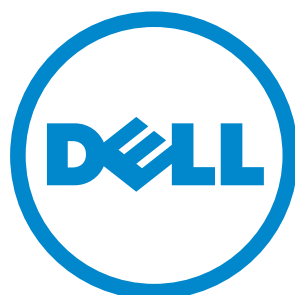





Open Automation Guide

Configuration and Command Line Reference

March 2014



Notes, Cautions, and Warnings

-  NOTE: A NOTE indicates important information that helps you make better use of your computer.
-  CAUTION: A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.
-  WARNING: A WARNING indicates a potential for property damage, personal injury, or death.

Information in this publication is subject to change without notice.

Copyright © 2014 Dell Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. Dell and the Dell logo are trademarks of Dell Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

March 2014

1	About this Guide	7
	Objectives	7
	Audience	7
	Supported Platforms and Required Dell Networking OS Versions	7
	Conventions	8
	Information Symbols	8
	Related Documents	9
2	Open Automation Framework	11
	Bare Metal Provisioning	12
	Smart Scripting	13
	Virtual Server Networking	13
	REST API	14
	Web Server with HTTP Support	14
3	Bare Metal Provisioning	15
	Introduction	15
	How it Works	15
	Prerequisites	16
	Standard Upgrades with BMP	16
	BMP Process Overview	17
	BMP Operations	17
	Configuring BMP	18
	Reload Modes	18
	BMP Mode	18
	Normal Mode	19
	BMP Commands and Examples	19
	System Boot and Set-Up Behavior in BMP Mode	19
	BMP Mode: Boot and Set-UP Behavior	22
	Reload Without a DHCP Server Offer	22
	Reload with a DHCP Server Offer Without a Dell Networking OS Offer	22
	Reload with a DHCP Server Offer and no Configuration File	23
	Reload with a DHCP Server Offer Without a DNS Server	24
	Software Upgrade Using BMP	26
	Applying Configurations Using BMP Scripts	26
	Preconfiguration Scripts	26
	Post Configuration Scripts	27
	Auto-Execution Scripts	27
	Preconfiguration Process for Scripts	28
	Running Scripts	29
	Using the Post Configuration Script (BMP Mode Only)	29
	Reload Using the Auto-Execution Script (Normal Mode Only)	29
	Script Examples	30
	Auto-Execution Script - Normal Mode	30
	Preconfiguration Script - BMP Mode	33

Post Configuration Script - BMP Mode	34
BMP Operations on Servers Overview	36
DHCP Server	36
DHCP Server Settings	36
DHCP Server IP Blacklist	37
MAC-Based Configuration	38
MAC-Based IP Address Assignment	38
Class-Based Configuration	39
File Server Settings	40
Domain Name Server Settings	40
4 Bare Metal Provisioning CLI	41
Overview	41
Commands	42
5 Smart Scripting	49
Overview	49
Downloading the Smart Scripting Package	51
Installing Smart Scripting	51
Displaying Installed Packages	53
Uninstalling Smart Scripting	53
Limits on System Usage	53
Supported UNIX Utilities	54
Smart Utils	56
Creating a User Name and Password for Smart Scripting	56
Logging in to a NetBSD UNIX Shell	57
Downloading Scripts to a Switch	57
Setting a Search Path for Scripts	58
Scheduling Time / Event-based Scripts	58
Triggering a Script to Run	58
SQLite	61
NET SNMP Client	62
Managing Executed Scripts	62
Viewing Script Information	64
Running a Script from the UNIX Shell	64
Using the PERL API	65
Running a PERL API Script	68
Using the Python API	68
Creating a Python API Script	69
Running a Python API Script	71
Using UNIX Shell Scripting	72
Creating a UNIX API Script	72
Running a UNIX API Script	74

Running Scripts with User Privileges	75
6 Smart Scripting CLI	77
Overview	77
Commands	77
7 Virtual Server Networking	95
Overview	96
Hypervisor Modes	97
VSN Persistency	97
VLAN configuration	97
Management VLAN	97
Data VLANS	98
Hypervisor-unaware VLANs	98
Installing VSN	98
Enabling VSN in a Hypervisor Session	100
Discovery	102
Connectivity	102
Running VSN Scripts	103
Stopping a Hypervisor Session	104
Disabling a Session	104
Removing a Session	104
Uninstalling VSN	105
Viewing VSN information	105
8 Virtual Server Networking CLI	109
Overview	109
Commands	109
9 REST API	121
HTTP and HTTPS	121
XML	121
Important Points to Remember	122
REST Authentication	122
POST and GET Request Examples	123
HTTP Status Error Codes	124
REST API - Protocol Data Unit (PDU) Structure	125
Configurations	126
TenGigabitEthernet	126
FortyGigabitEthernet	128
Port-Channel	130
VLAN	131

Static Route	132
BGP	133
Operational	134
TenGigabitEthernet	134
FortyGigabitEthernet	136
Port-channel	137
VLAN	139
IP Statistics	140
IP Route	141
BGP	142
BGP Neighbors	142
System Alarm	145
System Inventory	145
System Version	146
VLAN	146
System	147
BGP MIB	148
Forwarding Plane Statistics	158
IETF Interfaces	161
IF MIB	161
10 REST API CLI	165
Overview	165
11 Web Server with HTTP Support	167
Web Server	167
12 Web Graphical User Interface	169

About this Guide

Objectives

This document describes the components and uses of the Open Automation Framework designed to run on the Dell Networking Operating System (OS), including:

- Bare Metal Provisioning (BMP)
- Smart Scripting
- Virtual Server Networking (VSN)
- Representational state transfer (REST) application programming interface (API)
- Web graphic user interface (GUI) and HTTP Server

Audience

This document is intended for data center managers and network administrators responsible for virtualization or system management. It assumes basic knowledge about virtualization technology and networking.



Note: Although this document contains information on protocols, it is not intended to provide complete information on protocol configuration and usage. For this information, refer to the document listed in [Related Documents](#) and the IETF Requests for Comment (RFCs).

Supported Platforms and Required Dell Networking OS Versions

The Open Automation release is supported on the following Dell Networking switches and minimum Dell Networking OS versions.

- S4810 switches require Dell Networking OS version 8.3.10.1 or later.

- S4820T switches require Dell Networking OS version 9.2(0.0) or later.
- Z9000 switches require Dell Networking OS version 9.0.0.0 or later. (SmartScripts and SmartUtil support only)
- S6000 switches require Dell Networking OS version 9.0.2.0 or later.
- MXL Switches require Dell Networking OS version 9.2(0.0) or later.

