on the sequence of configuration steps taken to globally enable or disable Automatic Fabric (AF) and the AF Multiple VLAN Registration Protocol (MVRP) discovery process.

- The following table shows the change made to the AF MVRP status and the Spanning Tree (STP) mode when the global AF status is changed through the **auto-fabric admin-state** command:

| Current Status/Mode | | Global AF Command Option (Enable /Disable) | After Global AF Status Change | |
|---|---|---|---|---|
| **AF MVRP Status** | **STP Mode** | | **AF MVRP Status** | **STP Mode** |
| Disabled | X | X | Disabled | No change |
| Enabled | Flat | Enable | Enabled | No change |
| Enabled | Per-VLAN | Enable | Disabled with a warning. | No change |
| Enabled | Flat | Disable | Enabled but does not apply due to the change of the global AF status to disabled. | Changed to per-VLAN if the STP mode was set to the flat mode by AF. An SNMP trap is sent and a warning logged when STP mode is changed. |
| Enabled | Per-VLAN | Disable | No change | No change |

- The following table shows the change made to the AF MVRP discovery status and the STP mode when the AF MVRP discovery status is changed through the **auto-fabric protocols** command:

| Current Status/Mode | | MVRP AF Command Option (Enable /Disable) | After MVRP AF Status Change | |
|---|---|---|---|---|
| **AF Global Status** | **STP Mode** | | **AF MVRP Status** | **STP Mode** |
| Enabled | Flat | Enable | Enabled | No change |
| Enabled | Per-VLAN | Enable | Enabled, if not already enabled. An SNMP trap is sent and a warning logged when STP mode is changed to flat mode. | Flat |

| Current Status/Mode | | MVRP AF Command Option (Enable /Disable) | After MVRP AF Status Change | |
|---|---|---|---|---|
| **AF Global Status** | **STP Mode** | | **AF MVRP Status** | **STP Mode** |
| Enabled | Flat | Disable | Disabled, if not already disabled. An SNMP trap is sent and a warning logged when STP mode is changed to per-VLAN mode. | Changed to Per-VLAN if the STP mode was set to flat mode by AF. |
| Enabled | Per-VLAN | Disable | Disabled | No change |
| Disabled | X | X | Same as AF MVRP command option. | No change |

## SPB

- If there are any BVLANs manually configured that are not in the range of 4000-4015, Shortest Path Bridging (SPB) discovery will not run.
- If there are any standard VLAN IDs configured in the 4000-4015 range, SPB discovery will not run.
- SPB automatic discovery occurs after Virtual Chassis setup and LACP discovery.
- All ports or Automatic Fabric link aggregates will be considered SPB network ports. An SPB network port can be a single port or a link aggregate of ports. IS-IS IIH (Hello) PDUs will be sent out on all ports to discover SPB-aware devices. SPB will not operate if LACP aborts discovery for any reason.
- At the end of the SPB discovery period, all ports that do not have adjacencies will no longer be considered network (backbone facing) ports and will not be used by ISIS-SPB or become members of the default BVLAN IDs created by Automatic Fabric.
- When a port that is enabled for automatic discovery has its first link up event or a new link aggregate is formed, SPB will attempt to form an adjacency. If no adjacency is found after four Hello time periods, the port will not be treated as an SPB backbone port, unless manually configured.
- If at least one SPB adjacency is formed on the switch, an attempt will be made to convert ports that are not part of the adjacency to UNP access ports.

## Virtual Chassis

- Automatic Fabric cannot be used to create a VFL for a Virtual Chassis.
- Automatic Fabric will only run after the Virtual Chassis setup is complete.

## VRF

- Automatic IP protocol configuration is supported and will start in any max profile VRFs. Up to 64 max profiles are allowed.
- Automatic IP protocol configuration is not supported in low profile VRFs. This is due to the fact that low profiles do not support any routing protocols.

## UNP Dynamic SAPs

- A Service Access Point (SAP) is dynamically created for ports that are automatically converted to UNP access ports through the SPB discovery and configuration process.
- The SAP associated with the first port that joins a link aggregate is applied to the link aggregate.
- When an Automatic Fabric port is converted to a Universal Network Profile (SP access port and the discovered configuration is saved, the access port configuration is not written to the boot file unless traffic is detected on that port.
- Access port configuration is reverted and the entire discovery cycle will be attempted again if any of the following events occur:
  - An Automatic Fabric LACP discovery LLDP TLV is received.
  - A synchronization LLDP TLV is received.
  - A port flap is observed and the UNP SPB access port has not received any traffic on the port.
- Removing the UNP dynamic SAP configuration from a UNP SPB access port, moves the port into a default state. In other words, the port becomes eligible to participate again in the Automatic Fabric process.

# Configuring Automatic Fabric

This section describes commands to configure the Automatic Fabric capability on an OmniSwitch.


- "Enabling or Disabling Automatic Fabric" on page 15-25
- "Configuring the Discovery Status for Specific Protocols" on page 15-26
- "Configuring the Discovery Interval" on page 15-26
- "Manually Starting the Discovery Process" on page 15-27
- "Saving the Discovered Configuration" on page 15-27
- "Configuring the Default SPB SAP profile" on page 15-28


## Enabling or Disabling Automatic Fabric

Automatic Fabric is enabled globally for the switch when any of the following events occur:

- The switch boots up with no configuration file or the configuration file size is zero.
- The switch boots up with an existing configuration file that has the following Automatic Fabric entry:

```
-> show configuration snapshot auto-fabric
! Dynamic auto-fabric:
auto-fabric admin-state enable
```

- The **auto-fabric admin-state** command is used with the **enable** parameter option while the switch is up and running. For example:

```
-> auto-fabric admin-state enable
```

Automatic Fabric is also enabled on a per port basis using the **auto-fabric admin-state** command with the **interface** parameter. For example:

```
-> auto-fabric interface 1/1 admin-state enable
```

It is important to note that the port level setting for Automatic Fabric overrides the global switch setting. For example, if Automatic Fabric is globally enabled for the switch but disabled on port 1/2, Automatic Fabric does not activate automatic discovery on that port.

To disable Automatic Fabric globally or on a per-port basis, use the **auto-fabric admin-state** command with the **disable** parameter option. For example:

```
-> auto-fabric interface 1/1 admin-state disable
-> auto-fabric admin-state disable
```

When Automatic Fabric is globally disabled for the switch, the following configuration settings are removed unless they were previously saved to the switch configuration file:

- Spanning Tree is set back to the default 1x1 mode. This only occurs if there are no VLAN registrations on any port or link aggregate.
- SPB is globally disabled, which removes BVLANs 4000-4015 and administratively disables SPB. This only occurs if there are no SPB adjacencies formed on any ports or link aggregates.

- Automatic Fabric strops trying to learn IP routing protocols and neighbors on interfaces not already configured with a routing protocol. The configuration for IP interfaces on which routing protocols were previously discovered is not removed.

Use the **show auto-fabric config** command and the **show auto-fabric config interface** command to verify the Automatic Fabric status for the switch and switch ports.

## Configuring the Discovery Status for Specific Protocols

Discovery for the LACP, SPB, and MVRP protocols can be enabled or disabled globally or on a per-port basis. For the IP routing protocols (OSPFv2, OSPFv3, and IS-IS), the discovery status is set on a global basis and is used to specify which IP protocols the switch will attempt to detect and configure on active IP interfaces.

When the Automatic Fabric discovery window is started, only those protocols that are enabled for discovery are processed. By default, the discovery status for all the Automatic Fabric protocols, except MVRP, is enabled. MVRP is disabled by default.

To globally enable or disable protocol discovery, use the **auto-fabric protocols** command. For example:

```
-> auto-fabric protocols lacp admin-state disable
-> auto-fabric protocols mvrp admin-state enable
-> auto-fabric protocols ip ospfv2 admin-state enable
-> auto-fabric protocols ip ospfv3 admin-state disable
-> auto-fabric protocols ip isis admin-state disable
```

In this example, discovery is only attempted for MVRP and OSPFv2. Discovery for LACP, OSPFv3, and IS-IS is disabled.

To enable or disable protocol discovery on a specific port, use the **auto-fabric protocols** command with the **interface** parameter. For example:

```
-> auto-fabric protocols spb interface 1/3 admin-state disable
-> auto-fabric protocols lacp interface 1/10-15 admin-state disabled
```

It is important to note that the port level setting for Automatic Fabric discovery overrides the global switch setting. For example, if discovery is globally enabled for SPB but disabled on port 1/2, Automatic Fabric will not include that port in the discovery window for SPB.

The **auto-fabric protocols** command is also used to enable or disable loopback detection. For example:

```
-> auto-fabric protocols loopback-detection admin-state disable
```

When enabled, Loopback Detection is activated on UNP SPB access ports that are bound to a SAP.

## Configuring the Discovery Interval

When Automatic Fabric is enabled for the switch, the discovery interval time specifies how often the switch will automatically start the Automatic Fabric discovery process. For example, if this value is set to 30 minutes, every 30 minutes the switch will start the discovery process.

Setting the discovery interval value to a time that is more than twice the value of the switch MAC address aging time is recommended. For example, if the MAC address aging time is set to 5 minutes, set the discovery interval time to 11 minutes. Otherwise, inactive MAC addresses may not have aged out on Automatic Fabric ports by the next discovery interval start time.

By default, the discovery interval timer is set to zero, which means the timer is disabled. However, when a switch boots up without a configuration file, discovery is automatically started for a one time, initial run even when the interval timer is disabled.

To change the discovery interval time, use the use the **auto-fabric discovery-interval** command. For example:

```
-> auto-fabric discovery-interval 60
```

In this example, the timer value is changed to 60 minutes. So every 60 minutes the switch will automatically start discovery for the Automatic Fabric protocols.

## Manually Starting the Discovery Process

It is possible to manually start the Automatic Fabric Discovery process at any time after the switch boots up and there is no active discovery process (the discovery window is closed). To manually start the discovery process, use the **auto-fabric discovery start** command. For example:

```
-> auto-fabric discovery start
```

## Saving the Discovered Configuration

The LACP, SPB, MVRP, and IP protocols configuration can be saved to the switch configuration file. For MVRP, only the CLI configuration is saved. MVRP VLANs must be converted to static VLANs to be saved.

The discovered configuration remains in switch memory until one of the following occurs:

- The administrator does a **write memory** command to save the discovered configuration to the switch configuration file.
- The discovered configuration is automatically saved to the switch configuration file after a configurable amount of time. This automatic save functionality can be enabled or disabled.

By default the automatic save function is disabled. Use the **auto-fabric config-save admin-state** command to enable automatically saving the discovered configuration to the switch configuration file. For example:

```
-> auto-fabric config-save admin-state enable
```

Once this capability is enabled, the switch will save the discovered configuration every 300 seconds (the default). To change this time interval, use the **auto-fabric config-save interval** command. For example, the following command configures the switch to save the discovered configuration to the switch configuration file every 600 seconds:

```
-> auto-fabric config-save interval 600
```

If the discovered configuration is not saved to the switch configuration file, the discovered configuration is lost on the next switch reboot.

## Configuring the Default SPB SAP profile

The default SAP profile can be configured for the switch or for a specific port or range of ports. There are two options for this type of profile: automatic VLAN (the default) or single service.

When this option is set to automatic VLAN, a SAP is automatically created for each VLAN tag received on the port. The automatic VLAN profile is recommended for tagged traffic. The single service profile is recommended for untagged traffic.

To change the global default SAP profile setting for the switch, use the **auto-fabric protocols spb default-profile** command. For example:

```
-> auto-fabric protocols spb default-profile single-service
-> auto-fabric protocols spb default-profile auto-vlan
```

To set the default SAP profile for a specific port or range of ports on the switch, use the **auto-fabric protocols spb set-profile** command. For example:

```
-> auto-fabric protocols spb set-profile single-service interface 1/1/1
-> auto-fabric protocols spb set-profile auto-vlan interface 1/2/1-4
```

**Note.** The SAP profile configured for the port or range of ports will over ride the default SAP profile configured for the switch. By default the SAP profile is “auto-vlan”.

# Displaying the Automatic Fabric Configuration

You can use the following Command Line Interface (CLI) **show** commands to display the current configuration and status of the Automatic Fabric feature:

| | |
|---|---|
| **show auto-fabric config** | Displays details about the globally configured and operational parameters. |
| **show auto-fabric config interface** | Displays the Automatic Fabric port configuration applied on interfaces. |

In addition to the **show auto-fabric** commands, the **show vlan** and **show linkagg** commands indicate in the display output which VLANs and link aggregates were created through the Automatic Fabric discovery process. For example:

```
-> show vlan
 vlan    type   admin   oper    ip    mtu          name
------+-------+-------+------+------+------+-----------------
1      std       Ena     Ena   Ena    1500    RCFG VLAN
4000   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4001   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4002   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4003   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4004   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4005   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4006   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4007   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4008   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4009   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4010   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4011   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4012   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4013   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4014   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4015   spb       Ena     Dis   Dis    1524    AutoFabric 7/6/2016 09:19:03
4094   vcm       Ena     Dis   Dis    1500    VCM IPC

-> show configuration snapshot linkagg
! Link Aggregate:
linkagg lacp agg 125 size 8 hash tunnel-protocol admin-state enable
linkagg lacp agg 125 name "Created by Auto-Fabric on Fri Jul 15 01:02:44 2016
"
linkagg lacp agg 125 actor admin-key 65535
linkagg lacp port 1/3/7 actor admin-key 65535
linkagg lacp port 1/3/8 actor admin-key 65535
linkagg lacp port 1/3/11 actor admin-key 65535
linkagg lacp port 1/3/12 actor admin-key 65535
```

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

# 16 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 16-9.
- Selecting an NTP server for the NTP client and modifying settings for communicating with the server. See "NTP Servers" on page 16-10.
- Enabling authentication in NTP negotiations. See "Using Authentication" on page 16-12.

## 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<br>minpoll: 6<br>maxpoll: 10<br>prefer: no<br>key: 0<br>burst: no burst<br>iburst: no iburst |
| Used to activate client | **ntp client** | disabled |

| Parameter Description | Command | Default Value/Comments |
|---|---|---|
| Used to activate NTP client broadcast mode | **ntp broadcast-client** | 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
```

NTP server configuration can also be done with hostname/FQDN. For example:

```
-> ntp server clock3.ovcirrus.com
```

**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
IP address          = clock3.ovcirrus.com [123.108.200.124],
Host mode           = client,
Peer mode           = server,
Prefer              = no,
Version             = 4,
Key                 = 0,
Stratum             = 2,
Minpoll             = 6 (64 seconds),
Maxpoll             = 10 (1024 seconds),
Poll                = 1024 seconds,
when                = 283 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,
```

**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
Legends: + active peer, - passive peer, = client, * current system peer,
         ^ broadcast server, ' broadcast client
Mode IP Address       Ver   Key   St   when   poll   reach   Delay   Offset      Disp
--------------------+-----+----+-----+------+------+------+-------+----------+-----------
*    198.206.181.70    4    0     2    895    1024   377     0.167   0.323       0.016
=    198.206.181.123   4    0     16   591    1024   377     0.000   0.000       0.000
```

**5** You can check the client configuration using the **show ntp status** command, as shown:

```
-> show ntp status
Current time:                   Mon, Jan 21 2019  7:31:04.685 (UTC),
Last NTP update:                Mon, Jan 21 2019  7:30:10.160 (UTC),
Server reference:               10.10.10.10,
Client mode:                    enabled,
Broadcast client mode:          disabled,
Broadcast delay (microseconds): 4000,
Stratum:                        4,
Maximum Associations Allowed:   32,
Authentication:                 enabled,
VRF Name:                       default
```

# NTP Overview

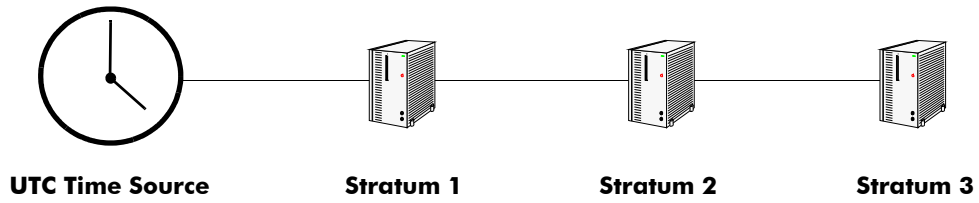
Network Time Protocol (NTP) is used to synchronize the time of a computer client or server to another server or reference time source, such as a radio or satellite receiver. It provides client time accuracies within a millisecond on LANs, and up to a few tens of milliseconds on WANs relative to a primary server synchronized to Universal Coordinated Time (UTC) (via a Global Positioning Service receiver, for example). Typical NTP configurations utilize multiple redundant servers and diverse network paths in order to achieve high accuracy and reliability. Some configurations include cryptographic authentication to prevent accidental or malicious protocol attacks.

It is important for networks to maintain accurate time synchronization between network nodes. The standard timescale used by most nations of the world is based on a combination of UTC (representing the Earth's rotation about its axis), and the Gregorian Calendar (representing the Earth's rotation about the Sun). The UTC timescale is disciplined with respect to International Atomic Time (TAI) by inserting leap seconds at intervals of about 18 months. UTC time is disseminated by various means, including radio and satellite navigation systems, telephone modems, and portable clocks.

Special purpose receivers are available for many time-dissemination services, including the Global Position System (GPS) and other services operated by various national governments. For reasons of cost and convenience, it is not possible to equip every computer with one of these receivers. However, it is possible to equip some computers with these clocks, which then act as primary time servers to synchronize a much larger number of secondary servers and clients connected by a common network. In order to do this, a distributed network clock synchronization protocol is required which can read a server clock, transmit the reading to one or more clients, and adjust each client clock as required. Protocols that do this include NTP.

## Stratum

Stratum is the term used to define the relative proximity of a node in a network to a time source (such as a radio clock). Stratum 1 is the server connected to the time source itself. (In most cases the time source and the stratum 1 server are in the same physical location.) An NTP client or server connected to a stratum 1 source would be stratum 2. A client or server connected to a stratum 2 machine would be stratum 3, and so on, as demonstrated in the diagram below:




The farther away from stratum 1 a device is, the more likely there will be discrepancies or errors in the time adjustments done by NTP. A list of stratum 1 and 2 sources available to the public can be found on the Internet.

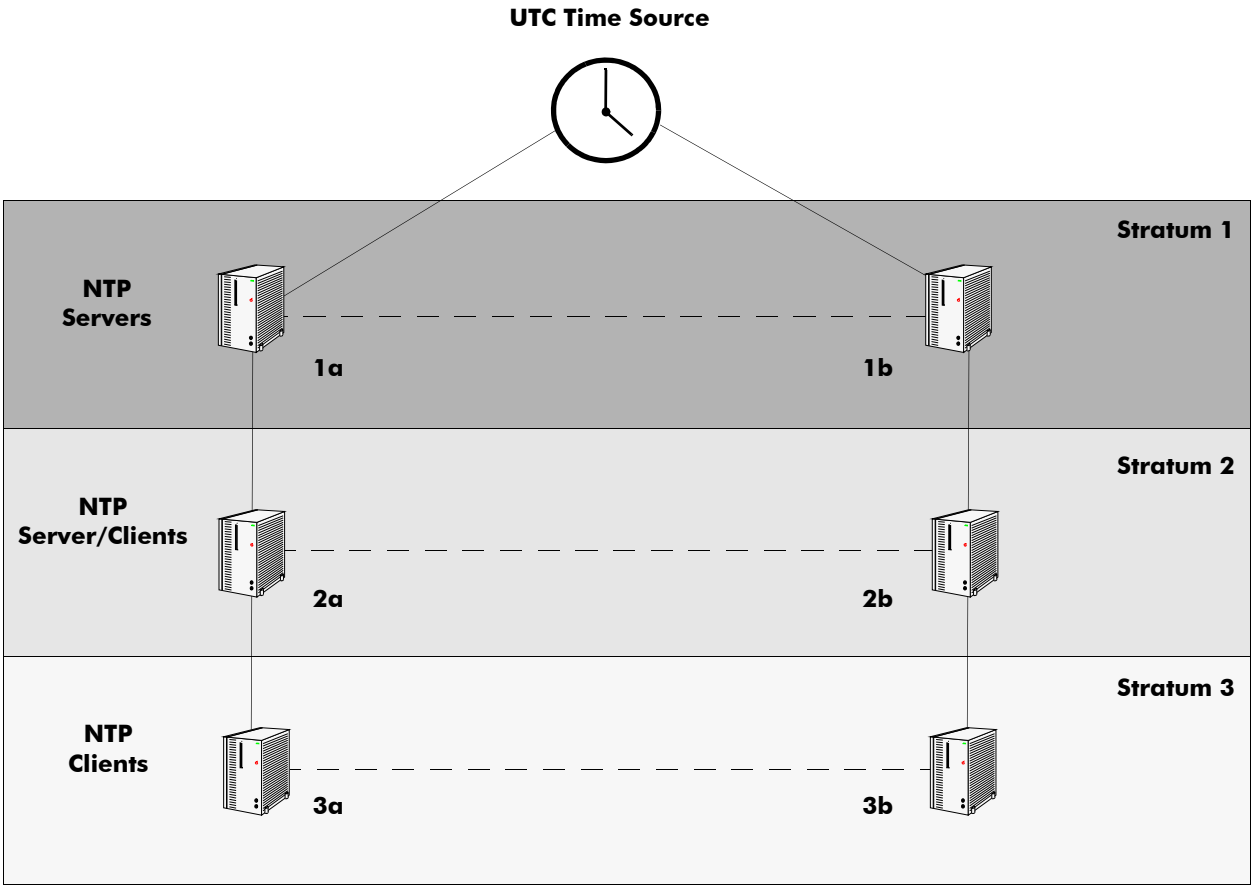
**Note.** It is not required that NTP be connected to an officially recognized time source (for example, a radio clock). NTP can use any time source to synchronize time in the network.

## Using NTP in a Network

NTP operates on the premise that there is one true standard time (defined by UTC), and that if several servers claiming synchronization to the standard time are in disagreement, then one or more of them must be out of synchronization or not functioning correctly. The stratum gradiation is used to qualify the accuracy of a time source along with other factors, such as advertised precision and the length of the network path between connections. NTP operates with a basic distrust of time information sent from other network entities, and is most effective when multiple NTP time sources are integrated together for checks and crosschecks. To achieve this end, there are several modes of operation that an NTP entity can use when synchronizing time in a network. These modes help predict how the entity behaves when requesting or sending time information, listed below:

- A switch can be a client of an NTP server (usually of a lower stratum), receiving time information from the server but not passing it on to other switches.
- A switch can be a client of an NTP server, and in turn be a server to another switch or switches.
- A switch (regardless of its status as either a client or server) must be peered with another switch. Peering allows NTP entities in the network of the same stratum to regard each other as reliable sources of time and exchange time information.
- The OmniSwitch by default will act as an NTP server and be able to respond to NTP client requests, and establish a client or 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 or SHA1 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 and SHA1 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
4  SHA1 33ba92508c0fd90dd1e87310e04fd32c48ed2dcd  # SHA1 key
5  SHA1 a787609efdeac26766810e7b507934e6d9da78e4  # SHA1 key
6  SHA1 90b4043bd301ddbf2b375f4574075ba469e690e9  # SHA1 key
```

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 IDs 2 and 14 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.

The key IDs 4, 5, and 6 indicates SHA1 keys.

The OmniSwitch establishes which key pair it is using for authentication by specifying a key ID for each NTP server configured.

For configuration information see “Using Authentication” on page 16-12.

# 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
```

**Note.** NTP client will not synchronize with an unsynchronized NTP server (Stratum 16).

### 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 0.pool.ntp.org
```

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
```

## 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. The minimum poll interval defaults to 6 (64 s), but can be decreased by the minpoll option to a lower limit of 3 (8 s) and an upper limit of 17 (36.4h).

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 Maximum Poll Time

The maximum poll specifies the maximum polling interval for NTP messages, in seconds. This number is determined by raising 2 to the power of the number entered. The maximum poll interval defaults to 10 (1,024 s), but can be increased by the maxpoll option to an upper limit of 17 (36.4 h) and a lower limit of 3 (8 s). The maxpoll must not be less than the minpoll value.

For example, to set the maximum poll time to 256 seconds, enter the following:

```
-> ntp server 1.1.1.1 maxpoll 8
```

This would set the maximum poll time to $2^8$ = 256 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
```

## Enabling Burst and iBurst Mode for NTP Server

The burst mode allows the exchange of eight NTP packets (instead of one) when the server is reachable and at each poll interval to achieve faster synchronization. The spacing between the first and the second packet is 16 seconds to allow a modem call to complete, while the spacing between the remaining packets is 2 seconds. This improves timekeeping quality with the server command.