Conventions



This document uses the following conventions to describe command syntax:

Convention	Description
keyword	Keywords are shown in bold and should be entered in the CLI as listed.
<i>parameter</i>	Parameters are shown in italics and require a number or word to be entered in the CLI.
{X}	Keywords and parameters within braces are required entries and must be entered in the CLI.
[X]	Keywords and parameters within brackets are optional.
x y	Keywords and parameters separated by bar require you to choose one.

Information Symbols

Table 1-1 describes the symbols used in this document.

Table 1-1. Information Symbols

Symbol	Type	Description
	Note	Informs you about important operational information.
	Dell Networking OS Behavior	Informs you about an Dell Networking OS behavior. These behaviors are inherent to the Dell Networking system or Dell Networking OS feature and are non-configurable.
S4810 S4820T S6000 Z9000 MXL Switch	Platform-specific Feature	Informs you of the platform supporting the Open Automation features. For example, the S4810 and S4820T platforms support all Open Automation features. The Z9000 and S6000 platform supports the SmartScripts and SmartUtil features. The MXL platform supports Representational State Transfer Interface (REST API) and the SmartScripts features.
	Exception	A note associated with some other text on the page that is marked with an asterisk.

Related Documents

For more information about the Dell Networking switches discussed in this document, refer to the following documents:

- S4810
 - *Dell Networking OS Command Line Reference Guide for the S4810 System*
 - *Dell Networking OS Configuration Guide for the S4810 System*
 - *Installing the S4810 System*
- S4820T
 - *Dell Networking OS Command Line Reference Guide for the S4820T System*
 - *Dell Networking OS Configuration Guide for the S4820T System*
 - *Installing the S4820T System*
- Z9000
 - *Dell Networking OS Command Line Reference Guide for the Z9000 System*
 - *Dell Networking OS Configuration Guide for the Z9000 System*
 - *Installing the Z9000 System*
- MXL Switch
 - *Dell Networking OS Command Line Reference Guide for the MXL 10/40GbE Switch IO Module*
 - *Dell Networking OS Configuration Guide for the MXL 10/40GbE Switch IO Module*
 - *MXL 10/40GbE Switch IO Module Getting Started Guide*
- [Dell Networking OS Release Notes](#) for the platform and version you are using.

Open Automation Framework

[Open Automation Framework](#) is supported on the **S4810, S4820T, S6000, Z9000** and **MXL** platforms.

Dell Networking's Open Automation Framework is designed to provide an open, industry standards-based automation technology that simplifies the management of dynamic virtual data centers and reduces risk and overhead.

With the Open Automation Framework, resources in a virtualized data center are managed more flexibly and efficiently without requiring the manual reconfiguration of virtual switches (vSwitches), virtual machines (VMs) on network servers, and VM control software each time there is a change in the network. Automated provisioning of network resources during virtual machine migration ensures that connectivity and security policies are maintained.

Industry-standard scripting languages, such as Perl and Python, are used to automate the monitoring and management of network devices. Virtual resources can be quickly allocated to adapt to configuration changes. Failure of a network device is more quickly detected and resolved. As a result, network uptime increases.

Automated bare metal provisioning allows you to reduce operational overhead by automatically configuring Dell Networking switches, accelerating switch installation, and simplifying operating system upgrades.

Support for multiple, industry-standard hypervisors, virtual switches, and system management tools ensure that automated solutions work within an established data-center environment in which heterogeneous server, storage, and networking equipment inter-operate. In addition, Open Automation allows you to customize automated solutions for your current multi-vendor virtualization environment.

An onboard Web-based graphical user interface (GUI) provides a user-friendly way to monitor and manage a data center network. HTTP and HTTPS daemons run on supported switches to provide additional management capability, such as the REST application programming interface (API).

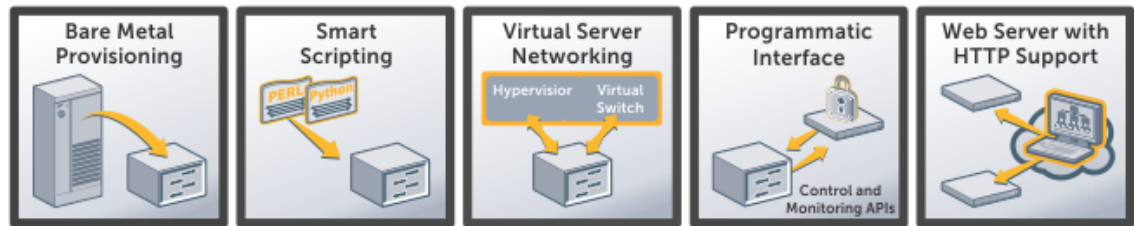
The Open Automation Framework consists of the following network management tools:

- Bare Metal Provisioning
- Smart Scripting
- Virtual Server Networking

- Representational State Transfer Application Programming Interface
- Web Server with HTTP Support

You can use these components together or independently to extend and add functionality to the Dell Networking OS without requiring updates to a Dell Networking OS release.

Figure 2-1. Open Automation Framework



Note: The *Open Automation Framework* is referred to as *Open Automation* in the rest of this document.

Bare Metal Provisioning

Bare Metal Provisioning (BMP) provides the following features:

- Automatic network switch configuration and automated configuration updates
- Enforced standard configurations
- Reduced installation time
- Simplified operating system upgrades

Automated bare metal provisioning reduces operational expenses, accelerates switch installation, simplifies upgrades and increases network availability by automatically configuring Dell Networking switches. BMP eliminates the need for a network administrator to manually configure a switch, resulting in faster installation, elimination of configuration errors, and enforcement of standard configurations.

With bare metal provisioning, after a switch is installed, the switch searches the network for a DHCP server. The DHCP server provides the switch with an IP address and the location of a file server, such as TFTP. The file server maintains a configuration file and an approved version of Dell Networking OS, the operating system for Dell Networking switches. The switch automatically configures itself by loading and installing an embedded Dell Networking OS image with the startup configuration file.

For more information on Bare Metal Provisioning, refer to the *Dell Networking OS Configuration Guide* for the S4810, S4820T, S6000 and Z9000 switches or to the *Configuration Guide for the MXL 10/40GbE Switch IO Module* for MXL switches.

Smart Scripting

Smart Scripting provides:

- Support for industry-standard languages, such as Perl and Python, avoiding the need to learn a new proprietary scripting language
- Customization of device monitoring and management to suit your network needs, including custom maintenance tasks, discovery programs, and event logging for faster problem resolution

Smart scripting increases network availability and manageability by allowing network administrators to deploy custom monitoring and management scripts on Dell Networking switches. Using custom scripts, network administrators can implement version control systems, automatically generate alerts, create custom logging tools and automate management of network devices. Any function that can be performed through the Dell Networking OS command-line interface (CLI) can be performed with smart scripting.

The scripting environment provided by Smart Scripting (Expect, Perl, Python, Tcl, UNIX and ZSH shell scripts) makes it easy for IT administrators to quickly develop scripts without having to learn a new scripting language.

Virtual Server Networking

Virtual Server Networking (VSN) provides:

- Automatic re-provisioning of VLANs when you migrate virtual machines (VMs).
- Support for multiple hypervisors, such as VMware and Citrix XenServer.

Virtual data centers require network infrastructure to be dynamic to ensure that network connectivity and QoS and security policies are maintained when VMs are migrated. VSN facilitates communication between Dell Networking switches and VM management software to automatically re-provision VMs and associated VLANs during virtual machine migration.

As a result, VSN greatly simplifies many of the tasks associated with virtualized computing environments. Network administrators can manage the network while server administrators manage the servers. No manual VLAN reconfiguration is required when you migrate VMs.

VSN software supports the following hypervisors:

- VMware vSphere 4.0/4.1/5.0
- Citrix XenServer 5.6/6.0

REST API

REST-API provides application programming interfaces (APIs) that allow Dell Networking OS switches to be managed by in-house or third-party system management tools.

- Common third-party management tool sets are supported as plug-ins to the Open Automation Framework, including Dell AIM, EMC Smarts Ionix, IBM Systems Director, HP Network Automation (NA), CA Spectrum Infrastructure Manager, and Oracle Enterprise Manager (OEM).
- Industry-standard management protocols are supported, such as SNMP (Get and Set) and Representational State Transfer (REST).
- User protocols are supported, such as CLI/CLI-script, XML (Get and Set) and Web-based commands.

REST-API greatly improves network manageability by allowing Dell Networking switches to be managed by third-party system management tools via standard program interfaces.

The programmatic management environment and set of interfaces communicate directly with third-party system management tools, avoiding the need for a dedicated network management tool. As a result, network management is simplified and the number of management tools is minimized.

Web Server with HTTP Support

The Open Automation Framework supports Web connectivity through its Web server with HTTP support.

- The Web Server consists of both HTTP and HTTPS daemons running on a switch.

Bare Metal Provisioning

Bare metal provisioning (BMP) is included as part of the Dell Networking Operating System (OS) image. BMP is supported on the **S4810**, **S4820T**, **S6000**, **Z9000** and **MXL** switch platforms.

Introduction

BMP improves operational efficiency to the system by automatically loading pre-defined configurations and Dell Networking OS images using standard protocols such as dynamic host configuration protocol (DHCP) and common file transfer mechanisms.

Bare metal provisioning:

- Reduces the time to install and configure the network device.
- Helps eliminate configuration errors and ensure consistent configurations.
- Functions on a single system or on multiple systems.
- Includes SNMP support.
- Includes support for pre- and post configuration scripts.

How it Works

With BMP, the system can retrieve a configuration file or a preconfiguration script indicated in the DHCP offer. Using the preconfiguration script, you can:

- Verify the integrity of the boot image downloaded from the DHCP offer.
- Decide what type of configurations you want to apply based on factors such as your network reachability, port status, and neighbor discovery.
- Monitor your CPU and memory utilization, your port traffic status, or perform link and topology checking with link layer discovery protocol (LLDP).
- Retrieve a configuration from a central repository.

If you disable BMP, auto-exec support is provided in the Normal mode. Using the auto-exec feature, you can apply script-based configurations at the start-up.

Prerequisites

Before you use BMP to auto-configure a supported Dell Networking switch, configure the following:

- External DHCP server (required) — a network device offering configuration parameters.
- File server (required) — a network device for storing and servicing files.
- Domain name server (DNS) (optional) — a server that associates domain names in the network with IP addresses.
- Relay agent (optional) — an intermediary network device that passes messages between the DHCP clients and the DHCP server when the server is not on the same subnet. It can also provide IP addresses for multiple subnets.

For more information, refer to [Domain Name Server Settings](#) and [File Server Settings](#).

Standard Upgrades with BMP

The standard upgrades performed with BMP provides a fallback mechanism that allows the switches to return to their previous state.

Switches normally have two partitions to store images. You can select which partition the boot up. BMP always downloads and saves the image in the inactive partition. When BMP reloads, it automatically activates this partition so the switch boots up with the newly downloaded image.

The downloaded configuration is copied to the running configuration and then applied to the switch.



Note: The downloaded configuration does not override the startup configuration.

Verify the operation and performance of the new image and the configuration on the switch before committing the changes. After the changes are deemed correct, commit them by performing the following steps:

1. Disable BMP using the **reload-type normal-reload** command.
2. Make the activating partition permanent using the **boot system** command.
3. If you did not enable auto-save, manually override the startup configuration file using the **write memory** command.



Note: If you enable auto-save, the switch automatically overrides the startup configuration.

You can revert to the previous state by not committing the changes, disabling BMP, activating the previous partition, and reloading the switch.

BMP Process Overview

When the system boots up in default BMP mode, the following items are requested:

1. Current (new) Dell Networking OS build image.
2. Configuration file or preconfiguration script (EXPECT, TCL, or Z-shell [ZSH] script).
3. A list of checksums for all these components.



Note: The configuration file maintains normal BMP functionality when a preconfiguration script is not sent.

BMP Operations

BMP is supported on the user ports and management ports of a switch.

If your network has virtual link trunking (VLT) enabled on aggregator switches and you are configuring the top-of-rack (ToR) switch to load BMP, configure the aggregator switches with the **lACP ungroup member-independent vlt** command if the DHCP and file servers are reachable via the interface configured as part of VLT link aggregation group (LAG).

BMP eases configuration in the following ways:

- Switch access is allowed through all ports (management and user ports) with or without DHCP-based dynamic IP address configuration of a switch.
- Booting up in Layer 3 mode with interfaces already in No Shutdown mode and basic protocols enabled to protect the system and the network.
- To access the configuration file or a preconfiguration script, use the DHCP offer.
- Download and execute scripts before configurations are applied allowing preconfiguration checks on the switch.