To enable burst mode, use **ntp server** command with **burst** keyword, as shown:

```
-> ntp server 1.1.1.1 burst
```

The iburst mode allows immediate exchange of eight NTP packets (instead of one) when the server is unreachable and at each poll interval, to achieve faster initial synchronization acquisition. As long as the server is unreachable, the spacing between the packets is 16 seconds to allow a modem call to complete. Once the server is reachable, the spacing between the packets is 2 seconds. This helps speed the initial synchronization acquisition with the server command.

To enable iburst mode, use **ntp server** command with **iburst** keyword, as shown:

```
-> ntp server 1.1.1.1 iburst
```

## Specifying Preempt Mode

This enables the preemption mode for the server rather than the default persistent. The specified server is marked unavailable for selection if any error (authentication failure) is detected on a connection between the local device and reference clock. The server is marked available for selection if no other connections are available and no error is detected on the connection between the local device and reference clock.

```
-> ntp server 1.1.1.1 preempt
```

# 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 keys. (This file should be obtained from the server administrator. For more information on the authentication file, see "Authentication" on page 16-8.)

Once both the client and server share a common encryption key, the key identification for the NTP server must be specified on and labeled as trusted on the client side. The Omniswitch will then use authentication. Key files must reside in **/flash/network/ntp.keys**.

In order to generate a key file, access to a Solaris/Unix environment is recommended. Also recommended is the ntp-keygen utility in Unix to generate the key file. As an alternative, the keys can be manually created.

## Setting the Key ID for the NTP Server

Enabling authentication requires the following steps:

**1** Make sure the key file is located in the **flash/network** 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 key is loaded into the switch memory by issuing the **ntp key load** command, as shown:

```
-> ntp key load
```

**3** Enable server authentication and 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 authentication. For example, to specify key identification number 2 for an NTP server with an IP address of 1.1.1.1, enter:

```
-> ntp authentication enable
-> 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 2 to trusted status, enter the following:

```
-> ntp key 2 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 16-3.

# A Software License and Copyright Statements

This appendix contains ALE USA, Inc. and third-party software vendor license and copyright statements.

## ALE USA, Inc. License Agreement

### ALE USA, INC. 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 ALE USA, Inc. ALE USA, Inc. 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 ALE USA, Inc. 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. **ALE USA, Inc.'s Rights.** Licensee acknowledges and agrees that the Licensed Materials are the sole property of ALE USA, Inc. 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 ALE USA, Inc. 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.** ALE USA, Inc. considers the Licensed Files to contain valuable trade secrets of ALE USA, Inc., the unauthorized disclosure of which could cause irreparable harm to ALE USA, Inc. 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 ALE USA, Inc. harmless from any claim, lawsuit, legal proceeding, settlement or judgment (including without limitation ALE USA, Inc.'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.** ALE USA, Inc. 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. ALE USA, Inc. 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 ALE USA, Inc. for either replacement or, if so elected by ALE USA, Inc., 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 ALE USA, INC. 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.** ALE USA, Inc.'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 ALE USA, Inc. for the Licensed Materials. IN NO EVENT SHALL ALE USA, Inc. BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL, OR EXEMPLARY DAMAGES OR LOST PROFITS, EVEN IF ALE USA, Inc. 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 ALE USA, Inc. and Licensee, if any, ALE USA, Inc. is under no obligation to maintain or support the copies of the Licensed Files made and distributed hereunder and ALE USA, Inc. 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 ALE USA, Inc. and certifying to ALE USA, Inc. 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. ALE USA, Inc. may terminate this License Agreement upon the breach by Licensee of any term hereof. Upon such termination by

ALE USA, Inc., Licensee agrees to return to ALE USA, Inc. 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 ALE USA, Inc.'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 ALE USA, Inc. by certain third party licensors. Some third party licensors are third part beneficiaries to this License Agreement with full rights of enforcement. Please refer to the section entitled "Third Party Licenses and Notices" on page -4 for the third party license and notice terms.

# Third Party Licenses and Notices

Legal Notices applicable to any software distributed alone or in connection with the product to which this document pertains, are contained in files within the software itself located at: **/flash/foss**.

Also, if needed, we provide all FOSS (Free and Open Source Software) source code used in this release at the following URL: https://github.com/Alcatel-LucentEnterpriseData.

# B SNMP Trap Information

This appendix lists the supported SNMP traps, alerts, and MIBs along with their descriptions.

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


- "SNMP Traps Table" on page -2
- "chassisTrapsAlertNumber" on page -64
- "MIBS Table" on page -67
- "System Events" on page -73

# 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<br>ifAdminStatus<br>ifOperStatus | interface | The SNMP agent in the switch recognizes a failure in one of the communications links configured for the switch. |
| | **IfIndex**—A unique value, greater than zero, for each interface. It is recommended that values are assigned contiguously starting from 1. The value for each interface sub-layer must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.<br>**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).<br>**ifOperStatus**—The current operational state of the interface. The testing (3) state indicates that no operational packets can be passed. If ifAdminStatus is down (2) then ifOperStatus should be down(2). If ifAdminStatus is changed to up (1) then ifOperStatus should change to up (1) if the interface is ready to transmit and receive network traffic; it should change to dormant (5) if the interface is waiting for external actions (such as a serial line waiting for an incoming connection); it should remain in the down (2) state if and only if there is a fault that prevents it from going to the up (1) state; it should remain in the notPresent (6) state if the interface has missing (typically, hardware) components. | | | |
| 3 | linkUp | ifIndex<br>ifAdminStatus<br>ifOperStatus | interface | The SNMP agent in the switch recognizes that one of the communications links configured for the switch has come up. |
| | **IfIndex**—A unique value, greater than zero, for each interface. It is recommended that values are assigned contiguously starting from 1. The value for each interface sub-layer must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.<br>**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).<br>**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. | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 4 | authenticationFailure | none | snmp | The SNMP agent in the switch has received a protocol message that is not properly authenticated. |
| 5 | entConfigChange | none | module | An entConfigChange notification is generated when the value of entLastChangeTime changes. It can be utilized by an NMS to trigger logical/physical entity table maintenance polls. An agent should not generate more than one entConfigChange 'notification-event' in a given time interval (five seconds is the suggested default). A 'notification-event' is the transmission of a single trap or inform PDU to a list of notification destinations. If additional configuration changes occur within the throttling period, then notification-events for these changes should be suppressed by the agent until the current throttling period expires. At the end of a throttling period, one notification-event should be generated if any configuration changes occurred since the start of the throttling period. In such a case, another throttling period is started right away. An NMS should periodically check the value of entLastChangeTime to detect any missed entConfigChange notification-events, e.g., due to throttling or transmission loss. |
| 6 | policyEventNotification | policyTrapEventDetailString<br>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.

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 7 | chassisTrapsStr | chassisTrapsStrLevel<br>chassisTrapsStrAppID<br>chassisTrapsStrSnapID<br>chassisTrapsStrfileName<br>chassisTrapsStrfileLineNb<br>chassisTrapsStrErrorNb<br>chassisTrapsStrcomments<br>chassisTrapsStrdataInfo | chassis | A software trouble report (STR) was sent by an application encountering a problem during its execution. |
| | **chassisTrapsStrLevel**—An enumerated value that provides the urgency level of the STR.<br>**chassisTrapsStrAppID**—The application identification number.<br>**chassisTrapsStrSnapID**—The subapplication identification number. You can have multiple snapIDs per Subapplication (task) but only one is to be used to send STRs.<br>**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.<br>**chassisTrapsStrfileLineNb**—Line number in the source file where the fault was detected. This is given by the C ANSI macro __LINE__.<br>**chassisTrapsStrErrorNb**—The fault identificator. The error number identifies the kind the detected fault and allows a mapping of the data contained in chassisTrapsdataInfo.<br>**chassisTrapsStrcomments**—Comment text explaining the fault.<br>**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. | | | |
| 8 | chassisTrapsAlert | physicalIndex<br>chassisTrapsObjectType<br>chassisTrapsObjectNumber<br>chassisTrapsAlertNumber<br>chassisTrapsAlertDescr | chassis | A notification that some change has occurred in the chassis. |
| | **physicalIndex**—The physical index of the involved object.<br>**chassisTrapsObjectType**—An enumerated value that provides the object type involved in the alert trap.<br>**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".<br>**chassisTrapsAlertNumber**—This number that identifies the alert among all the possible chassis alert causes.<br>**chassisTrapsAlertDescr**— The description of the alert matching ChassisTrapsAlertNumber. | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 9 | chassisTrapsStateChange | physicalIndex<br>chassisTrapsObjectType<br>chassisTrapsObjectNumber<br>chasEntPhysOperStatus | chassis | A status change was detected. (Note: Can take up to 10 seconds for operational status query to be reflected after state change trap is sent. |
| **physicalIndex**—The physical index of the involved object.<br>**chassisTrapsObjectType**—An enumerated value that provides the object type involved in the alert trap.<br>**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”.<br>**chasEntPhysOperStatus**—All modules (even empty slots) are in unknown state when the chassis first powers up. | | | | |
| 10 | chassisTrapsMacOverlap | physicalIndex<br>chasTrapMacRangeIndex | module | A MAC range overlap was found in the backplane eeprom. |
| **physicalIndex**—The physical index of the involved object.<br>**chasTrapMacRangeIndex**—The MAC range index of the involved object. | | | | |
| 11 | vrrpTrapNewMaster | vrrpOperMasterIpAddr | vrrp | The VRRP agent has transferred from the backup state to the master state. |
| **vrrpOperMasterIpAddr**—The master router’s real (primary) IP address. This is the IP address listed as the source in the VRRP advertisement last received by this virtual router. | | | | |
| 12 | vrrpTrapAuthFailure | vrrpTrapPacketSrc<br>vrrpTrapAuthErrorType | vrrp | A packet was received from the network whose authentication key conflicts with the switch’s authentication key or type. |
| **vrrpTrapPacketSrc**—The IP address of an inbound VRRP packet.<br>**vrrpTrapAuthErrorType**—Potential types of configuration conflicts. | | | | |
| 13 | healthMonModuleTrap | healthModuleSlot<br>healthMonRxStatus<br>healthMonRxTxStatus<br>healthMonMemoryStatus<br>healthMonCpuStatus | health | Indicates a module-level threshold was crossed. |
| **healthModuleSlot**—The (one-based) front slot number within the chassis.<br>**healthMonRxStatus**—Rx threshold status indicating if threshold was crossed or no change.<br>**healthMonRxTxStatus**—RxTx threshold status indicating if threshold was crossed or no change.<br>**healthMonMemoryStatus**—Memory threshold status indicating if threshold was crossed or no change.<br>**healthMonCpuStatus**—CPU threshold status indicating if threshold was crossed or no change. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 14 | healthMonPortTrap | healthPortSlot<br>healthPortIF<br>healthMonRxStatus<br>healthMonRxTxStatus | health | Indicates a port-level threshold was crossed. |
| **healthPortSlot**—The physical slot number for this port.<br>**healthPortIF**—The on-board interface number.<br>**healthMonRxStatus**—Rx threshold status indicating if threshold was crossed or no change.<br>**healthMonRxTxStatus**—RxTx threshold status indicating if threshold was crossed or no change. | | | | |
| 15 | healthMonCmmTrap | healthMonMemoryStatus, healthMonCpuStatus | health | This trap is sent when an NI memory or CPU threshold is crossed. |
| **healthMonMemoryStatus**—Memory threshold status indicating if threshold was crossed or no change.<br>**healthMonCpuStatus**—CPU threshold status indicating if threshold was crossed or no change. | | | | |
| 16 | bgpEstablished | bgpPeerLastError<br>or<br>bgpPeerState | bgp | The BGP routing protocol has entered the established state. |
| **bgpPeerLastError**—The last error code and subcode seen by this peer on this connection. If no error has occurred, this field is zero. Otherwise, the first byte of this two byte OCTET STRING contains the error code, and the second byte contains the subcode.<br>**bgpPeerState**—The BGP peer connection state. | | | | |
| 17 | bgpBackwardTransition | bgpPeerLastError<br>or<br>bgpPeerState | bgp | This trap is generated when the BGP router port has moved from a more active to a less active state. |
| **bgpPeerLastError**—The last error code and subcode seen by this peer on this connection. If no error has occurred, this field is zero. Otherwise, the first byte of this two byte OCTET STRING contains the error code, and the second byte contains the subcode.<br>**bgpPeerState**—The BGP peer connection state. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 18 | esmDrvTrapDropsLink | esmPortSlot<br>esmPortIF<br>ifInErrors<br>ifOutErrors<br>esmDrvTrapDrops | interface | This trap is sent when the Ethernet code drops the link because of excessive errors. |

**esmPortSlot**—The physical slot number for this Ethernet Port. The slot number has been added to be used by the private trap.
**esmPortIF**—The on-board interface number for this Ethernet port. The port number has been added to be used by the private trap.
**ifInErrors**—For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter caifIndexn occur at re-initialization of the management system and at other times as indicated by the value of ifCounterDiscontinuityTime.
**ifOutErrors**—For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system and at other times as indicated by the value of ifCounterDiscontinuityTime.
**esmDrvTrapDrops**— Partitioned port (separated due to errors).

| | | | | |
|---|---|---|---|---|
| 19 | portViolationTrap | ifIndex,<br>portViolationSource,<br>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 |

**ifIndex**—A unique value, greater than zero, for the interface.
**portViolationSource**—The source of the port violation. The source is the feature or module that has caused the violation - 1. Source Learning, 2. QOS Policy, 3. Net Sec, 4. UDLD, 5. NI Supervison (Fabric Stability). When there is no value the value is “0".
**portViolationReason**—The reason for the port violation. It is application specific, and indicates first Violation that happened on this port - 1. pvSLLpsShutDown, 2. pvSLLpsRestrict, 3. pvQosPolicy, 4. pvQosSpoofed, 5. pvQosBpdu, 6. pvQosBgp, 7. pvQosOspf, 8. pvQosRip, 9. pvQosVrrp, 10. pvQosDhcp, 11. pvQosPim, 12. pvQosDvmrp, 13. pvQosIsis, 14. pvQosDnsReply, 15. pvUdld.

| | | | | |
|---|---|---|---|---|
| 20 | dvmrpNeighborLoss | dvmrpInterfaceLocalAddress<br>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. |

**dvmrpInterfaceLocalAddress**—The IP address this system will use as a source address on this interface. On unnumbered interfaces, it must be the same value as dvmrpInterfaceLocalAddress for some interfaces on the system.
**dvmrpNeighborState**—State of the neighbor adjacency.

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 21 | dvmrpNeighborNotPruning | dvmrpInterfaceLocalAddress<br>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.<br>**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<br>alarmVariable<br>alarmSampleType<br>alarmValue<br>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.<br>**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.<br>**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.<br>**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.<br>**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<br>alarmVariable<br>alarmSampleType<br>alarmValue<br>alarmFallingThreshold | rmon | An Ethernet statistical variable has dipped below its falling threshold. The variable's falling threshold and whether it will issue an SNMP trap for this condition are configured by an NMS station running RMON. |

**alarmIndex**—An index that uniquely identifies an entry in the alarm table. Each such entry defines a diagnostic sample at a particular interval for an object on the device.
**alarmVariable**—The object identifier of the particular variable to be sampled. Only variables that resolve to an ASN.1 primitive type of INTEGER (INTEGER, Integer32, Counter32, Counter64, Gauge, or TimeTicks) may be sampled.
**alarmSampleType**—The method of sampling the selected variable and calculating the value to be compared against the thresholds. If the value of this object is absoluteValue (1), the value of the selected variable will be compared directly with the thresholds at the end of the sampling interval. If the value of this object is deltaValue (2), the value of the selected variable at the last sample will be subtracted from the current value, and the difference compared with the thresholds.
**alarmValue**—The value of the statistic during the last sampling period. For example, if the sample type is deltaValue, this value will be the difference between the samples at the beginning and end of the period. If the sample type is absoluteValue, this value will be the sampled value at the end of the period.
**alarmFallingThreshold**—A threshold for the sampled statistic. When the current sampled value is less than or equal to this threshold, and the value at the last sampling interval was greater than this threshold, a single event will be generated. A single event will also be generated if the first sample after this entry becomes valid is less than or equal to this threshold and the associated alarmStartupAlarm is equal to fallingAlarm (2) or risingOrFallingAlarm (3).

| | | | | |
|---|---|---|---|---|
| 24 | stpNewRoot | vStpNumber | stp | Sent by a bridge that became the new root of the spanning tree. |

**vStpNumber**—The Spanning Tree number identifying this instance.

| | | | | |
|---|---|---|---|---|
| 25 | stpRootPortChange | vStpNumber<br>vStpRootPortNumber | stp | A root port has changed for a spanning tree bridge. The root port is the port that offers the lowest cost path from this bridge to the root bridge. |

**vStpNumber**—The Spanning Tree number identifying this instance.
**vStpRootPortNumber**—The port ifindex of the port which offers the lowest cost path from this bridge to the root bridge for this spanning tree instance.

| | | | | |
|---|---|---|---|---|
| 26 | mirrorConfigError | mirmonPrimarySlot<br>mirmonPrimaryPort<br>mirroringSlot<br>mirroringPort<br>mirMonErrorNi<br>mirMonError | pmm | The mirroring configuration failed on an NI. This trap is sent when any NI fails to configure mirroring. Due to this error, port mirroring session will be terminated. |

**mirmonPrimarySlot**—Slot of mirrored or monitored interface.
**mirmonPrimaryPort**—Port of mirrored or monitored interface.
**mirroringSlot**—Slot of mirroring interface.
**mirroringPort**—Port of mirroring interface.
**mirMonErrorNi**—The NI slot number.
**mirMonError**—The Error returned by the NI which failed to configure Mirroring/Monitoring.

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 27 | mirrorUnlikeNi | mirmonPrimarySlot<br>mirmonPrimaryPort<br>mirroringSlot<br>mirroringPort<br>mirMonErrorNi | pmm | The mirroring configuration is deleted due to the swapping of different NI board type. The Port Mirroring session which was active on a slot cannot continue with the insertion of different NI type in the same slot. |
| **mirmonPrimarySlot**—Slot of mirrored or monitored interface.<br>**mirmonPrimaryPort**—Port of mirrored or monitored interface.<br>**mirroringSlot**—Slot of mirroring interface.<br>**mirroringPort**—Port of mirroring interface.<br>**mirMonErrorNi**—The NI slot number.<br>**mirMonError**—The Error returned by the NI which failed to configure Mirroring/Monitoring. | | | | |
| 28 | slbTrapOperStatus | slbTrapInfoEntityGroup<br>slbTrapInfoOperStatus<br>slbTrapInfoClusterName<br>slbTrapInfoServerIpAddr | load balancing | A change occurred in the operational status of the server load balancing entity. |
| **slbTrapInfoEntityGroup**—The entity group inside SLB management.<br>**slbTrapInfoOperStatus**—The operational status of an SLB cluster or server.<br>**slbTrapInfoClusterName**—A change occurred in the operational status of an SLB entity.<br>**slbTrapInfoServerIpAddr**—The IP address of a server.<br>**Note**: This trap is not supported. | | | | |
| 29 | sessionAuthenticationTrap | sessionAccessType<br>sessionUserName<br>sessionUserIpAddress<br>sessionAuthFailure | session | An authentication failure trap is sent each time a user authentication is refused. |
| **sessionAccessType**—The access type of the session.<br>**sessionUserName**—The user name of the user logged-in.<br>**sessionUserIpAddress**—The IP address of the user logged-in. | | | | |
| 30 | trapAbsorptionTrap | trapAbsorStamp<br>trapAbsorTrapId<br>trapAbsorCounter<br>trapAbsorTime | none | The absorption trap is sent when a trap has been absorbed at least once. |
| **trapAbsorStamp**—The time stamp of the absorbed trap.<br>**trapAbsorTrapId**—The trap identifier of the absorbed trap.<br>**trapAbsorCounter**—The number of the iterations of the absorbed trap.<br>**trapAbsorTime**—The time stamp of the last iteration. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 31 | alaDoSTrap | alaDoSType<br>alaDoSDetected | ip | Indicates that the sending agent has received a Denial of Service (DoS) attack. |

**alaDoSType**—Index field for the alaDoSTable. Integer indicating the DoS Type: 0=portscan, 1=tcpsyn, 2=pingofdeath, 3=smurf, 3=pepsi, 5=land, 6=teardropBonkBoink,loopbacksrcip(7), invalidip(8), mcastmismatch(9), ucastipmcastmac(10), pingattack(11), arpattack(12), arppoison(13), antispoof(14).
**alaDoSDetected**—Number of attacks detected

**pethMainPseConsumptionPower**—Measured usage power expressed in Watts.

| | | | | |
|---|---|---|---|---|
| 32 | ospfNbrStateChange | ospfRouterId<br>ospfNbrIpAddr<br>ospfNbrAddressLessIndex<br>ospfNbrRtrId<br>ospfNbrState | ospf | Indicates a state change of the neighbor relationship. |

**ospfRouterId**—A 32-bit integer uniquely identifying the router in the Autonomous System. By convention, to ensure uniqueness, this should default to the value of one of the router's IP interface addresses.
**ospfNbrIpAddr**—The IP address this neighbor is using in its IP Source Address. Note that, on address-less links, this will not be 0.0.0.0, but the address of another of the neighbor's interfaces.
**ospfNbrAddressLessIndex**—On an interface having an IP Address, zero. On address-less interfaces, the corresponding value of ifIndex in the Internet Standard MIB. On row creation, this can be derived from the instance.
**ospfNbrRtrId**—A 32-bit integer (represented as a type IpAddress) uniquely identifying the neighboring router in the Autonomous System.
**ospfNbrState**—The State of the relationship with this Neighbor.

| | | | | |
|---|---|---|---|---|
| 33 | ospfVirtNbrStateChange | ospfRouterId<br>ospfVirtNbrArea<br>ospfVirtNbrRtrId<br>ospfVirtNbrState | ospf | Indicates a state change of the virtual neighbor relationship. |

**ospfRouterId**—A 32-bit integer uniquely identifying the router in the Autonomous System. By convention, to ensure uniqueness, this should default to the value of one of the router's IP interface addresses.
**ospfVirtNbrArea**—The Transit Area Identifier.
**ospfVirtNbrRtrId**—A 32-bit integer uniquely identifying the neighboring router in the Autonomous System.
**ospfVirtNbrState**—The state of the Virtual Neighbor Relationship.

| | | | | |
|---|---|---|---|---|
| 34 | lnkaggAggUp | traplnkaggId<br>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. |

**traplnkaggId—**Index value of the Link Aggregate group.
**traplnkaggIfIndex**—Port of the Link Aggregate group.

| | | | | |
|---|---|---|---|---|
| 35 | lnkaggAggDown | traplnkaggId<br>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. |

**traplnkaggId—**Index value of the Link Aggregate group.
**traplnkaggIfIndex**—Port of the Link Aggregate group.

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 36 | lnkaggPortJoin | traplnkaggId<br>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.<br>**traplnkaggIfIndex**—Port of the Link Aggregate group. | | | | |
| 37 | lnkaggPortLeave | traplnkaggId<br>traplnkaggPortIfIndex | linkaggregation | This trap is sent when any given port detaches from the link aggregate group. |
| **traplnkaggId—**Index value of the Link Aggregate group.<br>**traplnkaggIfIndex**—Port of the Link Aggregate group. | | | | |
| 38 | lnkaggPortRemove | traplnkaggId<br>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.<br>**traplnkaggIfIndex**—Port of the Link Aggregate group. | | | | |
| 39 | monitorFileWritten | mirmonPrimarySlot<br>mirmonPrimaryPort<br>monitorFileName<br>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.<br>**mirmonPrimaryPort**—Port of mirrored or monitored interface.<br>**monitorFileName**—The name of the file in which the traffic will be stored (the default is "PMONITOR.ENC").<br>**monitorFileSize**—The number of bytes in 16K (16384) increments allowed for the file (default 16384 bytes). The file contains only the last **monitorFileName** bytes of the current port monitoring instance. | | | | |
| 40 | alaVrrp3TrapProtoError | alaVrrp3TrapProtoErrReason | vrrp | The error trap indicates that the sending agent has encountered the protocol error. |
| **alaVrrp3TrapProtoErrReason**—This indicates the reason for protocol error trap. | | | | |
| 41 | alaVrrp3TrapNewMaster | alaVrrp3OperMasterlpAddrType<br>alaVrrp3OperMasterlpAddr<br>alaVrrp3TrapNewMasterReason | vrrp | The newMaster trap indicates that the sending agent has transitioned to Master state. |
| **alaVrrp3OperMasterlpAddrType**—This specifies the type of alaVrrp3OperMasterlpAddr in this row.<br>**alaVrrp3OperMasterlpAddr**—The master switch's real (primary for vrrp over IPv4) IP address. This is the Ip address listed as the source in the advertisement last received by this virtual switch. For IPv6, a link local address.<br>**alaVrrp3TrapNewMasterReason**—This indicates the reason for NewMaster trap. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 42 | chassisTrapsPossibleDuplicateMac | physicalIndex<br>baseMacAddress | chassis | This trap is sent when there is a possiblity of duplicate a MAC address in the network. |
| **physicalIndex**—The Physical index of the involved object.<br>**baseMacAddress**—The base MAC Address. | | | | |
| 43 | lldpRemTablesChange | lldptatsRemTablesInserts<br>lldptatsRemTablesDeletes<br>lldptatsRemTablesDrops<br>lldptatsRemTablesAgeouts | aip | This trap is sent when the value ofthe LLDP Stats Rem Table Last ChangeTime changes. It can be utilized by an NMS to trigger LLDP remote systems table maintenance polls. |
| **lldptatsRemTablesInserts**—The number of times the complete set of information advertised by a particular MSAP has been inserted into tables contained in lldpRemoteSystemsData and lldpExtensions objects.<br>**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<br>**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<br>**lldptatsRemTablesAgeouts**—The number of times the complete set of information advertised by a particular MSAP has been deleted from tables contained in lldpRemoteSystemsData and lldpExtensions objects because the information timeliness interval has expired. | | | | |
| 44 | pimNeighborLoss | pimNeighborUpTime | ipmr | This trap is sent when an adjacency with a neighbor is lost.<br><br>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.<br><br>The notification is generated whenever the PIM NeighborLoss Count is incremented, subject to the rate limit specified by the PIM Neighbor Loss NotificationPeriod. |
| **pimNeighborUpTime**—The time since this PIM neighbor (last) became a neighbor of the local router. | | | | |
| 45 | pimInvalidRegister | PimGroupMappingPimMode<br>pimInvalidRegisterAddressType<br>pimInvalidRegisterOrigin<br>pimInvalidRegisterGroup<br>pimInvalidRegisterRp | ipmr | This trap is sent when an invalid PIM Register message is received.<br><br>The notification is generated whenever the PIM Invalid Register Message Reveived counter is incremented, subject to the rate limit specified by the Invalid Register NotificationPeriod. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| | | **pimGroupMappingPimMode**—The PIM mode used for groups in this group prefix.<br>**pimInvalidRegisterAddressType**—The address type stored in pimInvalidRegisterOrigin, pimInvalid RegisterGroup and pimInvalidRegisterRp. If no unexpected Register messages are received, the onject is set to "Unknown".<br>**pimInvalidRegisterOrigin**—The source address of the last unexpected Register message received by thisdevice<br>**pimInvalidRegisterGroup**—The IP multicast group address to which the last unexpected Register message received by this device was addressed.<br>**pimInvalidRegisterRp**—The RP address to which the last unexpected Register message received by this device was delivered. | | | |
| 46 | pimInvalidJoinPrune | pimGroupMappingPimMode<br>pimInvalidJoinPruneAddressType<br>pimInvalidJoinPruneOrigin<br>pimInvalidJoinPruneGroup<br>pimInvalidJoinPruneRp<br>pimNeighborUpTime | ipmr | This trap is sent when an invalid PIM Join/Prune message is received.<br><br>The notification is generated whenever the PIM Invalid Join Prune Messages Recieved counter is incremented, subject to the rate limit specified by the PIM Invalid Join/Prune Notification Period. |
| | | **pimGroupMappingPimMode**—The PIM mode used for groups in this group prefix.<br>**pimInvalidRegisterAddressType**—The address type stored in pimInvalidRegisterOrigin, pimInvalid RegisterGroup and pimInvalidRegisterRp. If no unexpected Register messages are received, the onject is set to "Unknown".<br>**pimInvalidJoinPruneOrigin**—The source address of the last unexpected Join/Prune message received<br>**pimInvalidJoinPruneGroup**—The IP multicast group address carried in the last unexpected Join/Prune message received<br>**pimInvalidJoinPruneRp**—The RP address carried in the last unexpected Join/Prune message received<br>**pimNeighborUpTime**—The time since this PIM neighbor (last) became a neighbor of the local router. | | | |
| 47 | PimRPMappingChange | pimGroupMappingPimMode<br>pimGroupMappingPrecedence | ipmr | This trap is sent when a change is detected to the active RP mapping on the device.<br><br>The notification is generated whenever the PIM RP Mapping Change Count is incremented, subject to the rate limit specified by PIM RP Mapping Change Notification Period |
| | | **pimGroupMappingPimMode**—The PIM mode used for groups in this group prefix.<br>**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 | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 48 | PimInterfaceElection | pimInterfaceAddressType<br>pimInterfaceAddress | ipmr | This trap is sent when a new DR or DR has been elected on a network.<br><br>The notification is generated whenever the counter PIM Interface Elections Win Count is incremented, subject to the rate limit specified by PIM Interface Election Notification Period. |
| **pimInterfaceAddressType**—The address type of the PIM interface.<br>**pimInterfaceAddress**—The primary IP address of this router on this PIM interface. | | | | |
| 49 | pimBsrElectedBSRLostElection | pimBsrElectedBSRAddressType, pimBsrElectedBSRAddress, pimBsrElected | ipmr | This trap is sent when the current E-BSR loses an election to a new Candidate-BSR. |
| **pimBsrElectedBSRAddressType**—The address type of the elected BSR.<br>**pimBsrElectedBSRAddress**—The unicast address of the elected BSR.<br>**pimBsrElectedBSRPriority**—The priority value for the elected BSR for this address type. Higher values for this object indicate higher priorities (0 - 255). | | | | |
| 50 | pimBsrCandidateBSRWinElection pim | pimBsrCandidateBSRElectedBSR | ipmr | This trap is sent when a C-BSR wins a BSR Election. |
| **pimBsrCandidateBSR ElectedBSR**—Indicates whether the local router is the elected BSR for this zone. | | | | |
| 51 | lpsViolationTrap | lpsTrapSwitchName<br>lpsTrapSwitchIpAddr<br>lpsTrapSwitchSlice<br>lpsTrapSwitchPort<br>lpsTrapViolatingMac<br>lpsTrapViolationType<br>systemServicesDate<br>systemServicesTime | bridge | A Learned Port Security (LPS) violation has occurred. |
| **lpsTrapSwitchName**—The name of the switch.<br>**lpsTrapSwitchIpAddr**—The IP address of switch.<br>**lpsTrapSwitchSlice**— The physical slice number for the LPS port on which the violation occurred.<br>**lpsTrapSwitchPort**—The physical port number on which the violation occurred.<br>**lpsTrapViolatingMac**—The violating MAC address.<br>**lpsTrapViolationType**—The type of violation that occurred on the LPS port.<br>**systemServicesDate**—This object contains the current System Date in the following format: MM/DD/YYYY.<br>**systemServicesTime**—This object contains the current System Time in the following format: HH:MM:SS. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 52 | lpsPortUpAfterLearningWindowExpiredT | lpsTrapSwitchName<br>lpsTrapSwitchSlice<br>lpsTrapSwitchPort<br>systemServicesDate<br>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.<br><br>This trap is also generated at the time the Learning Window expires, with a slice and port value of 0. |
| **lpsTrapSwitchName**—The name of the switch.<br>**lpsTrapSwitchSlice**—The slot number for the LPS port on which the violation occured<br>**lpsTrapSwitchPort**—The port number for the LPS port on which the violation occured<br>**systemServicesDate**—The current System Date in the following format: MM/DD/YYYY.<br>**systemServicesTime**—The current System Time in the following format: HH:MM:SS. | | | | |
| 53 | lpsLearnTrap | lpsLearnTrapThreshold | bridge | This trap is sent when the number of bridged MACs learned matches the configured Learned Trap Threshhold. A trap is then generated or every additional MAC that is learned. |
| **lpsLearnTrapThreshold**—The number of bridged MAC addresses that can be learned before a trap is sent. | | | | |
| 54 | gvrpVlanLimitReachedEvent | alaGvrpMaxVlanLimit | bridge | This trap is sent when the number of dynamically-learned VLANs has reached the configured limit. |
| **alaGvrpMaxVlanLimit**—The maximum number of dynamic VLANs that can be created on the system by GVRP before a trap is sent. | | | | |
| 55 | alaNetSecPortTrapAnomaly | alaNetSecPortTrapInfoIfId,<br>alaNetSecPortTrapInfoAnomaly,<br>alaNetSecPortTrapInfoType | netsec | This trap is sent when and anomalout port quarantine is detected. |
| **alaNetSecPortTrapInfoIfId**—The interface index of port on which anomaly is detected.<br>**alaNetSecPortTrapInfoAnomaly—**The type of anomaly detected on the interface.<br>**alaNetSecPortTrapInfoType—**The ature of anomaly. Informs if system attached to interface is source or target of the anomaly. | | | | |
| 56 | alaNetSecPortTrapQuarantine | alaNetSecPortTrapInfoIfId | netsec | This trap is sent when and anomalout port quarantine is detected. |
| **alaNetSecPortTrapInfoIfId**—The interface index of port on which anomaly is detected. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 57 | ifMauJabberTrap | ifMauJabberState | 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<br>alaUdldPrevState<br>alaUdldCurrentState | interface | This trap is sent when the UDLD state of a port has changed. |
| **alaUdldPortIfIndex**—The interface index of the port which troggered the UDLD trap.<br>**alaUdldPrevState**—The previous UDLD state of the port - notapplicable (0), shutdown (1), undetermined (2), bidirectional (3).<br>**alaUdldCurrentState**—he current UDLD state of the port - notapplicable (0), shutdown (1), undetermined (2), bidirectional (3). | | | | |
| 59 | ndpMaxLimitReached | none | ipv6 | This trap is sent when the hardware table has reached the maximum number of entries supported. |
| 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. |
| 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<br>alaErpRingState | erp | This trap is sent when the ERP Ring State has changed from "Idle" to "Protection". |
| **alaErpRingId**—The unique Ring identifier.<br>**alaErpRingState**—The current state of the Ring (0=Idle, 1=Protection). | | | | |
| 63 | alaErpRingMultipleRpl | alaErpRingId | erp | This trap is sent when multiple RPLs are detected in the Ring. |
| **alaErpRingId**—The unique Ring identifier. | | | | |
| 64 | alaErpRingRemoved | alaErpRingId | erp | This trap is sent when the Ring is removed dynamically. |
| **alaErpRingId**—The unique Ring identifier. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 65 | ntpMaxAssociation | | ntp | This trap is generated when the the maximum number of peer and client associations configured for the switch is exceeded. |
| **NtpMaxAssociation—**The maximum number of peer and client associations that the switch will serve. | | | | |
| 66 | ddmTemperatureThresholdViolated | ifIndex<br>ddmNotificationType<br>ddmTemperature | | This trap is sent when a transceiver's 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 a transceiver's temperature. |
| **ifIndex—**The interface index.<br>**ddmNotificationType—**The trap type for monitored DDM parameters (clearViolation(1), highAlarm(2), highWarning(3), lowWarning(4), lowAlarm(5).<br>**ddmTemperature—**The temperature, in tenths of a degree celcius. | | | | |
| 67 | ddmVoltageThresholdViolated | ifIndex<br>ddmNotificationType<br>ddmSupplyVoltage | port | This trap is sent when a transceiver's 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 a transceiver's supply voltage. |
| **ifIndex—**The interface index.<br>**ddmNotificationType—**The trap type for monitored DDM parameters (clearViolation(1), highAlarm(2), highWarning(3), lowWarning(4), lowAlarm(5)<br>**ddmSupplyVoltage—**The voltage, in tenths of a volt. | | | | |
| 68 | ddmCurrentThresholdViolated | ifIndex,<br>ddmNotificationType<br>ddmTxBiasCurrent | port | This trap is sent when a transceiver's bias current has crossed any threshold or reverted from previous threshold violation for a port represented by ifIndex. It also provides the current realtime value of a transceiver's bias current. |
| **ifIndex—**The interface index.<br>**ddmNotificationType—**The trap type for monitored DDM parameters (clearViolation(1), highAlarm(2), highWarning(3), lowWarning(4), lowAlarm(5).<br>**ddmTxBiasCurrent—**The current Transmit Bias Current of the SFP/XFP in 10s of milli-Amperes (mA). | | | | |
| 69 | ddmTxPowerThresholdViolated | ifIndex<br>ddmNotificationType<br>ddmTxOutputPower | port | This trap is sent when a transceiver's Tx output power has crossed any threshold or reverted from previous threshold violation for a port represented by ifIndex. It also provides the current realtime value of a transceiver's Tx output power. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| **ifIndex—**The interface index.<br>**ddmNotificationType—**The trap type for monitored DDM parameters (clearViolation(1), highAlarm(2), highWarning(3), lowWarning(4), lowAlarm(5).<br>**ddmTxOutputPower—**The current Output Power of the SFP/XFP in 10s of milli-Watts (mW). | | | | |
| 70 | ddmRxPowerThresholdViolated | ifIndex,<br>ddmNotificationType<br>ddmRxOpticalPower | port | This trap is sent when a transceiver's Rx optical power has crossed any threshold or reverted from previous threshold violation for a port represented by ifIndex. It also provides the current realtime value of a transceiver's Rx optical power |
| **ifIndex—**The interface index.<br>**ddmNotificationType—**The trap type for monitored DDM parameters (clearViolation(1), highAlarm(2), highWarning(3), lowWarning(4), lowAlarm(5).<br>**ddmRxOpticalPower—**The current Received Optical Power of the SFP/XFP in 10s of milli-Watts (mW). | | | | |
| 71 | webMgtServerErrorTrap | webMgtServerError | webmgt | This trap is sent when the Web Management server goes into error state after crashing twice within a minute. |
| **webMgtServerError—**Error code string when WebView Server is in error status. Format is 'Error Num: {Number}. {String message}.' where {Number} is an integer representing the error code and {String message} is the error string message. | | | | |
| 72 | multiChassisIpcVlanUp | multiChassisTrapIpcVlan | multi-chassis | Indicates the operational status for the multi-chassis communication VLAN is Up. |
| **multiChassisTrapIpcVlan—**The multi-chassis IPC VLAN number. | | | | |
| 73 | multiChassisIpcVlanDown | multiChassisTrapIpcVlan | multi-chassis | Indicates the operational status for the multi-chassis communication VLAN is Down. |
| **multiChassisTrapIpcVlan—**The multi-chassis IPC VLAN number. | | | | |
| 74 | multiChassisMisconfigurationFailure | multiChassisTrapFailure | 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 | multiChassisHelloIntervalConsisFailure | multiChassisTrapFailure | multi-chassis | This trap is sent when there is an MCM Hello Interval consistency falure. |
| **multiChassisTrapFailure**—Indicates multi-chassis failure. | | | | |
| 76 | multiChassisStpModeConsisFailure | multiChassisTrapFailure | multi-chassis | This trap is sent when ther is an STP mode consistency falure. |
| **multiChassisTrapFailure**—Indicates multi-chassis failure. | | | | |
| 77 | multiChassisStpPathCostModeConsisFailure | multiChassisTrapFailure | multi-chassis | This trap is sent when ther is an STP path cost mode consistency falure. |
| **multiChassisTrapFailure**—Indicates multi-chassis failure.. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 78 | multiChassisVflinkStatusConsisFailure | multiChassisTrapFailure | multi-chassis | This trap is sent when there is an MCM Virtual Fabric Link status consistency falure |
| **multiChassisTrapFailure**—Indicates multi-chassis failure. | | | | |
| 79 | multiChassisStpBlockingStatus | multiChassisTrapStpBlockingVlanList | 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, seperated by comas. | | | | |
| 80 | multiChassisLoopDetected | multiChassisTrapFailure | multi-chassis | This trap is sent when a loop is detected. |
| **multiChassisTrapFailure**—Indicates multi-chassis failure. | | | | |
| 81 | multiChassisHelloTimeout | multiChassisTrapFailure | multi-chassis | This trap is sent when the Hellow Timer expires. |
| **multiChassisTrapFailure**—Indicates multi-chassis failure. | | | | |
| 82 | multiChassisVflinkDown | multiChassisTrapFailure | multi-chassis | This trap is sent when the Virtual Fabric Link goes down. |
| **multiChassisTrapFailure**—Indicates multi-chassis failure. | | | | |
| 83 | multiChassisVFLMemberJoinFailure | multiChassisTrapVFL, multiChassisTrapVFLMemberPort, multiChassisTrapDiagnistic | multi-chassis | This trap is sent when a port configured as virtual fabric member is unable to join the virtual fabric link |
| **multiChassisTrapVFL**—The multi-chassis Virtual Fabric Link interface.<br>**multiChassisTrap VFLMemberPort**—The multi-chassis VFL member port number.<br>**multiChassisTrapDiagnistic**—The reason a port configured as virtual-fabric member is unable to join the virtual-fabric link - 1. Duplex Mode, 2. Speed. | | | | |
| 84 | alaDHLVlanMoveTrap | alaDHLSessionID, alaDHLPortFrom, alaDHLPortTo, alaDHLVlanMoveReason | vlan | When linkA or linkB goes down or comes up and both ports are are part of some vlan-map, this trap is sent to the Management Entity, with the DHL port information. |
| **alaDHLSessionID—**The DHL Session ID for which alaDHLVlanMoveTrap needs to be sent to the Management Entity.<br>**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.<br>**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<br>**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. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 86 | alaDhcpClientAddressExpiryTrap | ialaDhcpClient Address | 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 | alaDhcpClientA ddress, alaDhcpClientN ewAddress | 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.<br>**alaDhcpClientNewAddress—**The new IP address assigned to the DHCP client. | | | | |
| 88 | vRtrIsisDatabaseOverload | vRtrIsisSystem Leve<br>lisisSysL1 State<br>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.<br>**isisSysL1State**—Level 1 Routing (1)<br>**isisSysL2State**—Level 2 Routing (2) | | | | |
| 89 | vRtrIsisManualAddressDrops | isisManAreaAd drExistState | isis | This trap is sent when one of the manual area addresses assigned to this system is ignored when computing routes. The object vRtrIsisManAreaAddrExistState describes the area that has been dropped.<br><br>This trap is edge triggered, and should not be regenerated until an address that was used in the previous computation has been dropped. |
| **isisManAreaAddrExistState**—The area ID that was ignored when computing routes. | | | | |
| 90 | vRtrIsisCorruptedLSPDetected | vRtrIsisSystem Level<br>vRtrIsisTrapLS PID | isis | This trap is sent when an LSP that was stored in memory has become corrupted.<br><br>The LSP ID is forwarded. The ID may be known, but in some implementations there is a chance that the ID itself will be corrupted. |
| **vRtrIsisSystemLevel**—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm.<br>**vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 91 | vRtrIsisMaxSeqExceedAttempt | vRtrIsisSystemLevel<br>vRtrIsisTrapLSPID | isis | This trap is sent when the sequence number on an LSP wraps the 32 bit sequence counter. |