Configuring BMP

BMP supports two types of reload modes: BMP mode and Normal mode.

Reload Modes

This section describes the following Reload modes:

- [BMP Mode](#)
- [Normal Mode](#)
- [DHCP Server Settings](#)
- [Dell Networking OS Image Retrieval](#)

BMP mode is the default boot mode configured for a new system arriving from the Dell factory. This mode obtains the Dell Networking OS image and configuration file from a network source (DHCP and file servers).

To boot the switch up with the management port in No Shutdown mode, use Normal mode. If the management IP address is present in the start-up configuration file, it is assigned. If the management IP address is not present in the start-up configuration file, no IP address is assigned to the management interface.

BMP Mode

To auto-configure a switch, before you use BMP mode, first configure a DHCP, DNS, and file server in the network.

To facilitate configuration of the switch on a new factory-loaded switch, the switch boots up in default BMP mode. You can reconfigure a switch to reload between BMP and Normal mode.

- BMP mode (default) — the switch automatically configures all ports (management and user ports) as Layer 3 physical ports and acts as a DHCP client on the ports for a user-configured time (DHCP timeout). Set BMP mode using the **reload-type bmp** command.
- Normal mode — the switch loads the Dell Networking OS image and startup configuration file stored in the local Flash. New configurations require that you manually configure the Management IP and Management Interface. Set Normal mode using the **reload-type normal-reload** command.



Note: To apply the startup configuration, cancel the default BMP setup using the **stop bmp** command from the console. To disable BMP for the next reload, use the **reload-type normal-reload** command.

Normal Mode

When reloaded in Normal mode, the switch boots up with the management port in No Shutdown mode.

If the management IP address is present in the start-up configuration file, it is assigned. If the management IP address is not present in the start-up configuration file, no IP address is assigned to the management interface. You can connect to the management port with an IP address on the same network and log in to the system through a telnet or SSH session.

To configure a switch to reload using Normal mode, use the following step:

- The switch reloads in Normal mode using the Dell Networking OS image and startup configuration file stored in the local Flash.

```
GLOBAL CONFIGURATION mode
  reload-type normal-reload
```

DHCP Configuration

Normal mode does not require a separate DHCP server configuration.

Dell Networking OS Image Retrieval

The Dell Networking OS image is loaded from the local Flash.

BMP Commands and Examples

You can configure BMP on supported switches using a series of commands. To enable BMP mode on switches, and to apply configuration or run scripts using BMP commands, refer the *Bare Metal Provisioning CLI* section.

System Boot and Set-Up Behavior in BMP Mode

The following steps describe the system boot up process:

1. The system begins the boot up process in BMP mode (the default mode).
2. The system sends DHCP discover on all the interface up ports.

```
00:02:14: %STKUNIT0-M:CP %BMP-5-BMP_DISCOVER: DHCP DISCOVER
sent on Ma 0/0.
```

3. The IP address, boot image filename, and configuration filename are reserved for the system and provided in the DHCP reply (one-file read method). The system receives its IP address, subnet mask, DHCP server IP, TFTP server address, DNS server IP, bootfile name, and configuration filename from the DHCP server.
4. If a DHCP offer has neither an image path or configuration file path, it is considered an invalid BMP DHCP offer and is ignored. The first DHCP offers the following to choose:
 - IP address
 - Dell_Networking OS image
 - Configuration file or preconfiguration script
 - IP address and Dell Networking OS image or
 - IP address and configuration file or preconfiguration script
5. The DHCP OFFER is selected.
All other ports except the port on which the offer was received and selected are set to Shutdown mode.

```
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_DHCP_OFFER: DHCP OFFER
received on Te 0/21 IS SELECTED.
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_REQUEST: DHCP REQUEST sent
on Te 0/21.
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_ACK: DHCP ACK received on
Te 0/21.
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: DHCP acquired
IP 13.4.4.44 mask 255.255.255.0 server IP 13.4.4.1.
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: DHCP tftp IP
NIL sname NIL dns IP NIL router IP NIL.
00:02:27: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: DHCP image file
tftp://13.4.4.1/ftos-img-s4810.
00:02:27: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: DHCP config
file scp://anvltest:force10@13.4.4.1/tftpboot/basic-l3.
00:02:27: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: stacking info
NIL.
```

6. The system sends a message to the server to retrieve the named configuration file or preconfiguration script and/or boot file from the base directory of the server.
 - a If you use the optional **bootfile-name** command, the filename can be 256 bytes. If a filename field is specified in the DHCP offer, the filename can be 128 bytes. The name can be a fully qualified URL or it can be a filename only.
 - b When an Dell Networking OS build image is found, the system compares that build image to the version currently loaded on the chassis.

- If there is a mismatch between the build images, the system upgrades to the downloaded version and reloads.

```
00:02:55: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Downloaded Image Major Version : 1
00:02:55: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Downloaded Image Minor Version : 0
00:02:56: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Downloaded Image Main Version : 0
00:02:56: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Downloaded Image Patch Version : 1216
00:02:56: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Flash A Image Major Version : 9
00:02:56: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Flash A Image Minor Version : 3
00:02:56: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Flash A Image Main Version : 0
00:02:56: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Flash A Image Patch Version : 57
00:02:56: %STKUNIT0-M:CP %BMP-2-BMP_DOWNLOAD_START: The Dell
Networking OS image download has started.
00:03:29: %STKUNIT0-M:CP %BMP-5-BMP_DOWNLOAD: The Dell
Networking OS image download is successful.
00:03:31: %STKUNIT0-M:CP %BMP-5-BMP_MD5_VALIDATE_SUCCESS: The
Dell Networking OS image MD5 Checksum Validation Successful.
```

Erasing Sseries Primary Image, please wait

- If the versions match, the system downloads the configuration file or preconfiguration script.

```
00:03:07: %STKUNIT0-M:CP %BMP-2-BMP_DOWNLOAD_START: The config
file download has started.
00:03:19: %STKUNIT0-M:CP %BMP-5-BMP_DWNLD_FILE_IS_SCRIPT_FILE:
The downloaded file is a script file.
00:03:19: %STKUNIT0-M:CP
%BMP-5-BMP_DWNLD_CONFIG_SCRIPT_SUCCESS: The config/script file
download is successful.
00:03:21: %STKUNIT0-M:CP %BMP-5-BMP_PRE_CONFIG_SCRIPT_BEGIN:
The Pre-Config Script has started to Execute.
.
.
00:04:30: %STKUNIT0-M:CP %BMP-5-BMP_PRE_CONFIG_SCRIPT_END:
Pre-Config script completed with return status 0.
```

- c If you download the configuration file or preconfiguration script from the server, any saved startup configuration on the Flash is ignored. If you do not download the configuration file or preconfiguration script from the server, the startup configuration file on the Flash is loaded as in Normal reload.

7. When the Dell Networking OS build image and configuration file or preconfiguration script are downloaded, the IP address is released.

```
00:03:32: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE: DHCP RELEASE sent
on Te 0/21.
```

8. The system applies the configuration. The system is now up and running and is managed as usual.

BMP Mode: Boot and Set-UP Behavior

When a switch that is configured to reload in BMP mode, one of the following scenarios may occur.

- [Reload Without a DHCP Server Offer](#)
- [Reload with a DHCP Server Offer Without a Dell Networking OS Offer](#)
- [Reload with a DHCP Server Offer and no Configuration File](#)
- [Reload with a DHCP Server Offer Without a DNS Server](#)
- [Preconfiguration Scripts](#)
- [Post Configuration Scripts](#)
- [Auto-Execution Scripts](#)

Reload Without a DHCP Server Offer

A switch configured to reload in BMP mode. If the DHCP server cannot be reached, the system keeps on sending DISCOVER messages.

```
00:03:56: %STKUNIT0-M:CP %BMP-5-BMP_DISCOVER: DHCP DISCOVER
sent on Ma 0/0.
00:03:56: %STKUNIT0-M:CP %BMP-5-BMP_DISCOVER: DHCP DISCOVER
sent on Te 0/21.
00:03:55: %STKUNIT0-M:CP %BMP-5-BMP_DISCOVER: DHCP DISCOVER
sent on Te 0/8.
00:03:41: %STKUNIT0-M:CP %BMP-5-BMP_DISCOVER: DHCP DISCOVER
sent on Ma 0/0.
00:03:40: %STKUNIT0-M:CP %BMP-5-BMP_DISCOVER: DHCP DISCOVER
sent on Te 0/21.
00:03:40: %STKUNIT0-M:CP %BMP-5-BMP_DISCOVER: DHCP DISCOVER
sent on Te 0/8.
00:03:25: %STKUNIT0-M:CP %BMP-5-BMP_DISCOVER: DHCP DISCOVER
sent on Ma 0/0.
00:03:25: %STKUNIT0-M:CP %BMP-5-BMP_DISCOVER: DHCP DISCOVER
sent on Te 0/21.
00:03:25: %STKUNIT0-M:CP %BMP-5-BMP_DISCOVER: DHCP DISCOVER
sent on Te 0/8.
```

Reload with a DHCP Server Offer Without a Dell Networking OS Offer

Switches configured to reload in BMP mode that reach a DHCP server but do not locate a downloadable Dell Networking OS image file on the server will attempt to download the configuration file.

1. The system boots up with the BMP application.

```
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_DHCP_OFFER: DHCP OFFER
received on Te 0/21 IS SELECTED.
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_REQUEST: DHCP REQUEST sent
on Te 0/21.
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_ACK: DHCP ACK received on
Te 0/21.
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: DHCP acquired
IP 13.4.4.44 mask 255.255.255.0 server IP 13.4.4.1.
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: DHCP tftp IP
NIL sname NIL dns IP NIL router IP NIL.
00:02:27: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: DHCP image file
NIL.
00:02:27: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: DHCP config
file scp://anvltest:force10@13.4.4.1//tftpboot/basic-l3.
00:02:27: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: stacking info
NIL
```

2. The system downloads the customer.conf configuration file from the file-server address if you enabled **config-scr-download** command.
3. If the configuration download is successful, the following logs display:

```
00:01:22: %STKUNIT0-M:CP %BMP-2-BMP_DOWNLOAD_START: The config
file download has started.
00:01:23: %STKUNIT0-M:CP %BMP-5-BMP_DWNLD_FILE_IS_CONFIG_FILE:
The downloaded file is a configuration file.
00:01:23: %STKUNIT0-M:CP
%BMP-5-BMP_DWNLD_CONFIG_SCRIPT_SUCCESS: The config/script file
download is successful.
00:01:24: %STKUNIT0-M:CP %BMP-5-DOWNLOAD_INFO: /tftpboot/
signal.py config file successfully downloaded.
00:01:24: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE: DHCP RELEASE sent
on Ma 0/0.
00:01:24: %STKUNIT0-M:CP %BMP-5-BMP_DOWNLOAD: The config file
download is successful.
00:01:24: %STKUNIT0-M:CP %BMP-5-CFG_APPLY: The downloaded
config from dhcp server is being applied.
00:01:24: %STKUNIT0-M:CP %IFMGR-5-ASTATE_DN: Changed interface
Admin state to down: Ma 0/0
00:01:24: %STKUNIT0-M:CP %IFMGR-5-OSTATE_DN: Changed interface
state to down: Ma 0/0
00:01:25: %STKUNIT0-M:CP %SYS-5-CONFIG_LOAD: Loading
configuration file
```

Reload with a DHCP Server Offer and no Configuration File

If a switch is configured to reload in BMP mode reaches a DHCP server but cannot retrieve a configuration file, the switch looks for a configuration file on the file server only if you enabled **config-scr-download** command.

1. The system boots up with the BMP application.

2. The system receives a DHCP offer from a DHCP server with the following parameters.


```
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_DHCP_OFFER: DHCP OFFER
received on Te 0/21 IS SELECTED.
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_REQUEST: DHCP REQUEST sent
on Te 0/21.
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_ACK: DHCP ACK received on
Te 0/21.
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: DHCP acquired
IP 13.4.4.44 mask 255.255.255.0 server IP 13.4.4.1.
00:02:26: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: DHCP tftp IP
NIL sname NIL dns IP NIL router IP NIL.
00:02:27: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: DHCP image file
tftp://13.4.4.1/ftos-img-s4810.
00:02:27: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: DHCP config
file NIL.
00:02:27: %STKUNIT0-M:CP %BMP-5-BMP_BOOT_OFFER: stacking info
NIL.
```
3. The system downloads the build image from the file server.
4. The system compares the current local build image to the downloaded build image as follows.
 - a. If the build image versions match, the system does not try to load any image and comes up with the Dell prompt.


```
00:02:55: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Downloaded Image Major Version : 1
00:02:55: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Downloaded Image Minor Version : 0
00:02:56: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Downloaded Image Main Version : 0
00:02:56: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Downloaded Image Patch Version : 1216
00:02:56: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Flash A Image Major Version : 1
00:02:56: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Flash A Image Minor Version : 0
00:02:56: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Flash A Image Main Version : 0
00:02:56: %STKUNIT0-M:CP %BMP-5-BMP_RELEASE_HEADER_INFO:
Flash A Image Patch Version : 1216
```
 - b. If the build images versions are different, the system stores the downloaded build image in the local Flash and loads the build image from the Flash. This process is repeated until the build image versions match.