| **vRtrIsisSystemLevel**—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm.<br>**vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU. | | | | |
| 92 | vRtrIsisIDLenMismatch | vRtrIsisFieldLen<br>vRtrIsisIfIndex<br>vRtrIsisPDUFragment | isis | This trap is sent when when a PDU with a different System ID Length is received. The notification includes the index to identify the circuit for the PDU and the header of the PDU, which may help a network manager identify the source of the problem. |
| **vRtrIsisFieldLen**—The System ID Field length.<br>**vRtrIsisIfIndex**—The ISIS interface on which the PDU was received.<br>**vRtrIsisPDUFragment**—The first 64 bytes of a PDU that triggered the trap. | | | | |
| 93 | vRtrIsisMaxAreaAddrsMismatch | vRtrIsisMaxAreaAddress,<br>vRtrIsisIfIndex<br>vRtrIsisPDUFragment | isis | This trap is sent when a PDU with a different Maximum Area Addresses value is recieved. The notification includes the header of the packet, which may help a network manager identify the source of the problem. |
| **vRtrIsisMaxAreaAddress**—The maximum number of area addresses in the PDU.<br>**vRtrIsisIfIndex**—The ISIS interface on which the PDU was received.<br>**vRtrIsisPDUFragment**—The first 64 bytes of a PDU that triggered the trap. | | | | |
| 94 | vRtrIsisOwnLSPPurge | vRtrIsisIfIndex,<br>vRtrIsisTrapLSPID<br>vRtrIsisSystemLevel | isis | This trap is sent when sent when a PDU is received with the system ID and zero age. This notification includes the circuit Index if available, which may help a network manager identify the source of theproblem. |
| **vRtrIsisIfIndex**—The ISIS interface on which the PDU was received.<br>**vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU.<br>**vRtrIsisSystemLevel**—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm. | | | | |
| 95 | vRtrIsisSequenceNumberSkip | vRtrIsisTrapLSPID<br>vRtrIsisIfIndex<br>vRtrIsisSystemLevel | isis | If an LSP without System ID and different contents is received, the LSP may be reissued with a higher sequence number.<br><br>If two Intermediate Systems are configured with the same System ID, the sequence number is increased and this notification is sent. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| **vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU.<br>**vRtrIsisIfIndex**—The ISIS interface on which the PDU was received.<br>**vRtrIsisSystemLevel**—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm. | | | | |
| 96 | vRtrIsisAutTypeFail | vRtrIsisSystem Level,<br>vRtrIsisPDUFragment,<br>vRtrIsisIfIndex | isis | This trap is sent when a PDU with the wrong authentication type is received. The notification includes the header of the packet, which may help a network manager identify the source of the problem. |
| **vRtrIsisSystemLevel**—Identifies the level to which the notification applies.Routing within an area is referred. to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm.<br>**vRtrIsisPDUFragment**—Contains up to the first 64 bytes of a PDU that triggered the trap.<br>**vRtrIsisIfIndex**—The ISIS interface on which the PDU was received. | | | | |
| 97 | vRtrIsisAuthFail | vRtrIsisSystem Level,<br>vRtrIsisPDUFragment,<br>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.<br>**vRtrIsisPDUFragment**—Contains up to the first 64 bytes of a PDU that triggered the trap.<br>**vRtrIsisIfIndex**—The ISIS interface on which the PDU was received.. | | | | |
| 98 | vRtrIsisVersionSkew | vRtrIsisProtocol Version<br>vRtrIsisSystem Level<br>vRtrIsisPDUFragment<br>vRtrIsisIfIndex | isis | This trap is sent when a Hello PDU is received from an IS running a different version of the protocol.<br><br>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.<br>**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.<br>**vRtrIsisPDUFragment**—Contains up to the first 64 bytes of a PDU that triggered the trap.<br>**vRtrIsisIfIndex**—The ISIS interface on which the PDU was received. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 99 | vRtrIsisAreaMismatch | vRtrIsisLSPSize<br>vRtrIsisSystemLevel<br>vRtrIsisIfIndex<br>vRtrIsisPDUFragment | isis | This trap is sent when a Hello PDU from an IS that does not share any area address is received.<br><br>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.<br>**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.<br>**vRtrIsisIfIndex**—The ISIS interface on which the PDU was received.<br>**vRtrIsisPDUFragment**—Contains up to the first 64 bytes of a PDU that triggered the trap. | | | | |
| 100 | vRtrIsisRejectedAdjacency | vRtrIsisSystemLevel<br>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.<br>**vRtrIsisIfIndex**—The ISIS interface on which the PDU was received. | | | | |
| 101 | vRtrIsisLSPTooLargeToPropagate | vRtrIsisLSPSize<br>vRtrIsisSystemLevel<br>vRtrIsisTrapLSPID<br>vRtrIsisIfIndex | isis | This trap is sent when an LSP is larger than the Data Link Block Size for a circuit. |
| **vRtrIsisLSPSize**—The size of the LSP received.<br>**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.<br>**vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU.<br>**vRtrIsisIfIndex**—The ISIS interface on which the LSP was received. | | | | |
| 102 | vRtrIsisOrigLSPBufSizeMismatch | vRtrIsisOriginatingBufferSize<br>vRtrIsisSystemLevel<br>vRtrIsisTrapLSPID<br>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. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| | **vRtrIsisOriginatingBufferSize**—The buffer size advertised by the peer.<br>**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.<br>**vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU.<br>**vRtrIsisIfIndex**—The ISIS interface on which the LSP was received. | | | |
| 103 | vRtrIsisProtoSuppMismatch | vRtrIsisProtocolsSupported<br>vRtrIsisSystemLevel<br>vRtrIsisTrapLSPID<br>vRtrIsisIfIndex | isis | This trap is sent when a non-pseudonode segment 0 LSP is received that has no matching protocols supported.<br><br>This may be because the system does not generate the field, or because there are no common elements.<br><br>The list of protocols supported should be included in the notification: it may be empty if the TLV is not supported, or if the TLV is empty. |
| | **vRtrIsisProtocolsSupported**—The protocols supported by an adjacent system. This may be empty<br>**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.<br>**vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU.<br>**vRtrIsisIfIndex**—The ISIS interface on which the LSP was received. | | | |
| 104 | vRtrIsisAdjacencyChange | vRtrIsisSystemLevel<br>vRtrIsisIfIndex<br>vRtrIsisTrapLSPID<br>isisISAdjState | isis | This trap is sent when adjacency changes state, entering or leaving state up.<br><br>The first 6 bytes of the vRtrIsisTrapLSPID are the SystemID of the adjacent IS. The isisISAdjState is the new state of the adjacency. |
| | **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.<br>**vRtrIsisIfIndex**—The ISIS interface on which the trap was received.<br>**vRtrIsisTrapLSPID**—An Octet String that uniquely identifies a Link State PDU.<br>**isisISAdjState**—The state of the adjacent router. | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 105 | vRtrIsisCircIdExhausted | vRtrIsisIfIndex | isis | This trap is sent when sent when ISIS cannot be started on a LAN interface because a unique circid could not be assigned due to the exhaustion of the Circuit ID space. This can only happen on broadcast interfaces.<br><br>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. |
| **vRtrIsisIfIndex**—The ISIS interface. | | | | |
| 106 | vRtrIsisAdjRestartStatusChange | vRtrIsisSystemLevel<br>vRtrIsisIfIndex<br>vRtrIsisISAdjRestartStatus | isis | This trap is sent when an adjancency's graceful restart status changes. |
| **vRtrIsisSystemLevel**—Identifies the level to which the notification applies.Routing within an area is referred to as Level-1 routing. Routing between two or more areas is referred to as Level 2 routing. Each area runs a separate copy of the basic link-state routing algorithm.<br>**vRtrIsisIfIndex**—The ISIS interface.<br>**vRtrIsisISAdjRestartStatus**—The new graceful restart state of the adjacency. | | | | |
| 107 | alaMvrpVlanLimitReachedEvent | alaMvrpMaxVlanLimit | bridge | This trap is sent when the number of VLANs learned dynamically by MVRP reaches the configured limit. |
| **alaMvrpMaxVlanLimit—**The the maximum number of dynamic VLANs that can be created on the system by MVRP. If the number of VLANs created by MVRP reaches this limit, the system will prevent MVRP from creating additional VLANs (32 - 4094, Default = 256). | | | | |
| 108 | alaHAVlanClusterPeerMismatch | alaHAVlanClusterId | ha-vlan | This trap is sent when parameteras configured for this cluster ID (Level 1 check) does not match accross the MCLAG peers. |
| **alaHAVlanClusterId—**The Cluster ID Number. | | | | |
| 109 | alaHAVlanMCPeerMismatch | alaHAVlanClusterId<br>alaHAVlanMultiChassisId<br>alaHAVlanClusterPortIfIndex | ha-vlan | This trap is sent when the cluster parameters are matching on the peers, but MCLAG is not configured or clusters are not in operational state. |
| **alaHAVlanClusterId—**The Cluster ID Number.<br>**alaHAVlanMultiChassisId—**The Multi Chassis ID identifying the Multi Chassis Peer.<br>**alaHAVlanClusterPortIfIndex—**The ifindex identifying the cluster port. An ifindex of 1 shall be used for all ports. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 110 | alaHAVlanDynamicMAC | alaHAVlanClusterId<br>alaHAVlanClusterInetAddress<br>alaHAVlanClusterMacAddress<br>alaHAVlanClusterPortIfIndex | ha-vlan | The trap is sent when the dynamic MAC is learned on non-server cluster port |
| **alaHAVlanClusterId—**The Cluster ID Number.<br>**alaHAVlanClusterInetAddress—**The type of IP address associated with the L3 cluster (e.g., ipv4).<br>**alaHAVlanClusterMacAddress—**The type of ARP resolution used in L3 cluster (static, dynamic, invalid).<br>**alaHAVlanClusterPortIfIndex—**The ifindex identifying the cluster port. An ifindex of 1 shall be used for all ports. | | | | |
| 111 | unpMcLagMacIgnored | alaDaUnpMacAddr<br>alaDaUnpSourceIpAddr<br>alaDaUnpNativeVlan<br>alaDaUnpVlan<br>alaDaUnpMCLAGId | da-unp | This trap is sent when a MAC/ User is dropped because the VLAN does not exist or UNP is not enabled on the MCLAG.. |
| **alaDaUnpMacAddr—**The MAC that failed to get configured on peer chassis.<br>**alaDaUnpSourceIpAddr—**The IP address of the MAC that failed to get configured on peer chassis.<br>**alaDaUnpNativeVlan—**The native VLAN of MCLAG on which the MAC ingressed.<br>**alaDaUnpVlan—**The VLAN on which the MAC was classified on the local chassis.<br>**alaDaUnpMCLAGId—**The Link Agg Id for MCLAG.. | | | | |
| 112 | unpMcLagConfigInconsistency | alaDaUnpCommandType<br>alaDaUnpName<br>alaDaUnpMacAddr1<br>alaDaUnpMacAddr2<br>alaDaUnpIpAddr<br>alaDaUnpIpMask<br>alaDaUnpVlanTag<br>alaDaUnpMCLAGId | da-unp | This trap is sent when a configuration becomes “Out of Sync". |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| **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).<br>**alaDaUnpName—**Indicates which UNP Profile is out-of-sync. If there is no UNP Profile associated, a zero length string is sent.<br>**alaDaUnpMacAddr1—**The MAC for MAC rule or the lower limit of MAC Range Rule.<br>**alaDaUnpMacAddr2—**The upper limit of MAC Range Rule.<br>**alaDaUnpIpAddr—**The IP address in the IP Rule.<br>**alaDaUnpIpMask—**The IP Mask of the IP address in the IP Rule.<br>**alaDaUnpVlanTag—**The VLAN VLAN Tag Rule. A zero value means it is not applicable.<br>**alaDaUnpMCLAGId—**The Link Agg ID for MCLAG.. | | | | |
| 113 | multiChassisGroupConsisFailure | multiChassisTrapFailure | mcm | This trap is sent when there is an inconsistency between local and peer chassis group. |
| **multiChassisTrapFailure—**Indicate multi-chassis failure. | | | | |
| 114 | multiChassisTypeConsisFailure | | mcm | This trap is sent when there is an inconsistency between local and peer chassis type. |
| - | | | | |
| 115 | alaPimNonBidirHello | pimNeighborAdressType, pimNeighboAddress | pim | This trap is sent when a bidir-capable router has received a PIM hello from a non-bidir-capable router. It is generated whenever the counter alaPimsmNon-BidirHelloMsgsRcvd is incremented, subject to the rate limit specified by alaPimsmNonBidirHelloNotificationPeriod. |
| **pimNeighborAdressType—**The address type of the PIM neighbor.<br>**pimNeighborAddress—**The primary IP address of the PIM neighbor. The InetAddressType is given by the pimNeighborAddressType object. | | | | |
| 116 | dot1agCfmFaultAlarm | dot1agCfmMepHighestPrDefect | 802.1AG | This trap is sent when a MEP has a persistent defect condition. A notification (fault alarm) is sent to the management entity with the OID of the MEP that has detected the fault. |
| **dot1agCfmMepHighestPrDefect—**The highest priority defect that has been present since the MEPs Fault Notification Generator State Machine was last in the FNG_RESET state. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 117 | alaSaaIPIterationCompleteTrap | alaSaaCtrlOwnerIndex<br>alaSaaCtrlTestIndex<br>alaSaaIpResultsTestRunIndex<br>alaSaaCtrlLastRunResult<br>alaSaaCtrlLastRunTime | saa | This trap is sent when an IP SAA iteration is completed. |
| | **alaSaaCtrlOwnerIndex—**The Owner name to identify entries in the table. This is currently not supported and its value will always be the string 'USER'.<br>**alaSaaCtrlTestIndex—**A Unique name to identify the entries in the table. The name is unique across various SNMP users.<br>**alaSaaIpResultsTestRunIndex—**The row entry that reports results for a single OAM test run.The value of this object starts at 1 and can go upto a maximum of alaSaaCtrlMaxHistoryRows.<br>**alaSaaCtrlLastRunResult—**The result of the latest SAA test iteration (Undetermined/Success/Failed/ Aborted).<br>**alaSaaCtrlLastRunTime—**The time the last iteration of the SAA was run. | | | |
| 118 | alaSaaEthIterationCompleteTrap | alaSaaCtrlOwnerIndex<br>alaSaaCtrlTestIndex<br>alaSaaEthoamResultsTestRunInde,<br>alaSaaCtrlLastRunResult<br>alaSaaCtrlLastRunTime | saa | This trap is sent when when an eth-LB or Eth-DMM SAA iteration is completed. |
| | **alaSaaCtrlOwnerIndex—**The Owner name to identify entries in the table. This is currently not supported and its value will always be the string 'USER'.<br>**alaSaaCtrlTestIndex—**A Unique name to identify the entries in the table. The name is unique across various SNMP users.<br>**alaSaaEthoamResultsTestRunIndex—**The row entry that reports results for a single Eth-LB/DMM test run. The value of this object starts from 1 and can go upto a maximum of alaSaaCtrlMaxHistoryRows.<br>**alaSaaCtrlLastRunResult—**The result of the latest SAA test iteration (Undetermined/Success/Failed/ Aborted).<br>**alaSaaCtrlLastRunTime—**The time the last iteration of the SAA was run. | | | |
| 119 | alaSaaMacIterationCompleteTrap | | saa | This trap is sent when a MAC iteration is complete. |
| | **ala—**The ? | | | |
| 120 | virtualChassisStatusChange | virtualChassisOperChasId,<br>virtualChassisStatus | virtual chassis | This trap is sent when a virtual chassis status change is detected. |
| | **virtualChassisOperChasId—**The operational Virtual Chassis ID.<br>**virtualChassisStatus—**The Virtual Chassis status. | | | |
| 121 | virtualChassisRoleChange | virtualChassisOperChasId,<br>virtualChassisRole | virtual chassis | This trap is sent when a virtual chassis role change is detected. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| **virtualChassisOperChasId—**The operational Virtual Chassis ID.<br>**virtualChassisRole—**The Virtual Chassis role:<br>**unassigned(0):** Initial chassis role and election not complete.<br>**master(1):** Chassis is in master role after election.<br>**slave(2):** Chasis is in slave role after election.<br>**inconsistent(3): C**hassis is not consistent after election. | | | | |
| 122 | virtualChassisVflStatusChange | virtualChassisOperChasId,<br>virtualChassisVflIfIndex,<br>virtualChassisVflOperStatus | virtual chassis | This trap is sent when a VFL link status change is detected. |
| **virtualChassisOperChasId—**The operational Virtual Chassis ID.<br>**virtualChassisVflIfIndex—**The Virtual Fabric Link ID.<br>**virtualChassisVflOperStatus—**The Virtual Fabric Link Operational Status (Up/Down/Disabled). | | | | |
| 123 | virtualChassisVflMemberPortStatusCh | virtualChassisOperChasId,<br>virtualChassisVflIfIndex,<br>virtualChassisVflMemberPortIfindex,<br>virtualChassisVflMemberPortOperStatus | virtual chassis | This trap is sent when a VFL link member port has a change of status. |
| **virtualChassisOperChasId—**The operational Virtual Chassis ID.<br>**virtualChassisVflIfIndex—**The Virtual Fabric Link ID<br>**virtualChassisVflMemberPortIfindex—**The Virtual Fabric Link Member Port ifIndex.<br>**virtualChassisVflMemberPortOperStatus—**Virtual Fabric Link Member Port operational status (Up, Down, Disabled). | | | | |
| 124 | virtualChassisVflMemberPortJoinFail | virtualChassisOperChasId,<br>virtualChassisVflIfIndex,<br>virtualChassisVflMemberPortIfindex,<br>virtualChassisDiagnostic | virtual chassis | This trap is sent when a port configured as virtual-fabric member is unable to join the virtual-fabric link. |
| **virtualChassisOperChasId—**The operational Virtual Chassis ID.<br>**virtualChassisVflIfIndex—**The Virtual Fabric Link ID<br>**virtualChassisVflMemberPortIfindex—**The Virtual Fabric Link Member Port ifIndex.<br>**virtualChassisDiagnostic—**Indicates why a port configured as virtual-fabric member is unable to join the virtual-fabric link (Duplex Mode, Speed). | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 125 | lldpRemTablesChange | lldpStatsRemTablesInserts, lldpStatsRemTablesDeletes, lldpStatsRemTablesDrops, lldpStatsRemTablesAgeouts | lldp | This trap is sent when the value of lldpStatsRemTablelastChangeTime changes. It can be utilized by an NMS to trigger LLDP remote systems table maintenance polls. |
| | **lldpStatsRemTablesInserts—**The number of times the complete set of information advertised by a particular MSAP has been inserted into tables contained in lldpRemoteSystemsData and lldpExtensions objects.<br>**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.<br>**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.<br>**lldpStatsRemTablesAgeouts—**The number of times the complete set of information advertised by a particular MSAP has been deleted from tables contained in lldpRemoteSystemsData and lldpExtensions objects because the information timeliness interval has expired. | | | |
| 126 | vRtrLdpInstanceStateChange | vRtrLdpGenAdminState, vRtrLdpGenOperState, vRtrLdpInstanceNotifyReasonCode | ldp | This trap is sent when the LDP module changes state either administratively or operationally. |
| | **vRtrLdpGenAdminState—**The desired administrative state for this LDP instance.<br>**vRtrLdpGenOperState—**The current operational state of this LDP instance.<br>**vRtrLdpInstanceNotifyReasonCode—**The reason for the LDP instance state change (Admin Up, Admin Down, Oper Up, Oper Down) | | | |
| 127 | evbFailedCdcpTlvTrap | evbPortId | evb | This trap is sent when bridge receives a CDCP packet with:<br>- Wrong TLV type, or<br>- Wrong OUI, or<br>- Role is set to Bridge, or<br>- Wrong default channel(scid), or<br>- Incorrect channel number(scid). |
| | **evbPortId—**The IfIndex that uniquely identifies this port. | | | |
| 128 | evbFailedEvbTlvTrap | evbPortId, ieee8021BridgeEvbVSIVlanId | evb | This trap is sent when bridge receives an EVBTLV packet with:<br>- Wrong TLV type. or<br>- Incorrect TLV length, or<br>- Wrong OUI. |
| | **evbPortId—**The IfIndex that uniquely identifies this port**.**<br>**ieee8021BridgeEvbVSIVlanId—**The bridge EVB VSI VLAN. | | | |
| 129 | evbUnknownVsiManagerTrap | evbPortId, ieee8021BridgeEvbSbpPortNumber | evb | This trap is sent when bridge receives a VDP packet with:<br>- Unknown Manager ID type, or<br>- Wrong Manager ID length. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| | **evbPortId—**The IfIndex that uniquely identifies this port**.**<br>**ieee8021BridgeEvbSbpPortNumber—**The bridge EVN SBP Port. | | | |
| 130 | evbVdpAssocTlvTrap | evbPortId,<br>ieee8021BridgeEvbSbpPortNumber,<br>ieee8021BridgeEvbVSIID,<br>ieee8021BridgeEvbVSIIDType,<br>ieee8021BridgeEvbVSITypeVersion | evb | This trap is sent when bridge receives an ASSOC TLV in a VDP packet with:<br>- Null VID found and number of entry field is not 1, or<br>- Unknown filter format,<br>- Null VID on De-Assoc TLV type, or<br>- VSI included more than Max number of filter info entries |
| | **evbPortId—**The IfIndex that uniquely identifies this port**.**<br>**ieee8021BridgeEvbSbpPortNumber—**The EVB port number.<br>**ieee8021BridgeEvbVSIID—**The VSIID that uniquely identifies the VSI in the DCN.<br>**ieee8021BridgeEvbVSIIDType—**The VSIID Type for the VSIID in the DCN:<br>• vsiidIpv4 (1)<br>• vsiidIpv6 (2)<br>• vsiidMAC (3)<br>• vsiidLocal (4)<br>• vsiidUUID (5)<br>**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.<br>**ieee8021BridgeEvbSbpPortNumber—**The EVB SPB port. | | | |
| 131 | evbCdcpLldpExpiredTrap | none | 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 | none | 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 | none | 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. |

**alaSvcId—**The Service identifier.
**alaSvcType—**The service type (e.g., vpls, spb).
**alaSvcIsid—**The I-Domain Service Indentifier (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.
**alaSvcBVlan—**The Backbone VLAN ID (BVLAN), which defines the B-Domain for the PBB traffic.
**alaSvcMulticastMode—** The multicast replication mode for each service:

- 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. |

**alaSvcId—**The Service identifier.
**alaSvcType—**The service type (e.g., vpls, spb).
**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 IDfor this service instance.
**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.

| | | | | |
|---|---|---|---|---|
| 136 | smgrSapError | alaSvcId, alaSapPortId, alaSapEncapVa lue | service manager | This trap is sent when there is a failure to create/delete a Service Access Point. |

**alaSvcId—**The Service identifier.
**alaSapPortId—**The ID of the access port where this SAP is defined.
**alaSapEncapValue—**The value of the label used to identify this SAP on the access port specified by the SAP Port ID.

| | | | | |
|---|---|---|---|---|
| 137 | smgrSapHwError | alaSvcId, alaSapPortId, alaSapEncapVa lue, alaSvcVFI, alaSapVirtualPo rt | 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 |
|---|---|---|---|---|
| | **alaSvcId—**The Service identifier.<br>**alaSapPortId—**The ID of the access port where this SAP is defined.<br>**alaSapEncapValue—**The value of the label used to identify this SAP on the access port specified by SAP Port ID.<br>**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 IDfor this service instance.<br>**alaSapVirtualPort—**The logical representation of a SAP associated with a service instance where customer packets ingress and egress. | | | |
| 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. |
| | **alaSdpId—**The Service identifier.<br>**alaSdpNetworkPort—**The network port where ISIS discovered the neighbor node information (B-VLAN and BMAC).<br>**alaSdpBVlan—**The Backbone VLAN (B-VLAN) where ISIS discovered the neighbor node information (BVLAN and B-MAC).<br>**alaSdpSystemId —**The Backbone MAC (B-MAC) where ISIS discovered the neighbor node information (B-VLAN and B-MAC). | | | |
| 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. |
| | **alaSdpId—**The Service identifier.<br>**alaSdpNetworkPort—**The network port where ISIS discovered the neighbor node information (B-VLAN and BMAC).<br>**alaSdpBVlan—**The Backbone VLAN (B-VLAN) where ISIS discovered the neighbor node information (BVLAN and B-MAC).<br>**alaSdpSystemId —**The Backbone MAC (B-MAC) where ISIS discovered the neighbor node information (B-VLAN and B-MAC). | | | |
| 140 | smgrSdpBindError | alaSvcId, alaSdpBindId, alaSdpBindNet workPort, alaSdpBindBVl an, alaSdpBindSyst emId | service manager | This trap is sent when there is a failure to create/delete an SDP Bind. |
| | **alaSvcId—**The Service identifier.<br>**alaSdpBindId—**The SDP Binding identifier.<br>**alaSdpBindNetworkPort—**The network port associated with a service instance where MPLS-labeled or B-Domain packets ingress and egress.<br>**alaSdpBindBVlan—**The Backbone VLAN ID (B-VLAN) associated with the SDP Bind object.<br>**alaSdpBindSystemId—**The neighbor Backbone MAC (B-MAC) associated with the SDP Bind object. | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 141 | smgrSdpBindHwError | alaSvcId, alaSdpBindId, alaSdpBindNetworkPort, alaSdpBindBVlan, 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. |
| **alaSvcId—**The Service identifier.<br>**alaSdpBindId—**The SDP Binding identifier.<br>**alaSdpBindNetworkPort—**The network port associated with a service instance where MPLS-labeled or B-Domain packets ingress and egress.<br>**alaSdpBindBVlan—**The Backbone VLAN ID (B-VLAN) associated with the SDP Bind object.<br>**alaSdpBindSystemId—**The neighbor Backbone MAC (B-MAC) associated with the SDP Bind object.<br>**alaSdpBindVirtualPort—**The logical representation of a network port associated with a service instance where MPLS-labeled or B-Domain packets ingress and egress. | | | | |
| 142 | smgrGeneralError | alaSvcId, alaSvcType | service manager | This trap is sent when there is a .general system failure detected during normal system operation |
| **alaSvcId—**The Service identifier.<br>**alaSvcType—**The service type (e.g. vpls, spb). | | | | |
| 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. |
| **alaSvcId—**The Service identifier.<br>**alaSvcType—**The service type (e.g., vpls, spb).<br>**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.<br>**alaSvcNumSaps—**The number of SAPs defined on this service.<br>**alaSvcNumSdps—**The number of SDPs bound to this service. | | | | |
| 144 | portViolationNotificationTrap | ifIndex | port | This trap is sent when a port violation is cleared. |
| **ifIndex—**A unique value, greater than zero, for the interface. | | | | |
| 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. |
| **multiChassisConsisFailureRecovered—**Indicates that the system has recovered from a multi-chassis failure. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 146 | alaSaaPacketLossTrap | alaSaaCtrlOwnerIndex, alaSaaCtrlTestIndex, alaSaaCtrlLastRunResult, alaSaaCtrlLastRunTime, alaSaaMacResultsPktsSent, alaSaaMacResultsPktsRcvd | saa | This trap is sent when a a packet is lost during a test. |
| | **alaSaaCtrlOwnerIndex—**The Owner name to identify the responsibility of the entries in the table (Default = User).<br>**alaSaaCtrlTestIndex—**Unique name to identify the entries in the table. The name is unique across various SNMP users (up to 32 characters).<br>**alaSaaCtrlLastRunResult—**The result of the latest SAA test iteration (Undertermined (0), Success (1), Failed (2), Aborted (3)).<br>**alaSaaCtrlLastRunTime—**The time the last iteration of the SAA was run.<br>**alaSaaMacResultsPktsSent—**The number of packets sent during a single MAC-Ping iteration.<br>**alaSaaMacResultsPktsRcvd—**The number of packets received during a single MAC-Ping iteration. | | | |
| 147 | alaSaaJitterThresholdYellowTrap | alaSaaCtrlOwnerIndex, alaSaaCtrlTestIndex, alaSaaCtrlLastRunResult, alaSaaCtrlLastRunTime, alaSaaCtrlJitterThreshold, alaSaaMacResultsAvgJitter | saa | This trap is sent when the Jitter Threshold crosses 90%. |
| | **alaSaaCtrlOwnerIndex—**The Owner name to identify the responsibility of the entries in the table (Default = User).<br>**alaSaaCtrlTestIndex—**Unique name to identify the entries in the table. The name is unique across various SNMP users (up to 32 characters).<br>**alaSaaCtrlLastRunResult—**The result of the latest SAA test iteration (Undertermined (0), Success (1), Failed (2), Aborted (3)).<br>**alaSaaCtrlLastRunTime—**The time the last iteration of the SAA was run.<br>**alaSaaCtrlJitterThreshold—**The configured Jitter Threshold (Range = 0 - 1,000,000, Default = 0)<br>**alaSaaMacResultsAvgJitter—**The average jitter value. | | | |
| 148 | alaSaaRTTThresholdYellowTrap | alaSaaCtrlOwnerIndex, alaSaaCtrlTestIndex, alaSaaCtrlLastRunResult, alaSaaCtrlLastRunTime, alaSaaCtrlRTTThreshold, alaSaaMacResultsAvgRTT | saa | This trap is sent when the RTT Threshold crosses 90%. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| | | **alaSaaCtrlOwnerIndex—**The Owner name to identify the responsibility of the entries in the table (Default = User).<br>**alaSaaCtrlTestIndex—**Unique name to identify the entries in the table. The name is unique across various SNMP users (up to 32 characters).<br>**alaSaaCtrlLastRunResult—**The result of the latest SAA test iteration (Undertermined (0), Success (1), Failed (2), Aborted (3)).<br>**alaSaaCtrlLastRunTime—**The time the last iteration of the SAA was run.<br>**alaSaaCtrlRTTThreshold—**The configured RTT Threshold, micro-seconds (Range = 0 - 1,000,000, Default = 0)<br>**alaSaaMacResultsAvgRTT—**The average Round Trip Time. | | |
| 149 | alaSaaJitterThresholdRedTrap | alaSaaCtrlOwnerIndex,<br>alaSaaCtrlTestIndex,<br>alaSaaCtrlLastRunResult,<br>alaSaaCtrlLastRunTime,<br>alaSaaCtrlJitterThreshold,<br>alaSaaMacResultsAvgJitter | saa | This trap is sent when the Jitter threshold is crossed. |
| | | **alaSaaCtrlOwnerIndex—**The Owner name to identify the responsibility of the entries in the table (Default = User).<br>**alaSaaCtrlTestIndex—**Unique name to identify the entries in the table. The name is unique across various SNMP users (up to 32 characters).<br>**alaSaaCtrlLastRunResult—**The result of the latest SAA test iteration (Undertermined (0), Success (1), Failed (2), Aborted (3)).<br>**alaSaaCtrlLastRunTime—**The time the last iteration of the SAA was run.<br>**alaSaaCtrlJitterThreshold—**The configured Jitter Threshold (Range = 0 - 1,000,000, Default = 0)<br>**alaSaaMacResultsAvgJitter—**The average jitter value. | | |
| 150 | alaSaaRTTThresholdRedTrap | alaSaaCtrlOwnerIndex,<br>alaSaaCtrlTestIndex,<br>alaSaaCtrlLastRunResult,<br>alaSaaCtrlLastRunTime,<br>alaSaaCtrlRTTThreshold,<br>alaSaaMacResultsAvgRTT | saa | This trap is sent when the RTT threshold is crossed. |
| | | **alaSaaCtrlOwnerIndex—**The Owner name to identify the responsibility of the entries in the table (Default = User).<br>**alaSaaCtrlTestIndex—**Unique name to identify the entries in the table. The name is unique across various SNMP users (up to 32 characters).<br>**alaSaaCtrlLastRunResult—**The result of the latest SAA test iteration (Undertermined (0), Success (1), Failed (2), Aborted (3)).<br>**alaSaaCtrlLastRunTime—**The time the last iteration of the SAA was run.<br>**alaSaaCtrlRTTThreshold—**The configured RTT Threshold, micro-seconds (Range = 0 - 1,000,000, Default = 0)<br>**alaSaaMacResultsAvgRTT—**The average Round Trip Time. | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 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. |
| **physicalIndex—**The physical index of the involved object.<br>**baseMacAddress—**The base MAC Address. | | | | |
| 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. |
| **virtualChassisUpgradeCompleteStatus—**The Virtual Chassis upgrade completion status (Success (1), Failure (2)). | | | | |
| 154 | appFPSignatureMatchTrap | alaAppFPPort, alaAppFPDbAppGroupName, alaAppFPDbAppName, alaAppFPDbSrcMacAddr, alaAppFPDbVlanId, alaAppFPDbSrcIpAddrType, alaAppFPDbSrcIpAddr, alaAppFPDbSrcPort | app fingerprint | This trap is sent when a traffic flow matches an application signature. |
| **alaAppFPPort—**The port on which the flow was classified<br>**alaAppFPDbAppGroupName—**The name of the application group and signature that matched the flow.<br>**alaAppFPDbAppName—**The name of the application that matched the flow.<br>**alaAppFPDbSrcMacAddr—**The source MAC addresses of the flow.<br>**alaAppFPDbVlanId—**The VLAN on which the flow was learned and forwarded.<br>**alaAppFPDbSrcIpAddrType—**The source IP address type.<br>**alaAppFPDbSrcIpAddr—**The source IP address.<br>**alaAppFPDbSrcPort—**The source port number. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 155 | virtualChassisVflSpeedTypeChange | virtualChassisOperChasId, virtualChassisVflId, virtualChassisVflSpeedType | virtual-chassis | This trap is sent when the VFL speed type is changed. |
| **virtualChassisOperChasId—**The operational Virtual-Chassis ID.<br>**virtualChassisVflId—**The Virtual Fabric Link Interface If Index.<br>**virtualChassisVflSpeedType—**The Virtual Chassis VFL speed type:<br>• Unassigned - VFL speed type is unassigned.<br>• Unknown - VFL speed is unknown.<br>• Mismatch - This VFL has member ports operating at different speeds.<br>• Ten GB - All member ports of this VFL are operating at 10 Gbps.<br>• Forty GB - All member ports of this VFL are operating at 40 Gbps. | | | | |
| 156 | alaSIPSnoopingACLPreemptedBySOSCall | physicalIndex, alaSIPSnoopingEndedCallIpAddrA, alaSIPSnoopingEndedCallIpAddrB, alaSIPSnoopingEndedCallL4portA, alaSIPSnoopingEndedCallL4portB | sip snooping | This trap is sent when a SIP snooping RTP/RTCP ACL entry is preempted by an SOS call. |
| **physicalIndex—**The physical index of the involved object.<br>**alaSIPSnoopingEndedCallIpAddrA—**The Ended Call IP address for direction A to B.<br>**alaSIPSnoopingEndedCallIpAddrB—**The Ended Call IP address for direction B to A.<br>**alaSIPSnoopingEndedCallL4portA—**The Ended call L4port for call direction A to B.<br>**alaSIPSnoopingEndedCallL4portB—**The Ended call L4port for call direction A to B. | | | | |
| 157 | alaSIPSnoopingRTCPOverThreshold | alaSIPSnoopingActiveCallIpAddrA, alaSIPSnoopingActiveCallIpAddrB, alaSIPSnoopingActiveCallL4portB, alaSIPSnoopingActiveCallSipMediaType, alaSIPSnoopingCallViolationType | sip snooping | This trap is sent when one or more RTCP parameters exceeds the threshold limit. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| **alaSIPSnoopingActiveCallIpAddrA—**The Active Call IP address for direction A to B.<br>**alaSIPSnoopingActiveCallIpAddrB—**The Active Call IP address for direction B to A.<br>**alaSIPSnoopingActiveCallL4portB—**The Active call L4port for call direction B to A.<br>**alaSIPSnoopingCallViolationType—**The type of the Active Call violation:<br>jitterViolation (1), --jitter violation:<br>• RTD<br>• MOS<br>• Rfactor<br>• Packet Loss | | | | |
| 158 | alaSIPSnoopingRTCPPktsLost | physicalIndex | sip snooping | This trap is sent when RTCP packets are lost due to rate limiting. |
| **physicalIndex—**The physical index of the involved object. | | | | |
| 159 | alaSIPSnoopingSignallingLost | physicalIndex | sip snooping | This trap is sent when SIP signaling messages are lost due to rate limiting. |
| **physicalIndex—**The physical index of the involved object. | | | | |
| 160 | alaSIPSnoopingCallRecordsFileMoved | alaSIPSnoopingThresholdNumberOfCalls | sip snooping | 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_record.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 | alaIPv6NeighborLimit | ipv6 | This trap is sent when the system-wide neighbor cache limit is exceeded. |
| **alaIPv6NeighborLimit**—The system-wide maximum size of the neighbor cache. A value of 0 indicates that no limit will be enforced. The minimumvalue is 200 entries. | | | | |
| 162 | alaIPv6NeighborVRFLimitExceeded | alaVirtualRouterName, alaIPv6NeighborVRFLimit | ipv6 | This trap is sent when a per-VRF neighbor cache limit is exceeded. |
| **alaVirtualRouterName**—The neighbor router name.<br>**alaIPv6NeighborLimit**—The system-wide maximum size of the neighbor cache. A value of 0 indicates that no limit will be enforced. The minimumvalue is 200 entries. | | | | |
| 163 | alaIPv6InterfaceNeighborLimitExceed | ipv6IfIndex, alaIPv6NeighborVRFLimit | ipv6 | This trap is sent when a per-interface neighbor cache limit is exceeded. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| **ipv6IfIndex**—The ipv6IfIndex.<br>**alaIPv6NeighborLimit**—The system-wide maximum size of the neighbor cache. A value of 0 indicates that no limit will be enforced. The minimumvalue is 200 entries. | | | | |
| 164 | alaDyingGaspTrap | alaDying-GaspSlot, alaDyingGasp-PowerSupplyType, 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.<br>**alaDyingGaspPowerSupplyType**—The type of the power supply.<br>**alaDyingGaspTime**—The time of the failure. | | | | |
| 165 | alaDhcpSrvLeaseUtilizationThreshold | alaDhcpSrv-LeaseThreshold Status, alaDhcpSrv-SubnetDescript or | 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.<br>**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 | alaDHCPv6SrvLeaseUtilizationThres hold | alaDhcpv6Srv-LeaseThreshold Status, alaDHCPv6Srv SubnetDescri ptor | dhcp v6 | This trap is sent when the lease utilization on a subnet exceeds or falls below the configured threshold value. |
| **alaDhcpv6SrvLeaseThresholdStatus—**The threshold status of subnet utilization:<br>• Crossed Below 80 Percent of Threshold (1)<br>• Crossed Above 80 Percent of Threshold (2)<br>• Reached 100 Percent of Threshold (3)<br>**alaDHCPv6SrvSubnetDescriptor—**The IPv6 address. | | | | |
| 167 | smgrServiceStatusChange | alaSvcId alaSvcType alaSvcOperStat us alaSvcIsid alaSvcBVlan alaSvcMulticast Mode | service manager | 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. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| | **alaSvcId—**The service identifier.<br>**alaSvcType—**The service type (e.g., vpls, spb).<br>**alaSvcIsid—**The I-Domain Service Indentifier (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.<br>**alaSvcBVlan—**The Backbone VLAN ID (BVLAN), which defines the B-Domain for the PBB traffic.<br>**alaSvcMulticastMode—** The multicast replication mode for each service:<br>• 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.<br>• Tandem (2) - where a non-unicast packet entering the SAP port is replicated once at each node using the multicast group address. | | | |
| 168 | smgrSapStatusChange | alaSvcId<br>alaSapPortId<br>alaSapEncapValue<br>alaSapOperStatus | service manager | 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. |
| | **alaSvcId—**The service identifier.<br>**alaSapPortId—**The ID of the access port where this SAP is defined.<br>**alaSapEncapValue—**The value of the label used to identify this SAP on the access port specified by the SAP Port ID.<br>**alaSapOperStatus—**The operational state of the SAP:<br>• up (1)<br>• down (2)<br>• ingressQosMismatch (3)<br>• egressQosMismatch (4)<br>• portMtuTooSmall (5)<br>• svcAdminDown (6)<br>• iesIfAdminDown (7) | | | |
| 169 | smgrSdpStatusChange | alaSdpId<br>alaSdpOperStatus<br>alaSdpNetworkPort<br>alaSdpBVlan<br>alaSdpSystemId<br>alaSdpSystemName<br>alaSdpDynamicType<br>alaSdpIsid | service manager | 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**—SDP identifier.<br>**alaSdpOperStatus—**The operational state of this SDP:<br>• up (1)<br>• notAlive (2)<br>• notReady (3)<br>• invalidEgressInterface (4)<br>• transportTunnelDown (5)<br>• down (6)<br>• created (7) - dynamically created for SPB<br>• destroyed (8) - dynamically destroyed for SPB.<br>**alaSdpNetworkPort—**The network port where ISIS discovered the neighbor node information (B-VLAN and BMAC).<br>**alaSdpBVlan—**The Backbone VLAN (B-VLAN) where ISIS discovered the neighbor node information (BVLAN and B-MAC).<br>**alaSdpSystemId —**The Backbone MAC (B-MAC) where ISIS discovered the neighbor node information (B-VLAN and B-MAC).<br>**alaSdpSystemName**—The name of the neighbor associated with the SDP.<br>**alaSdpDynamicType**—The SDP type allocated for Unicast or Multicast Path according to the tunnel type.<br>**alaSdpIsid**—The I-Domain Service Identifier (I-SID) for the Group MAC assigned to this Multicast SDP. | | | |
| 170 | smgrSdpBindStatusChange | alaSvcId<br>alaSdpBindId<br>alaSdpBindOperStatus<br>alaSdpBindFarEndIpAddress<br>alaSdpBindVnid | service manager | 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. |
| | **alaSvcId**—The Service identifier.<br>**alaSdpBindId**—The SDP Binding identifier.<br>**alaSdpBindOperStatus**—The operational status of this Service-SDP binding:<br>• up (1)<br>• noEgressLabel (2)<br>• noIngressLabel (3)<br>• noLabels (4)<br>• down (5)<br>• svcMtuMismatch (6)<br>• sdpPathMtuTooSmall (7)<br>• sdpNotReady (8)<br>• sdpDown (9)<br>• sapDown (10)<br>• created (11) - dynamically created for SPB<br>• destroyed (12) - dynamically destroyed for SPB.<br>**alaSdpBindFarEndIpAddress**—The Unicast IP address or the Multicast Group Address of the SDP.<br>**alaSdpBindVnid**—The virtual network identifier (VNID). A 24-bit value used to designate the individual VXLAN overlay network on which the communicating VMs are situated. VMs in different VXLAN overlay networks cannot communicate with each other. Value 0 and 0xffffff are currently reserved. | | | |
| 171 | alaPethPwrSupplyConflictTrap | pethPsePortGroupIndex | inline power | This trap is sent when there is a power supply conflict. |
| | **pethPsePortIndex**—The port number. | | | |
| 172 | pethPwrSupplyNotSupportedTrap | pethPsePortGroupIndex<br>pethPsePortIndex | inline power | This trap is sent when a power supply is not supported. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| **pethPsePortGroupIndex—**The slot of the involved object.<br>**pethPsePortIndex—**The port of the involved object. | | | | |
| 173 | chasTrapsBPSLessAllocSysPwr | physicalIndex<br>chastrapsNi-RqstdBpsSysPower<br>chasTrapsNiGrantdBpsSysPower | chassis | This trap is sent when there is insufficient system power being provided by the BPS. |
| **physicalIndex**—The physical index of the involved object.<br>**chastrapsNi- RqstdBpsSysPower**—Requested system power from os-bps.<br>**chasTrapsNiGrantdBpsSysPower**—Granted system power from os-bps. | | | | |
| 174 | chasTrapsBPSStateChange | chasTrapBPSShelfId<br>chasTrapsBPSPowerSupply<br>chasTrapsBPSEventAlert | chassis | This trap is sent when a BPS power supply is inserted or removed. |
| **chasTrapBPSShelfId**—The BPS shelf ID.<br>**chasTrapsBPSPowerSupply**—The BPS power supply involved in the state change trap.<br>**chasTrapsBPSEventAlert**—The event alert. | | | | |
| 175 | chasTrapsNiBPSFETStateChange | chasTrapBPSShelfId<br>chasTrapsBPSFwType<br>chasTrapsBPSFwVersion | chassis | This trap is sent when there is a BPS FET state change. |
| **chasTrapBPSShelfId**—The BPS shelf ID.<br>**chasTrapsBPSFwType**—The FET state.<br>**chasTrapsBPSFwVersion**—The BPS firmware version. | | | | |
| 176 | alaDhcpBindingDuplicateEntry | dhcpSnoopingBindingMacAddress<br>dhcpSnoopingBindingVlan<br>dhcpSnoopingBindingIfIndex | udp relay | This trap is sent when there is MAC Movement in DHCP-Binding Table. |
| **dhcpSnoopingBindingMacAddress**—The MAC address from which the DHCP request originated.<br>**dhcpSnoopingBindingVlan**—The DHCP client VLAN.<br>**dhcpSnoopingBindingIfIndex**—The interface from which the DHCP request originated. | | | | |
| 177 | alaVCSPProtectionTrap | alaVCSPTableChassisID | vcsp | This trap is sent when a virtual chassis enters the split protection state. |
| **alaVCSPTableChassisID—**The chassis ID number. | | | | |
| 178 | alaVCSPRecoveryTrap | alaVCSPTableChassisID | vcsp | This trap is sent when a split virtual chassis enters the active state. |
| **alaVCSPTableChassisID—**The chassis ID number. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 179 | pethPsePortOnOffNotification | pethPsePortGroupIndex | inline power | Indicates if power inline port is or is not delivering power to the a power inline device. |
| **pethPsePortGroupIndex**—The slot of the involved object. | | | | |
| 180 | pethMainPowerUsageOnNotification | none | inline power | Indicates that the power inline usage is above the threshold. |
| **N/A** | | | | |
| 181 | pethMainPowerUsageOffNotification | none | inline power | Indicates that the power inline usage is below the threshold. |
| **N/A** | | | | |
| 182 | chasTrapsBPSFwUpgradeAlert | chasTrapBPSShelfId<br>chasTrapsBPSFwType<br>chasTrapsBPSFwVersion | chassis | This trap is sent when a BPS firmware upgrade is required. |
| **chasTrapBPSShelfId**—The BPS shelf ID.<br>**chasTrapsBPSFwType**—The FET state.<br>**chasTrapsBPSFwVersion**—The BPS firmware version that requires an upgrade. | | | | |
| 183 | alaAppMonAppRecordFileCreated | NA | application monitoring | This trap is sent after the application records monitored in the past hour are written to the flash file. |
| **NA** | | | | |
| 184 | alaAppMonFlowRecordFileCreated | NA | application monitoring | This trap is sent after the pre-configured number of application monitoring flow records are written to the flash file. |
| **NA** | | | | |
| 185 | alaDPIFlowRecordFileCreated | NA | deep packet inspection | This trap is sent after the pre-configured number of deep packet inspection flow records are written to the flash file. |
| **NA** | | | | |
| 186 | alaLbdStateChangeToShutdown | alaLbdPortIfIndex<br>alaLbdPreviousState<br>alaLbdCurrentState | load balancing | This trap is sent when a port is shut down. |
| **alaLbdPortIfIndex**—The ifIndex on which the load balancing trap was raised.<br>**alaLbdPreviousState**—The previous state of the port on which load balancing was running.<br>**alaLbdCurrentState**—The current state of the port on which load balancing was running. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 187 | alaLbdStateChangeForClearViolationAll | alaLbdPortIfIndex<br>alaLbdPreviousStateClearViolationAll,<br>alaLbdCurrentStateClearViolationAll | load balancing | This trap is sent when the port state changes from shutdown due to “clear-violation-all”. |
| **alaLbdPortIfIndex**—The ifIndex on which the load balancing trap was raised.<br>**alaLbdPreviousState**—The state of the port where LBD was running before clear-violation-all applied.<br>**alaLbdCurrentState**—The state of the port where LBD was running after clear-violation-all applied. | | | | |
| 188 | alaLbdStateChangeForAutoRecovery | alaLbdPortIfIndex<br>alaLbdPreviousStateAutoRecovery<br>alaLbdCurrentStateAutoRecovery | load balancing | This trap is sent when a port state changes from shutdown due to the auto-recovery mechanism. |
| **alaLbdPortIfIndex**—The ifIndex on which the load balancing trap was raised.<br>**alaLbdPreviousStateAutoRecovery**—The state of the port where LBD was running before auto-recovery.<br>**alaLbdCurrentStateAutoRecovery**—The state of the port where LBD was running after auto-recovery. | | | | |
| 189 | alaAutoConfigAutoFabricEnableTrap | NA | | This object specifies the threshold status of subnet utilization? |
| **NA** | | | | |
| 190 | alaVMSnoopingVMLearntAlert | alaVMSnoopingLearnedMacAddress<br>alaVMSnoopingLearnedVxlanUdpPort<br>alaVMSnoopingLearnedVxlanVni | vm snooping | This trap is sent when a new Virtual Machine is learned by the system. |
| **alaVMSnoopingLearnedMacAddress**—The MAC address of the virtual machine.<br>**alaVMSnoopingLearnedVxlanUdpPort**—The port on which the virtual machine was learned.<br>**alaVMSnoopingLearnedVxlanVni**—The virtual machine network identifier. | | | | |
| 191 | alaVMSnoopingVMRemovedAlert | alaVMSnoopingLearnedMacAddress | vm snooping | This trap is sent when virtual machine entry ages out or is removed from the database. |
| **alaVMSnoopingLearnedMacAddress**—The MAC address of the virtual machine. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 192 | alaVMSnoopingReservedHwResourceLimit | alaVMSnoopingChassisId<br>alaVMSnoopingNiSlot<br>alaVMSnoopingHwResourceTotal<br>alaVMSnoopingHwResourceUsed | vm snooping | This trap is sent when the reserved hardware resource reaches a cutoff limit. |
| **alaVMSnoopingChassisId**—The Chassis ID.<br>**alaVMSnoopingNiSlot**—The VM Snooping slot number.<br>**alaVMSnoopingHwResourceTotal**—The total number of reserved hardware resources.<br>**alaVMSnoopingHwResourceUsed**—The total number of used hardware resources. | | | | |
| 193 | alaDistArpItfChange | alaDistArpItfIfIndex<br>alaDistArpNiChassis<br>alaDistArpNiSlot,<br>alaDistArpNiDevice | ip | This trap is sent when an interface is re-assigned to a new designated NI. |
| **alaDistArpItfIfIndex**—The IP ifindex of the interface for the ARP statistics.<br>**alaDistArpNiChassis**—The chassis number of the NI.<br>**alaDistArpNiSlot**—The slot number of the NI.<br>**alaDistArpNiDevice**—The device number of the NI. | | | | |
| 194 | alaDistArpNiThreshold | alaDistArpNiChassis<br>alaDistArpNiSlot<br>alaDistArpNiDevice | ip | This trap is sent when the number of ARPs in hardware has reached the reassignment threshold. |
| **aalaDistArpNiChassis**—The chassis number of the NI.<br>**alaDistArpNiSlot**—The slot number of the NI.<br>**alaDistArpNiDevice**—The device number of the NI. | | | | |
| 195 | smgrVxlanSdpBindStatusChange | alaSvcId<br>alaSdpBindId<br>alaSdpBindOperStatus<br>alaSdpBindFarEndIpAddress<br>alaSdpBindVnid | service manager | This trap is sent when there is an change in SDP Bind operating status. An SDP Bind is dynamically created when a VTEP is discovered, or destroyed when the reachability to the VTEP is gone. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| | **alaSvcId**—The Service identifier.<br>**alaSdpBindId**—The SDP Binding identifier.<br>**alaSdpBindOperStatus**—The operational status of this Service-SDP binding:<br>• up (1)<br>• noEgressLabel (2)<br>• noIngressLabel (3)<br>• noLabels (4)<br>• down (5)<br>• svcMtuMismatch (6)<br>• sdpPathMtuTooSmall (7)<br>• sdpNotReady (8)<br>• sdpDown (9)<br>• sapDown (10)<br>• created (11) - dynamically created for SPB<br>• destroyed (12) - dynamically destroyed for SPB.<br>**alaSdpBindFarEndIpAddress**—The Unicast IP address or the Multicast Group Address of the SDP.<br>**alaSdpBindVnid**—The virtual network identifier (VNID). A 24-bit value used to designate the individual VXLAN overlay network on which the communicating VMs are situated. VMs in different VXLAN overlay networks cannot communicate with each other. Value 0 and 0xffffffff are currently reserved. | | | |
| 196 | alaAutoFabricSTPModeChangeAlert | alaAutoFabricSTPMode | fips | This trap is sent when auto-fabric changes STP mode. |
| | **alaAutoFabricSTPMode**—The STP mode. | | | |
| 197 | alaDaKerberosReqTimeoutTrap | alaDaKerberosIpAddress, alaDaKerberosUserMac | da-unp | This trap shall be raised when the KERBEROS server does not reply in time. |
| | **alaDaKerberosIpAddress -** The IP address of the Kerberos server.<br>**alaDaKerberosUserMac -** MAC address of the user. | | | |
| 198 | alaDaKerberosInactivityTimerExpiryTra | alaDaKerberosUserName, alaDaKerberosUserMac, alaDaKerberosUserDomain | da-unp | The trap shall be raised when the KERBEROS lease timer is expired for the user. |
| | **alaDaKerberosUserName** - Name of the user.<br>**alaDaKerberosUserMac -** MAC address of the user.<br>**alaDaKerberosUserDomain** - Domain of the user. | | | |
| 199 | alaDaKerberosRateLimitExceed | alaDaKerberosRateLimitString | da-unp | This trap shall be raised when the kerberos packets exceed the limit. |
| | **alaDaKerberosRateLimitString** - Failure string for the rate limit trap. | | | |
| 200 | pethMainPowerUsageNiFailNotificatio | pethMainPseConsumptionPower | module | This Notification indicates PSE Failed due to power unavailability. |
| | **pethMainPseConsumptionPower** - Measured usage power expressed in watts. | | | |
| 201 | systemSwlogSizeTrap | systemSwlogName | system | The file specified file may get lost if not backed up , since swlog file reached 90% of its size, please back up swlog before getting overwritten. |
| | **systemSwlogName** - SWLOG file name that might get overwritten since swlog file reached (90%) of size. | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 202 | esmStormThresholdViolationStatus | esmStormViolationThresholdNotificationType, esmStormViolationThresholdTrafficType | interface | This object notifies management station if User-Port ports gets the ingress traffic inflow above the configured value. |
| **esmStormViolationThresholdNotificationType** - This type defines the trap generated by storm control feature for high or low threshold.<br>**esmStormViolationThresholdTrafficType** - This type defines the traffic for which the trap generated by storm control feature for high or low threshold. | | | | |
| 203 | alaSTPLoopGuardError | vStpPortConfigIfIndex, vStpInsNumber | stp | This trap is sent by a bridge when a port enters the Loop inconsistent state (ERR state). |
| **vStpPortConfigIfIndex** - The ifindex of the port for which this entry contains Spanning Tree Protocol.<br>**vStpInsNumber -** The Spanning Tree number identifying this instance. In 1x1 mode the accepted range is 1-4095 and corresponds to the VLAN. In Flat mode the range is 0-4095 and corresponds to the Spanning tree instance. | | | | |
| 204 | alaSTPLoopGuardRecovery | vStpPortConfigIfIndex, vStpInsNumber | stp | This trap is sent by a bridge when a port leaves the Loop inconsistent state (ERR state). |
| **vStpPortConfigIfIndex** - The ifindex of the port for which this entry contains Spanning Tree Protocol.<br>**vStpInsNumber -** The Spanning Tree number identifying this instance. In 1x1 mode the accepted range is 1-4095 and corresponds to the VLAN. In Flat mode the range is 0-4095 and corresponds to the Spanning tree instance. | | | | |
| 205 | alaLldpTrustViolation | alaLLDPTrustPortIfIndex, alaLLDPTrustViolationReason | aip | Port configured to LLDP trust agent have violated. |
| **alaLLDPTrustPortIfIndex** - Interface index of the port used for sending the trap.<br>**alaLLDPTrustViolationReason -** Reason for the violation. | | | | |
| 206 | alaLicenseManagerDemoDayAlert | alaLicenseTimeRemaining | licensing | This is trap is sent with the number of days remaining for the demo license. |
| **alaLicenseTimeRemaining**—Generate trap for license manager. | | | | |
| 207 | alaAaaUserCreation | alaAaaUserNotificationInfo | aaa | The trap shall be raised when a user is added to the Network Device. |
| **alaAaaUserNotificationInfo**-AAA User Information Notification. | | | | |
| 208 | alaAaaUserDeletion | alaAaaUserNotificationInfo | aaa | The trap shall be raised when a new user is deleted from the Network Device. |
| **alaAaaUserNotificationInfo**-AAA User Information Notification. | | | | |
| 209 | alaAaaUserModification | alaAaaUserNotificationInfo | aaa | The trap shall be raised when a user is modified on the Network Device. |
| **alaAaaUserNotificationInfo**-AAA User Information Notification. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 210 | systemSwlogFailureTraps | systemSwlogFailure | system | When swlog fails to store log message to /flash/swlog_chassis file, then this trap is sent to the Management Entity with the message which was failed to persist. This trap will also be raised during failure of sending the swlog to external syslog server. |
| **systemSwlogFailure**-SWLOG to /flash/swlog_chassis file may fail due to the switch running out of file descriptor or space. This trap will also be raised during failure of sending the swlog to external syslog server. | | | | |
| 211 | pethPseMainTemperatureUpAlert | pethPsePortGroupIndex | module | Temperature Up Threshold Alert, Power budget reduced. |
| **pethPsePortGroupIndex-**A port associated with a temperature alert. | | | | |
| 212 | pethPseMainTemperatureDownAlert | pethPsePortGroupIndex | module | Temperature Down Threshold Alert, Power budget reconfigured. |
| **pethPsePortGroupIndex-**A port associated with a temperature alert. | | | | |
| 213 | systemRebootSwlogFailureTrap | systemRebootReason | system | When swlog fails to send log message to remote server or log service is not running, the system need to be rebooted. This trap is sent to the management entity to indicate the reboot reason. |
| **systemRebootReason**-This object indicates the system reboot reason. If this object value is swlogFailure(1) that means system reboot due to remote logging failure. If this object value is loggingServiceNotRunning(2) that means system reboot because syslogd/swlogd is not running. | | | | |
| 214 | ospfv3RestartStatusChange | ospfv3RouterId<br>ospfv3RestartStatus<br>ospfv3RestartInterval<br>ospfv3RestartExitReason | ospfv3 | An ospfv3RestartStatusChange notification signifies that there has been a change in the graceful restart state for the router. This notification should be generated when the router restart status changes. |
| **ospfv3RouterId-** The originator of the notification.<br>**ospfv3RestartStatus -** The current status of OSPF graceful restart capability.<br>**ospfv3RestartInterval -** Configured OSPF graceful restart timeout interval.<br>**ospfv3RestartExitReason-**Describes the outcome of the last attempt at a graceful restart. | | | | |
| 215 | ospfv3NbrRestartHelperStatusChange | ospfv3RouterId<br>ospfv3NbrRestartHelperStatus<br>ospfv3NbrRestartHelperAge<br>ospfv3NbrRestartHelperExitReason | ospfv3 | An ospfv3NbrRestartHelperStatusChange notification signifies that there has been a change in the graceful restart helper state for the neighbor. This notification should be generated when the neighbor restart helper status transitions for a neighbor. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| | **ospfv3RouterId-** The originator of the notification.<br>**ospfv3RestartNbrHelperStatus -** Indicates whether the router is acting as a graceful restart helper for the neighbor.<br>**ospfv3RestartNbrHelperAge -** Remaining time in current OSPF graceful restart interval, if the router is acting as a restart helper for the neighbor.<br>**ospfv3RestartNbrHelperExitReason-**Describes the outcome of the last attempt at acting as a graceful restart helper for the neighbor. | | | |
| 216 | ospfv3VirtNbrRestartHelperStatusChange | ospfv3RouterId<br>ospfv3VirtNbrRestartHelperStatus<br>ospfv3VirtNbrRestartHelperAge<br>ospfv3VirtNbrRestartHelperExitReason | ospfv3 | An ospfv3VirtNbrRestartHelperStatusChange notification signifies that there has been a change in the graceful restart helper state for the virtual neighbor. This notification should be generated when the virtual neighbor restart helper status transitions for a virtual neighbor. |
| | **ospfv3RouterId-** The originator of the notification.<br>**ospfv3VirtRestartNbrHelperStatus -** Indicates whether the router is acting as a graceful restart helper for the neighbor.<br>**ospfv3VirtRestartNbrHelperAge -** Remaining time in current OSPF graceful restart interval, if the router is acting as a restart helper for the neighbor.<br>**ospfv3VirtRestartNbrHelperExitReason-**Describes the outcome of the last attempt at acting as a graceful restart helper for the neighbor. | | | |
| 217 | smgrL2greSdpBindStatusChange | alaSvcId<br>alaSdpBindId<br>alaSdpBindOperStatus<br>alaSdpBindFarEndIpAddress | svcmgr | A change in SDP Bind operating status. A SDP Bind is dynamically created as layer 2 GRE tunnel. |
| | **alaSvcId-**The value of the object alaSvcId specifies the Service identifier. This value should be unique within the service domain.<br>**alaSdpBindId-**SDP Binding identifier.<br>**alaSdpBindOperStatus-**The value of alaSdpBindOperStatus indicates the operating status of this Service-SDP binding. 'up' The Service-SDP binding is operational. 'noEgressLabel' The ingress label is available but the egress one is missing. 'noIngressLabel' The egress label is available but the ingress one is not. 'noLabels' Both the ingress and the egress labels are missing. 'down' The binding is administratively down. 'svcMtuMismatch' Both labels are available, but a service MTU mismatch was detected between the local and the far-end devices. 'sdpPathMtuTooSmall' The operating path MTU of the corresponding SDP is smaller than the service MTU. 'sdpNotReady' The SDP's signaling session is down. 'sdpDown' The SDP is not operationally up. 'sapDown' The SAP associated with the service is down.<br>**alaSdpBindFarEndIpAddress-**This object specifies the Unicast IP address or the Multicast Group Address of the SDP. | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 218 | dot3OamThresholdEvent | dot3OamEventLogTimestamp<br>dot3OamEventLogOui<br>dot3OamEventLogType<br>dot3OamEventLogLocation<br>dot3OamEventLogWindowHi<br>dot3OamEventLogWindowLo<br>dot3OamEventLogThresholdHi<br>dot3OamEventLogThresholdLo<br>dot3OamEventLogValue<br>dot3OamEventLogRunningTotal<br>dot3OamEventLogEventTotal | bridge | A dot3OamThresholdEvent notification is sent when a local or remote threshold crossing event is detected. A local threshold crossing event is detected by the local entity, while a remote threshold crossing event is detected by the reception of an Ethernet OAM Event Notification OAMPDU that indicates a threshold event. This notification should not be sent more than once per second. The OAM entity can be derived from extracting the ifIndex from the variable bindings. The objects in the notification correspond to the values in a row instance in the dot3OamEventLogTable. The management entity should periodically check dot3OamEventLogTable to detect any missed events. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|