The system looks for the configuration file on the file server. If the system does not find a configuration file, the download fails. If the build image processing is successful, BMP stops. If the build image download also fails, the current offer is treated as invalid and BMP tries to send DISCOVER messages.

Reload with a DHCP Server Offer Without a DNS Server

Reload Dell Networking OS on a switch using a preconfiguration script.

To reload Dell Networking OS on a switch using a preconfiguration script, the following must be true.

- BMP must be enabled.
- The download of the script is from an external server. The location is specified in the DHCP offer.

- The first line of the script must contain one of the following:


```
#!/usr/bin/expect
#!/usr/bin/tclsh
#!/usr/bin/zsh
```
- After the first line, but before the start of the script, the script must contain the signature `#/DELL-NETWORKING`.
- The preconfiguration script is downloaded instead of the configuration file.
- The preconfiguration script is run before applying the start-up configuration file.
- The preconfiguration script has the ability to use Dell Networking OS CLI commands using the **f10do** utility.
- When the preconfiguration script completes, the start up configuration file is automatically applied.

If a preconfiguration script file download is successful, the following SYSLOGs display.

```
00:03:07: %STKUNIT0-M:CP %BMP-2-BMP_DOWNLOAD_START: The config
file download has started.
00:03:19: %STKUNIT0-M:CP %BMP-5-BMP_DWNLD_FILE_IS_SCRIPT_FILE:
The downloaded file is a script file.
00:03:19: %STKUNIT0-M:CP
%BMP-5-BMP_DWNLD_CONFIG_SCRIPT_SUCCESS: The config/script file
download is successful.
00:03:21: %STKUNIT0-M:CP %BMP-5-BMP_PRE_CONFIG_SCRIPT_BEGIN:
The Pre-Config Script has started to Execute.
```

After the preconfiguration script successfully executes, the startup configuration file loads.

```
00:04:30: %STKUNIT0-M:CP %BMP-5-BMP_PRE_CONFIG_SCRIPT_END:
Pre-Config script completed with return status 0.
00:03:03: %STKUNIT0-M:CP %IFMGR-5-ASTATE_DN: Changed interface
Admin state to down: Ma 0/0.
00:03:03: %STKUNIT0-M:CP %IFMGR-5-OSTATE_DN: Changed interface
state to down: Ma 0/0.
00:03:03: %STKUNIT0-M:CP %SYS-5-CONFIG_LOAD: Loading
configuration file.
00:03:04: %STKUNIT0-M:CP %IFMGR-5-ASTATE_UP: Changed interface
Admin state to up: Ma 0/0.
```

If a post configuration script is present in the applied configuration file, it executes and the following logs display:

```
00:04:22: %STKUNIT0-M:CP %BMP-5-BMP_POST_CONFIG_SCRIPT_BEGIN:
The Post-Config Script has started to Execute
.
.
00:05:15: %STKUNIT0-M:CP %BMP-5-BMP_POST_CONFIG_SCRIPT_END:
Post-Config script completed with return status 0.
```

If a post configuration script is not present in the applied configuration file, the following logs display:

```
00:03:03: %STKUNIT0-M:CP %SYS-5-CONFIG_LOAD: Loading
configuration file.
00:03:04: %STKUNIT0-M:CP %IFMGR-5-ASTATE_UP: Changed interface
Admin state to up: Ma 0/0.
00:03:04: %STKUNIT0-M:CP %BMP-5-BMP_POST_SCRIPT_NOT_PRESENT:
The Post-Config Script is not present.
```

Software Upgrade Using BMP

BMP has the capability that simplifies and eases configuration by allowing you to boot images and run configurations that are specified in a DHCP server, and also automatically downloading files from a file server and applied by the switch. In a network topology with a large number of devices, you can employ BMP to effectively, administer and maintain the devices and upgrade them in a robust, streamlined way. This method of upgrade and deployment allows judicious management of large numbers of switches.

You can configure an Auto-Configuration mode using the **reload-type bmp** command. You can reload the switch in configuration mode using the **reload** command.

Applying Configurations Using BMP Scripts

With Dell Networking OS version 9.1(0.0) or later, the system supports a scripting environment when a BMP or Normal reload occurs. BMP uses pre- and post configuration scripts, while a Normal reload uses an auto-execution script.

Preconfiguration Scripts

In BMP, the Dell Networking OS accesses the image, then the configuration file or preconfiguration script from the DHCP offer. Use preconfiguration scripts to:

- verify the integrity of the Dell Networking OS image downloaded from the DHCP offer.
- dynamically decide what types of configurations to apply to your system based on various factors such as network reachability, port status, neighbor discovery.
- use LLDP to monitor and generate reports periodically for CPU and memory utilization, port traffic status, and perform link and topology checking.

You can provide a preconfiguration script in DHCP option 209 to either configure the switch or download a configuration file. The script can download a configuration file or apply CLI commands. In the downloaded configuration, post configuration script CLIs are executed if present.

You can set the system to retry downloading a configuration up to six times or to automatically save the downloaded configuration or script on the switch.

After the configuration is applied, it can trigger a post configuration script to ensure the configurations and switch functions are correct. This post configuration script can only be triggered when the preconfiguration script is run.

Define configuration parameters on the DHCP server for each chassis based on the chassis MAC Address or Vendor-Class-Identifier in DHCP offer 60 or combination of both.

The system supports preconfiguration scripts in EXPECT, TCLSH, and ZSH.



Note: Dell Networking recommends adding the following **f10do** wrapper function at the beginning of TCLSH and EXPECT scripts to display a properly formatted output string:

```
# Execute F10do and return the output string
proc ExecF10Do {cmd_str} {
  set str [exec f10do "$cmd_str"]
  set tmp_str [string map {\n \r\n} $str ]
  return $tmp_str
}
...
set out_str [ExecF10Do "show version"]
puts $out_str
...
```

Post Configuration Scripts

In BMP mode, after the preconfiguration script completes and the configuration loads, you can run a post configuration script if one is present in the configuration file.