**dot3OamEventLogTimestamp**-The value of sysUpTime at the time of the logged event. For locally generated events, the time of the event can be accurately retrieved from sysUpTime. For remotely generated events, the time of the event is indicated by the reception of the Event Notification OAMPDU indicating that the event occurred on the peer. A system may attempt to adjust the timestamp value to more accurately reflect the time of the event at the peer OAM entity by using other information, such as that found in the timestamp found of the Event Notification TLVs, which provides an indication of the relative time between events at the peer entity.

**dot3OamEventLogOui**-The OUI of the entity defining the object type. All IEEE 802.3 defined events (as appearing in [802.3ah] except for the Organizationally Unique Event TLVs) use the IEEE 802.3 OUI of 0x0180C2. Organizations defining their own Event Notification TLVs include their OUI in the Event Notification TLV that gets reflected here.

**dot3OamEventLogType**-The type of event that generated this entry in the event log. When the OUI is the IEEE 802.3 OUI of 0x0180C2, the following event types are defined:
erroredSymbolEvent(1),erroredFramePeriodEvent(2),erroredFrameEvent(3),
erroredFrameSecondsEvent(4),linkFault(256),dyingGaspEvent(257),criticalLinkEvent(258). The first four are considered threshold crossing events, as they are generated when a metric exceeds a given value within a specified window. The other three are not threshold crossing events. When the OUI is not 71874 (0x0180C2 in hex), then some other organization has defined the event space. If event subtyping
is known to the implementation, it may be reflected here. Otherwise, this value should return all F's (2^32 - 1).

**dot3OamEventLogLocation**-Whether this event occurred locally (local(1)), or was received from the OAM peer via Ethernet OAM (remote(2)).

**dot3OamEventLogWindowHi**-If the event represents a threshold crossing event, the two objects dot3OamEventWindowHi and dot3OamEventWindowLo, form an unsigned 64-bit integer yielding the window over which the value was measured for the threshold crossing event (for example, 5, when 11 occurrences happened in 5 seconds while the threshold was 10). The two objects are combined as:
dot3OamEventLogWindow = ((2^32) * dot3OamEventLogWindowHi) + dot3OamEventLogWindowLo Otherwise, this value is returned as all F's (2^32 - 1) and adds no useful information.

**dot3OamEventLogWindowLo**-If the event represents a threshold crossing event, the two objects dot3OamEventWindowHi and dot3OamEventWindowLo form an unsigned 64-bit integer yielding the window over which the value was measured for the threshold crossing event (for example, 5, when 11 occurrences happened in 5 seconds while the threshold was 10). The two objects are combined as:
dot3OamEventLogWindow = ((2^32) * dot3OamEventLogWindowHi) + dot3OamEventLogWindowLo. Otherwise, this value is returned as all F's (2^32 - 1) and adds no useful information.

**dot3OamEventLogThresholdHi**-If the event represents a threshold crossing event, the two objects dot3OamEventThresholdHi and dot3OamEventThresholdLo form an unsigned 64-bit integer yielding the value that was crossed for the threshold crossing event (for example, 10, when 11 occurrences happened in 5 seconds while the threshold was 10). The two objects are combined as:
dot3OamEventLogThreshold = ((2^32) * dot3OamEventLogThresholdHi) + dot3OamEventLogThresholdLo. Otherwise, this value is returned as all F's (2^32 -1) and adds no useful information.

**dot3OamEventLogThresholdLo**-If the event represents a threshold crossing event, the two objects dot3OamEventThresholdHi and dot3OamEventThresholdLo form an unsigned 64-bit integer yielding the value that was crossed for the threshold crossing event (for example, 10, when 11 occurrences happened in 5 seconds while the threshold was 10). The two objects are combined as:
dot3OamEventLogThreshold = ((2^32) * dot3OamEventLogThresholdHi) + dot3OamEventLogThresholdLo. Otherwise, this value is returned as all F's (2^32 - 1) and adds no useful information.

**dot3OamEventLogValue**-If the event represents a threshold crossing event, this value indicates the value of the parameter within the given window that generated this event (for example, 11, when 11 occurrences happened in 5 seconds while the threshold was 10). Otherwise, this value is returned as all F's (2^64 - 1) and adds no useful information.

**dot3OamEventLogRunningTotal**-Each Event Notification TLV contains a running total of the
number of times an event has occurred, as well as the number of times an Event Notification for the event has been transmitted. For non-threshold crossing events, the number of events (dot3OamLogRunningTotal) and the number of resultant Event Notifications dot3OamLogEventTotal) should be identical. For threshold crossing events, since multiple occurrences may be required to cross the threshold, these values are likely different. This value represents the total number of times this event has happened since the last reset (for example, 3253, when 3253 symbol errors have occurred since the last reset, which has resulted in 51 symbol error threshold crossing events since the last reset).

**dot3OamEventLogEventTotal**-Each Event Notification TLV contains a running total of the number of times an event has occurred, as well as the number of times an Event Notification for the event has been transmitted. For non-threshold crossing events, the number of events (dot3OamLogRunningTotal) and the number of resultant
Event Notifications (dot3OamLogEventTotal) should be identical. For threshold crossing events, since multiple occurrences may be required to cross the threshold, these values are likely different. This value represents the total number of times one or more of these occurrences have resulted in an Event Notification (for example, 51 when 3253 symbol errors have occurred since the last reset, which has resulted in 51 symbol error threshold crossing events since the last reset).

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 219 | dot3OamNonThresholdEvent | dot3OamEventLogTimestamp<br>dot3OamEventLogOui<br>dot3OamEventLogType<br>dot3OamEventLogLocation<br>dot3OamEventLogEventTotal | bridge | A dot3OamNonThresholdEvent notification is sent when a local or remote non-threshold crossing event is detected. A local event is detected by the local entity, while a remote event is detected by the reception of an Ethernet OAM Event Notification OAMPDU that indicates a non-threshold crossing event. This notification should not be sent more than once per second. The OAM entity can be derived from extracting the ifIndex from the variable bindings. The objects in the notification correspond to the values in a row instance of the dot3OamEventLogTable. The management entity should periodically check dot3OamEventLogTable to detect any missed events. |

**dot3OamEventLogTimestamp**-The value of sysUpTime at the time of the logged event. For locally generated events, the time of the event can be accurately retrieved from sysUpTime. For remotely generated events, the time of the event is indicated by the reception of the Event Notification OAMPDU indicating that the event occurred on the peer. A system may attempt to adjust the timestamp value to more accurately reflect the time of the event at the peer OAM entity by using other information, such as that found in the timestamp found of the Event Notification TLVs, which provides an indication of the relative time between events at the peer entity.

**dot3OamEventLogOui-**The OUI of the entity defining the object type. All IEEE 802.3 defined events (as appearing in [802.3ah] except for the Organizationally Unique Event TLVs) use the IEEE 802.3 OUI of 0x0180C2. Organizations defining their own Event Notification TLVs include their OUI in the Event Notification TLV that gets reflected here.

**dot3OamEventLogType-**The type of event that generated this entry in the event log. When the OUI is the IEEE 802.3 OUI of 0x0180C2, the following event types are defined: erroredSymbolEvent(1),erroredFramePeriodEvent(2),erroredFrameEvent(3), erroredFrameSecondsEvent(4),linkFault(256),dyingGaspEvent(257),criticalLinkEvent(258). The first four are considered threshold crossing events, as they are generated when a metric exceeds a given value within a specified window. The other three are not threshold crossing events. When the OUI is not 71874 (0x0180C2 in hex), then some other organization has defined the event space. If event subtyping is known to the implementation, it may be reflected here. Otherwise, this value should return all F's (2^32 - 1).

**dot3OamEventLogLocation-**Whether this event occurred locally (local(1)), or was received from the OAM peer via Ethernet OAM (remote(2)).

**dot3OamEventLogEventTotal-**Each Event Notification TLV contains a running total of the number of times an event has occurred, as well as the number of times an Event Notification for the event has been transmitted. For non-threshold crossing events, the number of events (dot3OamLogRunningTotal) and the number of resultant

Event Notifications (dot3OamLogEventTotal) should be identical. For threshold crossing events, since multiple occurrences may be required to cross the threshold, these values are likely different. This value represents the total number of times one or more of these occurrences have resulted in an Event Notification (for example, 51 when 3253 symbol errors have occurred since the last reset, which has resulted in 51 symbol error threshold crossing events since the last reset).

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 220 | alaDot3OamThresholdEventClear | dot3OamEventLogTimestamp<br>dot3OamEventLogOui<br>dot3OamEventLogType<br>dot3OamEventLogLocation<br>dot3OamEventLogEventTotal | svcmgr | An alaDot3OamThresholdEventClear notification is sent when a local or remote threshold crossing event is recovered. The OAM entity can be derived from extracting the ifIndex from the variable bindings. The objects in the notification correspond to the values in a row instance in the dot3OamEventLogTable. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|

**dot3OamEventLogTimestamp**-The value of sysUpTime at the time of the logged event. For locally generated events, the time of the event can be accurately retrieved from sysUpTime. For remotely generated events, the time of the event is indicated by the reception of the Event Notification OAMPDU indicating that the event occurred on the peer. A system may attempt to adjust the timestamp value to more accurately reflect the time of the event at the peer OAM entity by using other information, such as that found in the timestamp found of the Event Notification TLVs, which provides an indication of the relative time between events at the peer entity.

**dot3OamEventLogOui**-The OUI of the entity defining the object type. All IEEE 802.3 defined events (as appearing in [802.3ah] except for the Organizationally Unique Event TLVs) use the IEEE 802.3 OUI of 0x0180C2. Organizations defining their own Event Notification TLVs include their OUI in the Event Notification TLV that gets reflected here.

**dot3OamEventLogType**-The type of event that generated this entry in the event log. When the OUI is the IEEE 802.3 OUI of 0x0180C2, the following event types are defined:
erroredSymbolEvent(1),erroredFramePeriodEvent(2),erroredFrameEvent(3),
erroredFrameSecondsEvent(4),linkFault(256),dyingGaspEvent(257),criticalLinkEvent(258). The first four are considered threshold crossing events, as they are generated when a metric exceeds a given value within a specified window. The other three are not threshold crossing events. When the OUI is not 71874 (0x0180C2 in hex), then some other organization has defined the event space. If event subtyping
is known to the implementation, it may be reflected here. Otherwise, this value should return all F's (2^32 - 1).

**dot3OamEventLogLocation**-Whether this event occurred locally (local(1)), or was received from the OAM peer via Ethernet OAM (remote(2)).

**dot3OamEventLogWindowHi**-If the event represents a threshold crossing event, the two objects dot3OamEventWindowHi and dot3OamEventWindowLo, form an unsigned 64-bit integer yielding the window over which the value was measured for the threshold crossing event (for example, 5, when 11 occurrences happened in 5 seconds while the threshold was 10). The two objects are combined as:
dot3OamEventLogWindow = ((2^32) * dot3OamEventLogWindowHi) + dot3OamEventLogWindowLo Otherwise, this value is returned as all F's (2^32 - 1) and adds no useful information.

**dot3OamEventLogWindowLo**-If the event represents a threshold crossing event, the two objects dot3OamEventWindowHi and dot3OamEventWindowLo form an unsigned 64-bit integer yielding the window over which the value was measured for the threshold crossing event (for example, 5, when 11 occurrences happened in 5 seconds while the threshold was 10). The two objects are combined as:
dot3OamEventLogWindow = ((2^32) * dot3OamEventLogWindowHi) + dot3OamEventLogWindowLo. Otherwise, this value is returned as all F's (2^32 - 1) and adds no useful information.

**dot3OamEventLogThresholdHi**-If the event represents a threshold crossing event, the two objects dot3OamEventThresholdHi and dot3OamEventThresholdLo form an unsigned 64-bit integer yielding the value that was crossed for the threshold crossing event (for example, 10, when 11 occurrences happened in 5 seconds while the threshold was 10). The two objects are combined as:
dot3OamEventLogThreshold = ((2^32) * dot3OamEventLogThresholdHi) + dot3OamEventLogThresholdLo. Otherwise, this value is returned as all F's (2^32 -1) and adds no useful information.

**dot3OamEventLogThresholdLo**-If the event represents a threshold crossing event, the two objects dot3OamEventThresholdHi and dot3OamEventThresholdLo form an unsigned 64-bit integer yielding the value that was crossed for the threshold crossing event (for example, 10, when 11 occurrences happened in 5 seconds while the threshold was 10). The two objects are combined as:
dot3OamEventLogThreshold = ((2^32) * dot3OamEventLogThresholdHi) + dot3OamEventLogThresholdLo. Otherwise, this value is returned as all F's (2^32 - 1) and adds no useful information.

**dot3OamEventLogValue**-If the event represents a threshold crossing event, this value indicates the value of the parameter within the given window that generated this event (for example, 11, when 11 occurrences happened in 5 seconds while the threshold was 10). Otherwise, this value is returned as all F's (2^64 - 1) and adds no useful information.

**dot3OamEventLogRunningTotal**-Each Event Notification TLV contains a running total of the
number of times an event has occurred, as well as the number of times an Event Notification for the event has been transmitted. For non-threshold crossing events, the number of events (dot3OamLogRunningTotal) and the number of resultant Event Notifications dot3OamLogEventTotal) should be identical. For threshold crossing events, since multiple occurrences may be required to cross the threshold, these values are likely different. This value represents the total number of times this event has happened since the last reset (for example, 3253, when 3253 symbol errors have occurred since the last reset, which has resulted in 51 symbol error threshold crossing events since the last reset).

**dot3OamEventLogEventTotal**-Each Event Notification TLV contains a running total of the number of times an event has occurred, as well as the number of times an Event Notification for the event has been transmitted. For non-threshold crossing events, the number of events (dot3OamLogRunningTotal) and the number of resultant
Event Notifications (dot3OamLogEventTotal) should be identical. For threshold crossing events, since multiple occurrences may be required to cross the threshold, these values are likely different. This value represents the total number of times one or more of these occurrences have resulted in an Event Notification (for example, 51 when 3253 symbol errors have occurred since the last reset, which has resulted in 51 symbol error threshold crossing events since the last reset).

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 221 | alaDot3OamNonThresholdEventClear | dot3OamEventLogTimestamp<br>dot3OamEventLogOui<br>dot3OamEventLogType<br>dot3OamEventLogLocation<br>dot3OamEventLogEventTotal | bridge | A alaDot3OamNonThresholdEventClear notification is sent when a local or remote non-threshold crossing event is recovered. The OAM entity can be derived from extracting the ifIndex from the variable bindings. The objects in the notification correspond to the values in a row instance of the dot3OamEventLogTable. |

**dot3OamEventLogTimestamp-**The value of sysUpTime at the time of the logged event. For locally generated events, the time of the event can be accurately retrieved from sysUpTime. For remotely generated events, the time of the event is indicated by the reception of the Event Notification OAMPDU indicating that the event occurred on the peer. A system may attempt to adjust the timestamp value to more accurately reflect the time of the event at the peer OAM entity by using other information, such as that found in the timestamp found of the Event Notification TLVs, which provides an indication of the relative time between events at the peer entity.

**dot3OamEventLogOui-**The OUI of the entity defining the object type. All IEEE 802.3 defined events (as appearing in [802.3ah] except for the Organizationally Unique Event TLVs) use the IEEE 802.3 OUI of 0x0180C2. Organizations defining their own Event Notification TLVs include their OUI in the Event Notification TLV that gets reflected here.

**dot3OamEventLogType-**The type of event that generated this entry in the event log. When the OUI is the IEEE 802.3 OUI of 0x0180C2, the following event types are defined:
erroredSymbolEvent(1),erroredFramePeriodEvent(2),erroredFrameEvent(3), erroredFrameSecondsEvent(4),linkFault(256),dyingGaspEvent(257),criticalLinkEvent(258). The first four are considered threshold crossing events, as they are generated when a metric exceeds a given value within a specified window. The other three are not threshold crossing events. When the OUI is not 71874 (0x0180C2 in hex), then some other organization has defined the event space. If event subtyping is known to the implementation, it may be reflected here. Otherwise, this value should return all F's (2^32 - 1).

**dot3OamEventLogLocation-**Whether this event occurred locally (local(1)), or was received from the OAM peer via Ethernet OAM (remote(2)).

**dot3OamEventLogEventTotal-**Each Event Notification TLV contains a running total of the number of times an event has occurred, as well as the number of times an Event Notification for the event has been transmitted. For non-threshold crossing events, the number of events (dot3OamLogRunningTotal) and the number of resultant
Event Notifications (dot3OamLogEventTotal) should be identical. For threshold crossing events, since multiple occurrences may be required to cross the threshold, these values are likely different. This value represents the total number of times one or more of these occurrences have resulted in an Event Notification (for example, 51 when 3253 symbol errors have occurred since the last reset, which has resulted in 51 symbol error threshold crossing events since the last reset).

| | | | | |
|---|---|---|---|---|
| 222 | alaAlarmReplayAlarmInputEvent | alaAlarmInputConfigNameInfo | unknown | The trap shall be raised when device receives a new alarm input and alaAlarmInputConfigTrapActionEnable is enabled. |

**alaAlarmInputConfigNameInfo-**Alarm Management Information Notification.

| | | | | |
|---|---|---|---|---|
| 223 | alaHWRoutingCapacityExceeded | alaIpTrapInfoChassisId | ip | The number of ASIC routing entries has been exceeded. |

**alaIpTrapInfoChassisId-**The Chassis ID.

| | | | | |
|---|---|---|---|---|
| 224 | vrrpv3NewMaster | vrrpv3OperationsMasterIpAddr,<br>vrrpv3StatisticsNewMasterReason | vrrp | The new Master notification indicates that the sending agent has transitioned to master state. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| **vrrpv3OperationsMasterIpAddr** -The master router's real IP address. The master router would set this address to vrrpv3OperationsPrimaryIpAddr while transitioning to master state. For backup routers, this is the IP address listed as the source in the VRRP advertisement last received by this virtual router<br>**vrrpv3StatisticsNewMasterReason** -This indicates the reason for the virtual router to transition to master state. If the virtual router never transitioned to master state, the value of this object is notMaster(0). Otherwise, this indicates the reason this virtual router transitioned to master state the last time. Used by vrrpv3NewMaster notification. | | | | |
| 225 | vrrpv3ProtoError | vrrpv3StatisticsProtoErrReason | vrrp | The notification indicates that the sending agent has encountered the protocol error indicated by vrrpv3StatisticsProtoErrReason. |
| **vrrpv3StatisticsProtoErrReason -** Conformance group for objects contained in VRRPv3 notifications. | | | | |
| 226 | alaDhcpBindingTcamFail | alaDhcpTcamFailMsg | ip-helper | Binding entry creation fail due to TCAM resource failure. |
| **alaDhcpTcamFailMsg -**This object specifies binding entry creation fail due to TCAM resource failure. | | | | |
| 227 | systemStorageLockTrap | systemStorageLockStatus | storage-locking | When system storage is locked or unlocked, this trap is sent to management entity to indicate the storage lock status. |
| **systemStorageLockStatus -** This object indicates the system storage lock status. | | | | |
| 228 | alaDhcpIsfDrop | alaDhcpIsfDropInfo | ip-helper | Trap to notify DHCP ISF drop. |
| **alaDhcpIsfDropInfo -**This object specifies drop information including starting time, ending time, number of dropped packets, chassis, slot and VLAN (optional). | | | | |
| 229 | alaDaRouterAuthUserPassedAuthThresh | sysName, systemServicesDate, systemServicesTime, | da-unp | This trap will be generated when the number of IP addresses that passed authentication reaches the threshold (default 16). |
| **sysName -** An administratively-assigned name for this managed node. By convention, this is the node's fully qualified domain name. If the name is unknown, the value is the zero-length string.<br>**systemServicesDate -** This object contains the current System Date in the following format: MM/DD/YYYY.<br>**systemServicesTime -** This object contains the current System Time in the following format: HH:MM:SS. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 230 | alaDaRouterAuthUserMaxRetryFailed | sysName, systemServicesDate, systemServicesTime, alaDaRouterAuthUserSourceIpAddressType, alaDaRouterAuthUserSourceIpAddress, alaDaRouterAuthUserDestinationIpAddressType, alaDaRouterAuthUserDestinationIpAddress alaDaRouterAuthUserName, alaDaRouterAuthUserAttempts | da-unp | This trap will be generated for every IP address whose number of authentication failure exceeds maximum attempts (default is 3 times. |
| | **sysName -** An administratively-assigned name for this managed node. By convention, this is the node's fully qualified domain name. If the name is unknown, the value is the zero-length string.<br>**systemServicesDate -** This object contains the current System Date in the following format: MM/DD/YYYY.<br>**systemServicesTime -** This object contains the current System Time in the following format: HH:MM:SS.<br>**alaDaRouterAuthUserSourceIpAddressType -** The type of the source IP address of the user.<br>**alaDaRouterAuthUserSourceIpAddress -** Source IP address of the user.<br>**alaDaRouterAuthUserDestinationIpAddressType -** The type of destination IP in the packet.<br>**alaDaRouterAuthUserDestinationIpAddress -** Destination IP in the packet.<br>**alaDaRouterAuthUserName -** User name.<br>**alaDaRouterAuthUserAttempts -**Number of attempts by the user. | | | |
| 231 | alaDaRouterAuthUserFailedAuthThresh | sysName, systemServicesDate, systemServicesTime, | da-unp | This trap will be generated if the number of IP addresses that failed to authenticate reaches the threshold (default 16). |
| | **sysName -** An administratively-assigned name for this managed node. By convention, this is the node's fully qualified domain name. If the name is unknown, the value is the zero-length string.<br>**systemServicesDate -** This object contains the current System Date in the following format: MM/DD/YYYY.<br>**systemServicesTime -** This object contains the current System Time in the following format: HH:MM:SS. | | | |
| 232 | alaDaRouterAuthConfigThresholdExceeded | sysName, systemServicesDate, systemServicesTime, laDaRouterAuthNumberOfConfigUsed | da-unp | This trap will be generated if more than 50 percent of slice or the reserved block is used up for programming the trap rule. It indicates that no more source or destination subnets can be added. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| **sysName -** An administratively-assigned name for this managed node. By convention, this is the node's fully qualified domain name. If the name is unknown, the value is the zero-length string.<br>**systemServicesDate -** This object contains the current System Date in the following format: MM/DD/YYYY.<br>**systemServicesTime -** This object contains the current System Time in the following format: HH:MM:SS.<br>**alaDaRouterAuthNumberOfConfigUsed -**Total number of Router Auth Config used on the system. | | | | |
| 233 | alaDaRouterAuthMaxCapacityReached | sysName, systemServicesDate, systemServicesTime, alaDaRouterAuthNumberOfAuthenticatedUsers | da-unp | This trap will be generated when the total slice capacity of 256 is reached. |
| **sysName -** An administratively-assigned name for this managed node. By convention, this is the node's fully qualified domain name. If the name is unknown, the value is the zero-length string.<br>**systemServicesDate -** This object contains the current System Date in the following format: MM/DD/YYYY.<br>**systemServicesTime -** This object contains the current System Time in the following format: HH:MM:SS.<br>**alaDaRouterAuthNumberOfAuthenticatedUsers -**Total number of router authenticated users on the system. | | | | |
| 234 | alaTestOamTxDoneTrap | alaTestOamConfigTestId, alaTestOamConfigSourceEndpoint, alaTestOamConfigTestIdStatus | bridge | This trap shall be sent to NMS from Generator device, once the test-duration has expired on it. Once the test-duration has expired the Generator device shall send this trap after some time interval to NMS (around 5 to 10 sec). |
| **alaTestOamConfigTestId -** Unique name to identify the entries in the table.<br>**alaTestOamConfigSourceEndpoint -** This object is to identify the local or transmitting device. For bidirectional test, this also identifies the analyzer device.<br>**alaTestOamConfigTestIdStatus -** The test status. | | | | |
| 235 | alaTestOamRxReadyTrap | alaTestOamConfigTestId, alaTestOamConfigDestinationEndpoint, alaTestOamConfigTestIdStatus | bridge | This trap shall be sent to NMS once the device with Analyzer or Loopback Role is ready to receive the test traffic. Once this trap is received on NMS the Generator shall be activated for generating the test traffic. |
| **alaTestOamConfigTestId -** Unique name to identify the entries in the table.<br>**alaTestOamConfigDestinationEndpoint -** This object is to identify the remote device. For unidirectional, this identifies the analyzer device. For bidirectional test, this identifies the device that needs to activate the loopback function.<br>**alaTestOamConfigTestIdStatus -** The test status. | | | | |
| 236 | alaTestOamTestAbortTrap | alaTestOamConfigTestId | bridge | This trap shall be send to NMS from the device, if the test is aborted during takeover or if any of the NI goes down. |
| **alaTestOamConfigTestId -** Unique name to identify the entries in the table. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 237 | alaTestOamGroupTxDoneTrap | alaTestOamConfigGroupId,<br>alaTestOamGroupConfigSourceEndpoint,<br>alaTestOamGroupConfigStatus | bridge | This trap shall be send to NMS from Generator device once the test-duration for the Test OAM Group has expired on it. Once the test-duration has expired the Generator device shall send this trap after some time interval to NMS (around 5 to 10 sec). |
| **alaTestOamConfigGroupId -** Unique Name to Identify the Test OAM Group entries in the table.<br>**alaTestOamGroupConfigSourceEndpoint -** This Object is to identify the local or transmitting device. For bidirectional test, this also identifies the analyzer device**.**<br>**alaTestOamGroupConfigStatus -**Test OAM Group Status. | | | | |
| 238 | alaTestOamGroupRxReadyTrap | alaTestOamConfigGroupId,<br>alaTestOamGroupConfigDestinationEndpoint,<br>alaTestOamGroupConfigStatus | bridge | This trap shall be send to NMS once the device with Analyzer or Loopback Role is ready to receive the test traffic. Once this trap is received on NMS the Generator shall be activated for generating the test traffic for the Test OAM Group. |
| **alaTestOamConfigGroupId -** Unique Name to Identify the Test OAM Group entries in the table.<br>**alaTestOamGroupConfigdestinationEndpoint -** This object is to identify the remote device for test group. For unidirectional, this identifies the analyzer device. For bidirectional test, this identifies the device that needs to activate the loopback function.<br>**alaTestOamGroupConfigStatus -**Test OAM Group Status. | | | | |
| 239 | alaTestOamGroupAbortTrap | alaTestOamConfigGroupId | bridge | This trap shall be send to NMS from the device, if the Test is aborted for Test OAM Group during takeover or if any of the NI goes down. |
| **alaTestOamConfigGroupId -** Unique Name to Identify the Test OAM Group entries in the table. | | | | |
| 240 | alaTestOamStatsWriteDoneTrap | alaTestOamStatsWriteDoneTrapStr | bridge | This trap shall be sent to NMS from the device if the maximum number of stats records (64) have been written to the testoam stats file maintained in /flash. |
| **alaTestOamStatsWriteDoneTrapStr -**The string mentioning that the maximum of records have been written on flash file. | | | | |
| 241 | alaDhcpv6IsfTcamFail | alaDhcpv6IsfTcamFailMsg | ip | Binding entry creation fail due to TCAM resource failure. |
| **alaDhcpv6IsfTcamFailMsg -** This object specifies binding entry creation fail due to TCAM resource failure. | | | | |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| 242 | alaHWArpCapacityExceeded | alaIpTrapInfoChassisId<br>alaHWArpMaxThresholdState<br>alaHWRoutingCapacityExceededGroup | ip | An alaHwArpCapacityExceeded trap indicates that there has been a change in the maximum HW ARP utilization on this system. When HW ARP utilization reaches HIGH threshold, this trap is generated with the alaHwArpMaxThresholdState set to 'reached'. Threshold levels are configured are as system level capability values. When HW ARP utilization transitions back to below LOW threshold utilization, this trap is generated again with the alaHwArpMaxThresholdState set to 'cleared'. |
| **alaIpTrapInfoChassisId -** The Chassis ID.<br>**alaHWArpMaxThresholdState -** Resource above/below threshold limits.<br>**alaHWRoutingCapacityExceededGroup -** A collection of objects to support number of ASIC routing entries has been exceeded. | | | | |
| 243 | alaHWMacCapacityExceeded | alaHWMacMaxThresholdState | capability | The alaHWMacCapacityExceeded trap indicates that there has been a change in the maximum HW MAC utilization on this system. When HW MAC utilization reaches 95%, this trap is generated with the alaHWMacMaxThresholdState set to reached. When HW MAC utilization transitions back to below 90% utilization, this trap is generated again with the alaHWMacMaxThresholdState set to cleared. |
| **alaHWMacMaxThresholdState -** Reached: This trap value indicates if the MAC table had exceeded the configured MAC high threshold limit. Cleared: This trap value indicates if the MAC table had fallen below the configured MAC low threshold limit. | | | | |
| 244 | alaDaUnpMaxUserExceeded | sysName,<br>systemServicesDate,<br>systemServicesTime,<br>alaDaUnpMaxUserSupported,<br>alaDaUnpMaxUserCurrentNumberOfUsers | da-unp | This trap will be generated when max UNP user exceeded in system. |

| No. | Trap Name | Objects | Family | Description |
|---|---|---|---|---|
| | **sysName -** An administratively-assigned name for this managed node. By convention, this is the node's fully qualified domain name. If the name is unknown, the value is the zero-length string.<br>**systemServicesDate -** This object contains the current System Date in the following format: MM/DD/YYYY.<br>**systemServicesTime -** This object contains the current System Time in the following format: HH:MM:SS.<br>**alaDaUnpMaxUserSupported -** Maximum number of users supported on the system.<br>**alaDaUnpMaxUserCurrentNumberOfUsers -** Maximum current number of users. | | | |
| 245 | alaDaUnpHWResourceExhaust | sysName,<br>systemServicesDate,<br>systemServicesTime,<br>alaDaUnpHWResourceChassisId,<br>alaDaUnpHWResourceSlot<br>alaDaUnpHWResourceTtiAllocated | da-unp | This trap will be generated when the TTI resources on a NI slot are exhausted (chassis/slot). |
| | **sysName -** An administratively-assigned name for this managed node. By convention, this is the node's fully qualified domain name. If the name is unknown, the value is the zero-length string.<br>**systemServicesDate -** This object contains the current System Date in the following format: MM/DD/YYYY.<br>**systemServicesTime -** This object contains the current System Time in the following format: HH:MM:SS.<br>**alaDaUnpHWResourceChassisId -** The chassis ID of the switch.<br>**alaDaUnpHWResourceSlot -** The slot number of the chassis.<br>**alaDaUnpHWResourceTtiAllocate -** The number of TTI resources has been allocated. | | | |

## chassisTrapsAlertNumber

The following provides descriptions on the possible values for **chassisTrapsAlertNumber** included in the **chassisTrapsAlert** Trap.

(1) runningWorking - The working version is used.

(2) runningCertified - The certified version is used.

(3) certifyStarted - CERTIFY process started.

(4) certifyFlashSyncStarted - CERTIFY w/FLASH SYNCHRO process started.

(5) certifyCompleted - CERTIFY process completed successfully.

(6) certifyFailed - CERTIFY process failed.

(7) synchroStarted - Flash Synchronization process started.

(8) synchroCompleted - Flash Synchronization completed successfully.

(9) synchroFailed - Flash Synchronization failed.

(10) restoreStarted - RESTORE process started.

(11) restoreCompleted - RESTORE process completed successfully.

(12) restoreFailed - RESTORE process failed.

(13) takeoverStarted - CMM take-over being processed.

(14) takeoverDeferred - CMM take-over deferred.

(15) takeoverCompleted - CMM take-over completed.

(16) macAllocFailed - CMS MAC allocation failed.

(17) macRangeFailed - CMS MAC range addition failed.

(18) fanFailed - One or more of the fans is inoperable.

(19) fanOk - Fan is operable.

(20) fansOk - All fans are operable.

(21) tempOverThreshold - CMM temperature over the threshold.

(22) tempUnderThreshold - CMM temperature under the threshold.

(23) tempOverDangerThreshold - CMM temperature over danger threshold.

(24) powerMissing - Not enough power available.

(25) psNotOperational - Power Supply is not operational.

(26) psOperational - Power supply is operational.

(27) psAllOperational - All power supplies are operationa.l

(28) redundancyNotSupported - Hello protocol disabled, Redundancy not supported.

(29) redundancyDisabledCertifyNeeded - Hello protocol disabled, Certify needed.

(30) cmmStartingAsPrimary - CMM started as primary.

(31) cmmStartingAsSecondary - CMM started as secondary.

(32) cmmStartupCompleted - end of CMM start up.

(33) cmmAPlugged - cmm a plugged.

(34) cmmBPlugged - cmm b plugged.

(35) cmmAUnPlugged - cmm a unplugged.

(36) cmmBUnPlugged - cmm b unplugged.

(37) lowNvramBattery - NV RAM battery is low.

(38) notEnoughFabricsOperational - Not enough Fabric boards operational.

(38) simplexNoSynchro - Only simplex CMM no flash synchro done.

(40) secAutoActivate - secondary CMM autoactivating.

(41) secAutoCertifyStarted - secondary CMM autocertifying.

(42) secAutoCertifyCompleted - secondary CMM autocertify end.

(43) secInactiveReset - cmm b unplugged.

(44) activateScheduled - ACTIVATE process scheduled.

(45) activateStarted - secondary CMM reset because of inactivity.

(46) getAfileCompleted - Get A file process completed.

(47) getAfileFailed - Failed to get a file from other CMM/Stack.

(48) sysUpdateStart - sysUpdate starts.

(49) sysUpdateInProgress - sysUpdate in progress.

(50) sysUpdateError - sysUpdate error.

(51) sysUpdateEnd - sysUpdate ends.

(52) reloadInProgress - the system is already in reload workign process.

(53) c20UpgradeOk - the c20 license upgrade ok.

(54) c20UpgradeFailed - the c20 license upgrade failed.

(55) c20RestoreOk - the c20 license restore ok.

(56) c20RestoreFailed - the c20 license restore failed.

(57) c20NiFailed - the c20 ni board reports failure.

(58) airflowReverse - ps and fan have opposit air flow direction.

(59) tempSWHigh - the cmm/ni temperature is over SW high level.

(60) tempHWHigh - the cmm/ni temperature is over HW high level.

(61) tempDanger - the cmm/ni temperature is over HW danger set level.

(62) imageFileChecksumChanged - the image file MD5 checksum has changed.

(63) cmmDown - A CMM went down.

(64) niDown - an NI went down.

(65) cfmDown - A fabric board went down.

# MIBS Table

The following is a list of the supported MIBs.