To check the status of configured ports or protocols, use the post configuration script. To set the host name of the system or perform additional configuration settings, use the post configuration script. The system supports post configuration scripts in EXPECT, TCLSH, and ZSH. If you installed the SmartScripts package in the Dell switches, the system also supports post configuration scripts in PERL and Python.

Auto-Execution Scripts

An auto-execution (auto-exec) script is the same as a preconfiguration script except that it is executed on every reboot in Normal mode.

Store scripts in a `flash://autoexec` file. Auto-exec scripts are independent of BMP.

The auto-exec script only executes when:

- BMP is disabled.
- The script is stored in a `flash://autoexec` file.
- Use the **reload-type normal-reload** command before you reload the system.

If the auto-exec script fails, the system generates a message indicating the failure and does not load the configuration file. Before continuing the upgrade, correct the error in the script.

Preconfiguration Process for Scripts

To preconfigure scripts, follow these steps:

1. Decide what information you want to preconfigure; for example, request username and password information. Verify the integrity of the boot image downloaded from the DHCP offer and apply configuration types.
2. Create a preconfiguration script in EXPECT, TCL or ZSH. The script cannot be binary.
3. Store the script on the DHCP server in the DHCP offer.
4. Change the reload-type to BMP and reload the switch. The system boots in BMP mode.
5. The system receives an IP address via the DHCP server which it uses to get an Dell Networking OS image to boot, a configuration file (if supplied), and a preconfiguration script.
6. The system runs the preconfiguration script.
 - The default timer on the script is 10 minutes. The maximum amount of time the script can run is one hour.
7. The preconfiguration script can access Dell Networking OS CLI commands through the support of the **f10do** utility. The **f10do** utility has no pagination, is always set to “terminal length 0”, and has Dell Networking OS CLI privilege 15 enabled. It works in the following modes:
 - Continuous mode (**f10do** command) — use to retain the Dell Networking OS context.
 - Reset mode (**f10do-r** command) — use to reset the CLI command to its original context.
8. To execute CLI commands during boot time, the system uses an utility called **f10do**.

Running Scripts

Using the Post Configuration Script (BMP Mode Only)

To reload Dell Networking OS on a switch, ensure the following:

- BMP is enabled.
- You can write the post configuration script in EXPECT, TCLSH, or ZSH. If you already installed the SmartScripts package, you can also write the post configuration script in PERL or Python.
- No restraints are required for the post configuration script, for example, the signature `#/DELL-NETWORKING` that is required for the preconfiguration script.
- Configure the post configuration script by using the **script post-config** command.
- Execute the post configuration script after the start-up configuration process required by BMP has been applied.
- The post configuration script has the ability to use configuration Dell Networking OS CLI commands using the utility name **F10do**.

Reload Using the Auto-Execution Script (Normal Mode Only)

To use the auto-execution script, the following conditions must be true:

- BMP is disabled.
- Always store the auto-execution script in `flash://auto-exec`.
- You can write the auto-execution script in EXPECT, TCLSH, or ZSH. If you already installed the SmartScripts package, you can also write the post configuration script in PERL or Python.
- No restraints are required for the auto-execution script, for example, the signature `#/DELL-NETWORKING` that is required for the preconfiguration script.
- The auto-execution script has the ability to use configuration Dell Networking OS CLI using the utility name **F10do**.
- When the auto-execution script is complete, the start-up configuration is applied depending on the return status of the script:
 - Success – 0 — the start-up configuration is applied.
 - Failure – non-zero — the start-up configuration is not applied.

If the system is rebooted with reload-type set as normal-reload and an auto-execution script is present in the Flash directory, the following logs display:

```
Starting Dell Networking application
00:00:13: %STKUNIT1-M:CP %RAM-6-ELECTION_ROLE: Stack unit 1 is
transitioning to Management unit.
00:00:15: %STKUNIT1-M:CP %CHMGR-5-STACKUNITDETECTED: Stack unit
1 present
00:01:02: %STKUNIT0-M:CP %BMP-5-AUTOEXEC_START: The AutoExec
Script is Started.
```

After the script execution is successful, Dell Networking OS displays the following log:

```
00:04:05: %STKUNIT0-M:CP %BMP-5-AUTOEXEC_SUCCESS: The AutoExec
Script execution returned Success.
copy startup-config running-config
!
4774 bytes successfully copied
00:04:06: %STKUNIT0-M:CP %SYS-5-CONFIG_LOAD: Loading
configuration file
```

If the script returns a non-zero value (indicating a failure), the configuration file does not load in the system. You must debug it manually and reload the system with corrected script. If the auto-exec script fails, the system displays the following syslog:

```
00:00:34: %STKUNIT1-M:CP %BMP-5-AUTOEXE_FAILURE: The AutoExec
Script execution returned Failure.
```

Script Examples

The following are BMP script example.

Auto-Execution Script - Normal Mode

```
Dell#show reload-type
Reload-Type : normal-reload [Next boot : normal-reload]
Dell#show file flash://autoexec
#! /usr/bin/tclsh
puts [ exec f10do "show version" ]
puts [ exec date ]
puts "this is Autoexec script"
Dell#
Dell#
Dell#reload
System configuration has been modified. Save? [yes/no]: no
Proceed with reload [confirm yes/no]: yes
00:32:16: %STKUNIT1-M:CP %CHMGR-5-RELOAD: User request to
reload the chassis syncing disks... done
unmounting file systems...
unmounting /f10/flash (/dev/ld0h)...
unmounting /usr/pkg (/dev/ld0g)...
unmounting /usr (mfs:35)...
unmounting /f10 (mfs:21)...
unmounting /kern (kernfs)...
unmounting / (/dev/md0a)... done
rebooting
```

```

..
..
..
Starting Dell Networking application
00:00:13: %STKUNIT1-M:CP %RAM-6-ELECTION_ROLE: Stack unit 1 is
transitioning to Managementunit.
00:00:15: %STKUNIT1-M:CP %CHMGR-5-STACKUNITDETECTED: Stack unit
1 present.