- ALCATEL-IND1-AAA-MIB.mib
- ALCATEL-IND1-ALARM-MGR-MIB.mib
- ALCATEL-IND1-ALSRV-MIB.mib
- ALCATEL-IND1-APP-FINGERPRINT-MIB.mib
- ALCATEL-IND1-APP-MON-MIB.mib
- ALCATEL-IND1-AUTO-CONFIG-MIB.mib
- ALCATEL-IND1-AUTO-FABRIC-MIB.mib
- ALCATEL-IND1-BASE.mib
- ALCATEL-IND1-BFD-MIB.mib
- ALCATEL-IND1-BGP-MIB.mib
- ALCATEL-IND1-CAPMAN-MIB.mib
- ALCATEL-IND1-CHASSIS-MIB.mib
- ALCATEL-IND1-CONFIG-MGR-MIB.mib
- ALCATEL-IND1-DA-MIB.mib
- ALCATEL-IND1-DCBX-MIB.mib
- ALCATEL-IND1-DEVICES.mib
- ALCATEL-IND1-DHCPSRV-MIB.mib
- ALCATEL-IND1-DHCPV6-MIB.mib
- ALCATEL-IND1-DHL-MIB.mib
- ALCATEL-IND1-DOT3-OAM-MIB.mib
- ALCATEL-IND1-DP-MIB.mib
- ALCATEL-IND1-DPI-MIB.mib
- ALCATEL-IND1-DVMRP-MIB.mib
- ALCATEL-IND1-E-SERVICE-MIB.mib
- ALCATEL-IND1-EOAM.mib
- ALCATEL-IND1-ERP-MIB.mib
- ALCATEL-IND1-EVB-MIB.mib
- ALCATEL-IND1-EVENT-SCRIPTING-MIB.mib
- ALCATEL-IND1-FIPS-MIB.mib

- ALCATEL-IND1-GLOBALROUTETABLE-MIB.mib
- ALCATEL-IND1-GVRP-MIB.mib
- ALCATEL-IND1-HA-VLAN-MIB.mib
- ALCATEL-IND1-HEALTH-MIB.mib
- ALCATEL-IND1-INLINE-POWER-MIB.mib
- ALCATEL-IND1-INTERSWITCH-PROTOCOL-MIB.mib
- ALCATEL-IND1-IP-MIB.mib
- ALCATEL-IND1-IPMRM-MIB.mib
- ALCATEL-IND1-IPMS-MIB.mib
- ALCATEL-IND1-IPRM-MIB.mib
- ALCATEL-IND1-IPRMV6-MIB.mib
- ALCATEL-IND1-IPSEC-MIB.mib
- ALCATEL-IND1-IPV6-MIB.mib
- ALCATEL-IND1-ISIS-MIB.mib
- ALCATEL-IND1-ISIS-SPB-MIB.mib
- ALCATEL-IND1-KEY-MANAGEMENT-MIB.mib
- ALCATEL-IND1-LAG-MIB.mib
- ALCATEL-IND1-LBD-MIB.mib
- ALCATEL-IND1-LLDP-MED-MIB.mib
- ALCATEL-IND1-LLDP-TRUST-MIB.mib
- ALCATEL-IND1-LPS-MIB.mib
- ALCATEL-IND1-MAC-ADDRESS-MIB.mib
- ALCATEL-IND1-MAC-SERVER-MIB.mib
- ALCATEL-IND1-MSGSRV-MIB.mib
- ALCATEL-IND1-MULTI-CHASSIS-MIB.mib
- ALCATEL-IND1-MVRP-MIB.mib
- ALCATEL-IND1-NETSEC-MIB.mib
- ALCATEL-IND1-NTP-MIB.mib
- ALCATEL-IND1-OPENFLOW-MIB.mib
- ALCATEL-IND1-OSPF-MIB.mib
- ALCATEL-IND1-OSPF3-MIB.mib
- ALCATEL-IND1-PIM-MIB.mib

- ALCATEL-IND1-POLICY-MIB.mib
- ALCATEL-IND1-PORT-MAPPING.mib
- ALCATEL-IND1-PORT-MIB.mib
- ALCATEL-IND1-PORT-MIRRORING-MONITORING-MIB.mib
- ALCATEL-IND1-PPPOE-IA-MIB.mib
- ALCATEL-IND1-PRIVATE-VLAN-MIB.mib
- ALCATEL-IND1-QCN-MIB.mib
- ALCATEL-IND1-QOS-MIB.mib
- ALCATEL-IND1-RDP-MIB.mib
- ALCATEL-IND1-RIP-MIB.mib
- ALCATEL-IND1-RIPNG-MIB.mib
- ALCATEL-IND1-ROUTEMAP-MIB.mib
- ALCATEL-IND1-SAA-MIB.mib
- ALCATEL-IND1-SECY.mib
- ALCATEL-IND1-SERVICE-MGR-MIB.mib
- ALCATEL-IND1-SESSION-MGR-MIB.mib
- ALCATEL-IND1-SIP-SNOOPING-MIB.mib
- ALCATEL-IND1-SLB-MIB.mib
- ALCATEL-IND1-SNMP-AGENT-MIB.mib
- ALCATEL-IND1-STATIC-FRR-MIB.mib
- ALCATEL-IND1-SYSTEM-MIB.mib
- ALCATEL-IND1-TCAM-MIB.mib
- ALCATEL-IND1-TEST-OAM-MIB.mib
- ALCATEL-IND1-TIMETRA-CHASSIS-MIB.mib
- ALCATEL-IND1-TIMETRA-FILTER-MIB.mib
- ALCATEL-IND1-TIMETRA-GLOBAL-MIB.mib
- ALCATEL-IND1-TIMETRA-LDP-MIB.mib
- ALCATEL-IND1-TIMETRA-MPLS-MIB.mib
- ALCATEL-IND1-TIMETRA-OAM-TEST-MIB.mib
- ALCATEL-IND1-TIMETRA-PORT-MIB.mib
- ALCATEL-IND1-TIMETRA-QOS-MIB.mib
- ALCATEL-IND1-TIMETRA-SAP-MIB.mib

- ALCATEL-IND1-TIMETRA-SDP-MIB.mib
- ALCATEL-IND1-TIMETRA-SERV-MIB.mib
- ALCATEL-IND1-TIMETRA-TC-MIB.mib
- ALCATEL-IND1-TIMETRA-VRTR-MIB.mib
- ALCATEL-IND1-TRAP-MGR-MIB.mib
- ALCATEL-IND1-UDLD-MIB.mib
- ALCATEL-IND1-UDP-RELAY-MIB.mib
- ALCATEL-IND1-VC-SPLIT-PROTECTION-MIB.mib
- ALCATEL-IND1-VIRTUAL-CHASSIS-MIB.mib
- ALCATEL-IND1-VIRTUAL-FLOW-CONTROL-MIB.mib
- ALCATEL-IND1-VIRTUALROUTER-MIB.mib
- ALCATEL-IND1-VLAN-MGR-MIB.mib
- ALCATEL-IND1-VLAN-STP-MIB.mib
- ALCATEL-IND1-VM-SNOOPING-MIB.mib
- ALCATEL-IND1-VRRP-MIB.mib
- ALCATEL-IND1-VRRP3-MIB.mib
- ALCATEL-IND1-WEBMGT-MIB.mib
- ALCATEL-IND1-ZEROCONF-MIB.mib
- ATM-TC-MIB.mib
- BGP4-MIB.mib
- BRIDGE-MIB.mib
- DOT3-OAM-MIB.mib
- DVMRP-STD-MIB.mib
- ENERGY-OBJECT-MIB.mib
- ENTITY-MIB.mib
- EtherLike-MIB.mib
- HCNUM-TC.mib
- IANA-ADDRESS-FAMILY-NUMBERS-MIB.mib
- IANA-RTPROTO-MIB.mib
- IANAifType-MIB.mib
- IEEE8021-BRIDGE-MIB.mib
- IEEE8021-CFM-MIB.mib

- IEEE8021-CN-MIB.mib
- IEEE8021-EVBB-MIB.mib
- IEEE8021-PAE-MIB.mib
- IEEE8021-PFC-MIB.mib
- IEEE8021-SECY-MIB.mib
- IEEE8021-SPB-MIB.mib
- IEEE8021-TC-MIB.mib
- IEEE8023-LAG-MIB.mib
- IF-MIB.mib
- IGMP-STD-MIB.mib
- INET-ADDRESS-MIB.mib
- IP-FORWARD-MIB.mib
- IP-MIB.mib
- IPMCAST-MIB.mib
- IPV6-FLOW-LABEL-MIB.mib
- IPV6-ICMP-MIB.mib
- IPV6-MIB.mib
- IPV6-MLD-MIB.mib
- IPV6-TC.mib
- ISIS-MIB.mib
- LANGTAG-TC-MIB.mib
- LLDP-EXT-DOT1-EVB-EXTENSIONS-MIB.mib
- LLDP-EXT-DOT1-MIB.mib
- LLDP-EXT-DOT1-V2-MIB.mib
- LLDP-EXT-DOT3-MIB.mib
- LLDP-EXT-DOT3-V2-MIB.mib
- LLDP-EXT-MED-MIB.mib
- LLDP-MIB.mib
- LLDP-V2-MIB.mib
- LLDP-V2-TC-MIB.mib
- MAU-MIB.mib
- MPLS-LDP-MIB.mib

- MPLS-LSR-MIB.mib
- MPLS-TE-MIB.mib
- OSPF-MIB.mib
- OSPF-TRAP-MIB.mib
- OSPFV3-MIB.mib
- P-BRIDGE-MIB.mib
- PIM-BSR-MIB.mib
- PIM-STD-MIB.mib
- POWER-ETHERNET-MIB.mib
- Q-BRIDGE-MIB.mib
- RIPv2-MIB.mib
- RMON-MIB.mib
- RS-232-MIB.mib
- SFLOW-MIB.mib
- SNMP-COMMUNITY-MIB.mib
- SNMP-FRAMEWORK-MIB.mib
- SNMP-MPD-MIB.mib
- SNMP-NOTIFICATION-MIB.mib
- SNMP-PROXY-MIB.mib
- SNMP-TARGET-MIB.mib
- SNMP-USER-BASED-SM-MIB.mib
- SNMP-USM-AES-MIB.mib
- SNMP-VIEW-BASED-ACM-MIB.mib
- SNMPv2-MIB.mib
- TCP-MIB.mib
- TOPSEC-ENT1-DEVICES.mib
- TUNNEL-MIB.mib
- UDP-MIB.mib
- VRRP-MIB.mib
- VRRPV3-MIB.mib

# System Events

The following table provides information about important system events and trap events on the switch in user-friendly and customer readable format. Unlike AOS syslog, Readable Customer Event only provides important events which are listed in the below tables.

**Note.** For information on configuring and viewing customer event logs, see Chapter 37, "Using Switch Logging" in *OmniSwitch AOS Release 8 Network Configuration Guide*.

## System Events

| No. | System Events | Application | User Readable Log Description |
|---|---|---|---|
| 1 | System Ready | Chassis Supervision | System Ready |
| 2 | VC take-over | Chassis Supervision/VCM | Sending VC Takeover to NIs and applications [L6] |
| 3 | Reboot | Chassis Supervision | System Reboot |

## TRAP Events

| No | Trap Name | Family | User Readable Log Description |
|---|---|---|---|
| 1 | coldStart | chassis | The switch was restarted by a power cycle or due to some type of failure |
| 2 | warmStart | chassis | The switch was restarted by the user |
| 3 | linkDown | interface | Link c/s/p operationally down |
| 4 | linkUp | interface | Link c/s/p operationally up |
| 5 | authenticationFailure | snmp | SNMP message authentication failed |
| 6 | chassisTrapsAlert | chassis | (The chassis trap alert) |
| 7 | chassisTrapsStateChange | chassis | (The state change) |
| 8 | chassisTrapsMacOverlap | module | MAC Range overlap found in backplane EEPROM |
| 9 | healthMonModuleTrap | health | NI c/s rising above or falling below cpu/ memory threshold (depending on rising/ falling) |
| 10 | healthMonPortTrap | health | Port c/s/p rising above or falling below threshold (depending on rising/falling) |
| 11 | healthMonCmmTrap | health | CMM chassis-id (plus cmmA or cmmB, if chassis) rising above or falling below cpu/ memory threshold (depending on rising/ falling) |

| No | Trap Name | Family | User Readable Log Description |
|---|---|---|---|
| 12 | esmDrvTrapDropsLink | interface | Link c/s/p dropped due to excessive errors |
| 13 | portViolationTrap | interface | Port c/s/p in violation - source <info> reason <info> |
| 14 | risingAlarm | rmon | Alarm entry <info> crossing rising threshold |
| 15 | fallingAlarm | rmon | Alarm entry <info> crossing falling threshold |
| 16 | stpNewRoot | stp | STP instance <value> : Bridge has become new Root |
| 17 | stpRootPortChange | stp | STP instance <value> : Root port change detected |
| 18 | sessionAuthenticationTrap | session | Authentication failure detected: user <info> |
| 19 | alaDoSTrap | ip | Denial of Service attack detected: <info> |
| 20 | lnkaggAggUp | linkaggregation | Link Aggregation <value> operationally up |
| 21 | lnkaggAggDown | linkaggregation | Link Aggregation <value> operationally down |
| 22 | chassisTrapsPossibleDuplicateMac | chassis | The old Master chassis cannot be detected in the virtual chassis. There is a possiblity of duplicate MAC address in the network. |
| 23 | alaErpRingStateChanged | bridge | ERP Ring <value>: State changed from Idle to Protection |
| 24 | alaErpRingMultipleRpl | bridge | ERP Ring<value>: Multiple Ring Protection Links detected |
| 25 | alaErpRingRemoved | bridge | ERP Ring <value>: Ring dynamically removed |
| 26 | alaDhcpClientAddressAddTrap | ip-helper | DHCP Client : New IP address <value> assigned |
| 27 | alaDhcpClientAddressExpiryTrap | ip-helper | DHCP Client: IP address <value> lease expired |
| 28 | alaDhcpClientAddressModifyTrap | ip-helper | DHCP Client: Unable to obtain IP address <value> - assigned new IP address <value> |
| 29 | dot1agCfmFaultAlarm | bridge | OAM: CFM Fault Alarm <info> detected |
| 30 | virtualChassisStatusChange | vcm | Virtual Chassis: Chassis chassis-id Status changed to <info> |
| 31 | virtualChassisRoleChange | vcm | Virtual Chassis: Chassis chassis-id Role changed to <info> |
| 32 | virtualChassisVflStatusChange | vcm | Virtual Chassis: VFL Link c/s/p Status changed to <info> |
| 33 | virtualChassisVflMemberPortJoinFail | vcm | Virtual Chassis: Member Port c/s/p unable to join VFL Link |
| 34 | portViolationNotificationTrap | interface | Port c/s/p  violation cleared - reason <info> |

| No | Trap Name | Family | User Readable Log Description |
|---|---|---|---|
| 35 | chassisTrapsDuplicateMacCleared | chassis | The old Master chassis has rejoined the virtual chassis as a slave. There is no longer a possiblity of duplicate MAC address in the network. |
| 36 | virtualChassisUpgradeComplete | vcm | Virtual Chassis: Software Upgrade Complete - status timeout/successful |
| 37 | virtualChassisVflSpeedTypeChange | vcm | Virtual Chassis: VFL Link c/s/p Speed type changed to <info> |
| 38 | alaIPv6NeighborLimitExceeded | ip | IPv6 global neighbor cache limit <value> exceeded |
| 39 | alaIPv6NeighborVRFLimitExceeded | ip | IPv6 VRF <value> neighbor cache limit <info> exceeded |
| 40 | alaIPv6InterfaceNeighborLimitExceed | ip | IPv6 interface <value> neighbor cache limit <value> exceeded |
| 41 | alaPethPwrSupplyConflictTrap | module | Power Supply Type <info> conflict detected |
| 42 | alaPethPwrSupplyNotSupported Trap | module | Power Supply Type <info> not supported |
| 43 | chasTrapsBPSLessAllocSysPwr | chassis | Insufficient system power given by backup power supply <info> |
| 44 | chasTrapsBPSStateChange | chassis | Backup power supply <info> insertion/ removal detected |
| 45 | chasTrapsNiBPSFETStateChange | chassis | Backup power supply <info> FET state changed to <info> |
| 46 | alaVCSPProtectionTrap | vcm | Virtual Chassis: Chassis id <c> entering Split VC Protection state |
| 47 | alaVCSPRecoveryTrap | vcm | Virtual Chassis: Chassis id <c> entering Split VC Active state |
| 48 | pethPsePortOnOffNotification | module | PSE port c/s/p is (not) delivering power |
| 49 | pethMainPowerUsageOnNotification | module | PSE Power Usage Threshold indication is on |
| 50 | pethMainPowerUsageOffNotification | module | PSE Power Usage Threshold indication is off |
| 51 | chasTrapsBPSFwUpgradeAlert | chassis | Alert - Backup power supply firmware upgrade is required |
| 52 | pethMainPowerUsageNiFailNotification | module | NI c/s cannot be powered |
| 53 | esmStormThresholdViolationStatus | interface | Port c/s/p Storm Threshold violation - ingress traffic exceeds configured value <value> |
| 54 | alaSTPLoopGuardError | stp | STP Port c/s/p entering Loop inconsistent error state |
| 55 | alaSTPLoopGuardRecovery | stp | STP Port c/s/p exiting Loop inconsistent error state |
| 56 | alaLldpTrustViolation | aip | Port c/s/p LLDP Trust violation detected |
| 57 | alaLicenseManagerDemoDayAlert | licensing | Demo License will expire on date: mm/dd/ yyyy |

| No | Trap Name | Family | User Readable Log Description |
|---|---|---|---|
| 58 | alaAaaUserCreation | aaa | New user <info> has been created |
| 59 | alaAaaUserDeletion | aaa | User <info> has been deleted |
| 60 | alaAaaUserModification | aaa | User <info> has been modified |
| 61 | systemSwlogFailureTraps | system | SWLOG failed to store log messages on flash ; SWLOG failed to send log messages to external syslog server |
| 62 | pethPseMainTemperatureUpAlert | module | Temperature Up Threshold Alert Power budget reduced |
| 63 | pethPseMainTemperatureDownAlert | module | Temperature Up Threshold Alert Power budget re-configured |
| 64 | systemRebootSwlogFailureTrap | system | System will reboot due to SWLOG failure to send log messages to remote server, reason <info> |

# Index

## Symbols

**!!** command 5-6

## A

aaa authentication command 8-7, 8-8, 8-9, 9-4
aaa radius-server command 8-7
accounting
  for Authenticated Switch Access 8-11
application example
  Ethernet OAM 11-3
application examples
  applying configuration files 6-3
  Authenticated Switch Access 8-7
  CMM 4-3
  configuration file 6-2
  Emergency Restore 4-22, 4-24
  file management 3-15
  logging into the switch 2-3
  network administrator user accounts 7-5
  NTP 16-3
  Server Load Balancing 11-8, 12-29
  SNMP 10-3
  Trap Filters 10-4
  WebView 9-4
applying configuration files
  application examples 6-3
ASA
  *see* Authenticated Switch Access
ASA Configuration
  verify information about 8-20
Authenticated Switch Access 8-4
  accounting 8-11
  application examples 8-7
  management interfaces 8-9
authentication
  MD5 10-10
  SHA 10-10
  traps 10-15
Automatic Remote Configuration 14-5
  Bootup Configuration File 14-11
  Debug Configuration File 14-11
  Firmware upgrade Files 14-11
  Instruction File 14-11
  Script File 14-11
  Troubleshooting 14-20
Automatic Remote Configuration network components 14-6
  TFTP File Server 14-6

## B

banner
  login 2-15
  pre-login text 2-16
boot.cfg file 4-2

## C

**cd** command 3-7
certified directory 4-2
  copying to working directory 4-17
Chassis Management Module
  *see* CMM
chmod command 3-9
CLI 5-1, 12-1
  domains and families 7-14
  logging commands 5-7–5-8
CLI usage
  verify information about 5-10
CMM 4-1
  application examples 4-3
  boot.cfg file 4-2
  cancelling a reboot 4-11, 4-13, 4-16
  certified directory 4-2
  checking reboot status 4-11
  configuration files 4-2
  copying
    certified directory to working directory 4-17
    running configuration to working directory 4-12
  displaying current configuration 4-15, 4-19
  displaying switch files 4-15
  image files 4-2
  managing 4-10
  rebooting 4-10, 4-16
  rebooting from the working directory 4-13, 4-17
  running configuration 4-2
  scheduling a reboot 4-11, 4-16
  swapping primary for secondary 4-18
  synchronizing primary and secondary 4-17
  working directory 4-2
CMM Conditions
  verify information about 4-24
CMM scenarios 4-3
  lost running configuration 4-3
  rollback to previous software 4-6
  running configuration saved to working directory 4-5
  working directory saved to certified directory 4-5
Command Line Interface
  *see* CLI
commands
  domains and families 8-15
community strings 10-9
configuration apply command 6-2, 6-3
  for a specific timeperiod 6-4
configuration cancel command 6-6
configuration error-file limit command 6-6
configuration file
  application examples 6-2
configuration files 4-2, 5-2

errors 6-6
configuration snapshot all command 6-9
configuration syntax check 6-6
console port 2-4
copy flash-synchro command 4-17
copy working certified flash-synchro command 4-17

## D

date 3-17, 6-3
Daylight Savings Time
*see* DST
defaults
dynamic link aggregation 13-3, 15-3
login 2-2
NTP 16-1
SNMP 10-2
startup 7-3
switch security 8-2
user accounts 7-2
WebView 9-2
delete command 3-9
DES encryption 10-10
directories
certified 4-2
flash 3-6
managing 4-10
working 4-2
DNS resolver 2-18
Domain Name Server
*see* DNS resolver
DST 3-18
dynamic link aggregation
defaults 13-3, 15-3

## E

editor
vi 6-7
Emergency Restore
application examples 4-22, 4-24
encryption
DES 10-10
errors 6-6
Ethernet OAM
application example 11-3
exit command 3-14

## F

File Configuration
verify information about 6-11
file management
application examples 3-15
files
attributes 3-9
boot.cfg 4-2
configuration 4-2
image 4-2
names 6-8
permissions 3-9
snapshots 6-8
filters
traps 10-4
freespace command 3-10
fsck command 3-10
FTP client 3-13
ftp command 3-13
FTP server 3-12

## H

help 5-5
HTTP
web browser 2-5
http port command 9-3
http ssl command 9-3
https port command 9-3

## I

image files 4-2
**i**p domain-lookup command 2-18
ip domain-name command 2-18
ip name-server command 2-18

## K

keywords 5-4, 5-5

## L

lacp linkagg size command 13-19, 13-20, 15-26, 15-27
LDAP accounting servers
Authenticated Switch Access 8-11
LDAP servers
for switch security 8-4
logging into the switch
application examples 2-3
login
defaults 2-2
login banner 2-15
login settings
verify information about 2-20
**ls** command 3-5, 5-5

## M

Management Information Bases
*see* MIBs
MD5
authentication 10-10
memory 3-10
**mkdir** command 3-7

## N

network administrator user accounts
application examples 7-5
Network Management Station
*see* NMS

Network Time Protocol
*see* NTP
NMS 10-7
NTP 16-1
application examples 16-3
configuring 16-9
client 16-9
defaults 16-1
overview 16-5
stratum 16-6
using in a network 16-6
ntp broadcast command 16-9
ntp broadcast-delay command 16-9
NTP client
broadcast delay 16-9
broadcast mode 16-9
ntp client command 16-3, 16-9
NTP Configuration
verify information about 16-13
ntp key command 16-12
ntp key load command 16-12
NTP server
designating 16-10
minimum poll time 16-10
preferred server 16-11
version number 16-10
ntp server command 16-3, 16-10

## O

OSPF
specifications 2-19

## P

partition management 10-14
password command 7-7
passwords
expiration 7-10
global settings 7-6
user-configured 7-7
pre_banner.txt file 2-16
Prefix Recognition 5-6
primary CMM
swapping with the secondary 4-18
synchronizing with secondary 4-17
prompt 5-9
pwd command 3-6

## R

RADIUS accounting servers
Authenticated Switch Access 8-11
RADIUS servers
for switch security 8-4
RAM 4-2
reboot
cancelling 4-11, 4-13, 4-16
checking status 4-11
primary 4-10, 4-16
scheduling 4-11, 4-16
secondary 4-16
working directory 4-13, 4-17
reload cancel command 4-13
reload command 4-10, 4-11, 4-16
reload secondary command 4-16
reload working command 4-13
rmdir command 3-8
running configuration 4-2
copying to working directory 4-12

## S

screen
display 5-9
prompt 5-9
secondary CMM
swapping with the primary 4-18
synchronizing with primary 4-17
Secure Shell 2-4, 2-10
algorithms 2-12
key exchange 2-12
security
SNMP 10-9
Server Load Balancing
application examples 11-8, 12-29
session banner command 2-15
session login-attempt command 2-17
session login-timeout command 2-17
session prompt command 5-9
session timeout command 2-17
sftp command 3-14
SHA
authentication 10-10
show command-log command 5-8
show command-log status command 5-8
show configuration status command 6-2, 6-6
show history command 5-6
show ip helper command 6-2
show microcode command 4-15, 5-6
show ntp client command 16-4
show ntp client server-list command 16-3
show ntp server status command 16-3
show reload command 4-11
show running-directory command 4-15, 4-19
show snmp community map command 10-9
show snmp mib family command 10-16
show snmp station command 10-3
show snmp trap replay command 10-15
show user command 7-5, 10-4, 10-10
show vlan svlan command 13-38, 15-29
show vlan svlan port-config command 13-38
snapshots 6-8, 6-11
SNMP
access for user accounts 7-16
agent 10-6
application examples 10-3
defaults 10-2
management station 10-7

manager 10-6
security 10-9, 10-12
traps table B-2, B-62, B-65
versions 10-7
snmp community map mode command 7-15
SNMP configuration
verify information about 10-17
snmp security command 7-15, 10-12
snmp trap filter command 10-5
software rollback
configuration scenarios 4-3
specifications
OSPF 2-19
switch security 8-2
ssh command 2-14
SSL
*see* Secure Socket Layer
startup
defaults 7-3
switch
rebooting 4-10, 4-16
switch security
defaults 8-2
specifications 8-2
syntax 5-3, 12-15, 12-20, 12-25
syntax checking 5-6
System Clock 3-17
system date command 3-17, 3-22
system time command 3-17
system timezone command 3-17


## T


takeover command 4-18
Telnet 2-4, 2-9
telnet command 2-9
time 3-17, 6-3
time zone 3-17
timed sessions 6-3
cancelling 6-6
future timed session 6-4
Trap Filters
application examples 10-4
Traps 10-14
traps
authentication 10-15
families 10-14
filters 10-14
management 10-15
tty command 5-9


## U


user accounts
defaults 7-2
for switch access 7-3
saving settings 7-6
SNMP access 7-16
user command 7-11, 8-7, 10-4
creating a user 7-7
user configuration
verify information about 7-17
user database
switch management 8-5
user password-expiration command 7-10
user password-size min command 7-9
users
*see* user accounts
UTC 16-1


## V


verbose mode 6-7
vi command 3-8


## W


WebView 9-1, 11-1
application examples 9-4
browser setup 9-2
CLI commands 9-3
defaults 9-2
disabling 9-3
enabling 9-3
Secure Socket Layer 9-3
who command 7-17
whoami command 7-18
working directory 4-2