```

The following line indicates the start of the auto-execution script.

```

00:00:16: %STKUNIT0-M:CP %BMP-5-AUTOEXEC_START: The AutoExec
Script is Started.
00:00:19: %STKUNIT1-M:CP %CHMGR-5-CHECKIN: Checkin from Stack
unit 1 (type S4810, 64 ports)
00:00:20: %00:00:20: %STKUNIT1-M:CP %CHMGR-0-PS_UP: Power
supply 0 in unit 1 is up
00:00:20: %STKUNIT1-M:CP %CHMGR-5-STACKUNITUP: Stack unit 1 is
up
00:00:21: %STKUNIT1-M:CP %CHMGR-5-SYSTEM_READY: System ready
00:00:21: %STKUNIT1-M:CP %RAM-5-STACK_STATE: Stack unit 1 is in
Active State.
00:00:22: %STKUNIT1-M:CP %IFMGR-5-OSTATE_UP: Changed interface
state to up: Ma 1/0
00:00:26: %S4810:1 %IFAGT-5-INSERT_OPTICS: Optics SFP inserted
in slot 1 port 30
00:00:27: %STKUNIT1-M:CP %CHMGR-5-PSU_FAN_UP: Fan 0 in PSU 0 of
Unit 1 is up
00:00:29: %S4810:1 %IFAGT-5-INSERT_OPTICS_PLUS: Optics SFP+
inserted in slot 1 port 11
00:00:36: %STKUNIT1-M:CP
%IFMGR-5-IFM_ISCSI_ACL_REGION_NOT_ALLOCATED: iSCSI Session
monitoring cannot be enabled without ACL regions allocated to
it. To enable iSCSI SessionMonitoring allocate cam-blocks to
iscsiopacl using cam-acl CLI and then save and reload.
00:00:36: %STKUNIT1-M:CP %IFMGR-5-IFM_ISCSI_ENABLE: iSCSI has
been enabled causing flowcontrol to be enabled on all
interfaces. EQL detection and enabling iscsi
profile-compellenton an interface may cause some automatic
configurations to occur like jumbo frames on allports and no
storm control and spanning tree port-fast on the port of
detection
00:00:36: %STKUNIT1-M:CP %SEC-5-LOGIN_SUCCESS: Login successful
on consoleDell>Dell#terminal length 0

```

The following line indicates that the auto-execution script is executing:

```

Dell#show version
Dell Networking Real Time Operating System Software
Dell Networking Operating System Version: 2.0
Dell Networking Application Software Version: 1-0(0-338)
Copyright (c) 1999-2012 by Dell Inc. All Rights Reserved.
Build Time: Thu Dec 27 21:32:28 2012
Build Path: /sites/sjc/work/build/buildSpaces/build06/
FIT-INDUS-1-0-0/SW/SRC
System image file is "dt-maa-s4810-72"
System Type: S4810Control
Processor: Freescale QorIQ P2020 with 2147483648 bytes of
memory.
128M bytes of boot flash memory.
1 52-port GE/TE/FG (SE)48 Ten GigabitEthernet/IEEE 802.3
interface(s)
4 Forty GigabitEthernet/IEEE 802.3 interface(s)
Dell#Wed Jan 2 22:47:34 GMT 2013
this is Autoexec script

```

The following line indicates the auto-execution script has completed successfully:

```
00:04:05: %STKUNIT0-M:CP %BMP-5-AUTOEXEC_SUCCESS: The AutoExec  
Script execution returned Success.
```

The following line indicates that the configuration file is loaded into the switch:

```
Dell#00:00:51: %STKUNIT1-M:CP %SYS-5-CONFIG_LOAD:  
Loading configuration file  
00:00:52: %STKUNIT1-M:CP %IFMGR-5-ASTATE_UP: Changed interface  
Admin state to up: Te 0/36  
00:00:53: %STKUNIT1-M:CP %IFMGR-5-ASTATE_DN: Changed interface  
Admin state to down: Ma 0/0
```


Preconfiguration Script - BMP Mode

```
#!/usr/bin/expect

#/DELL-NETWORKING

# Execute F10do and Print
proc print_f10do {cmd_str} {
    set str [exec f10do "$cmd_str"]
    set tmp_str [string map {\n \r\n} $str ]
    puts $tmp_str
}
set ftp_ip          "20.0.0.1"
set ftp_username    "lab"
set ftp_passwd      "lab"
set config_file     "s4810-10-startup-config"
set post_conf       "s4810-10-post-config.exp"
puts "Executing Pre-Config Script !!!!\r\n"
exec rm -rf "$config_file"
exec rm -rf "$post_conf"
puts "Downloading Startup Config and Post-Config Script from
$ftp_ip ...\r\n"
spawn ftp "$ftp_ip"
expect "Name .*:   "
send "$ftp_username\r\n"
expect "Password:  "
send "$ftp_passwd\r\n"
send "cd scripts\r\n"
expect "ftp>"
send "ls\r\n"
expect "ftp>"
send "get $post_conf\r\n"
expect "ftp>"
send "get $config_file\r\n"
expect "ftp>"
send "bye\r\n"
expect eof
after 5000
puts "Download Complete !!!\r\n"
if {[file exists $config_file]} {
    puts "Config File: $config_file downloaded successfully\r\n"
} else {
    puts "ERROR: Config File: $config_file - Not Found\r\n"
}
if {[file exists $post_conf]} {
    puts "Post Config Script: $post_conf downloaded
successfully\r\n"
} else {
    puts "ERROR: Post Config Script: $post_conf - Not
Found\r\n"
}
# Copy Config to Startup Config
print_f10do "show version"
after 5000
print_f10do "copy flash://$config_file startup-config"
print_f10do "yes"
after 5000
puts "Pre-Config Script Execution Successful !!!!!\r\n"
exit 0
```

Post Configuration Script - BMP Mode

The following example shows the post configuration script for the S4810 or S4820T platform:

```
#!/usr/bin/expect
#/DELL-NETWORKING
# Post Config Script for S4810-10
# Execute F10do and Print
proc print_f10do {cmd_str} {
set str [exec f10do "$cmd_str"]
set tmp_str [string map {\n \r\n} $str ]
puts $tmp_str
}
# Interfaces Configurations
set interface_list_slot [list "0/1" "0/5" "0/9"]
set interface_list [list "Te 0/1" "Te 0/5" "Te 0/9"]
set {remote_intf(Te 0/1)} "TenGigabitEthernet 0/1"
set {remote_intf(Te 0/5)} "TenGigabitEthernet 0/5"
set {remote_intf(Te 0/9)} "TenGigabitEthernet 0/9"
set hostname "S4810-10"
set max_min 10
set status_file "s4810-10-current-status.dat"
set ftp_ip "20.0.0.1"
puts [exec rstimer 30]
puts "\r\nReset Timer Complete\r\n"
# Open Staus File
set fp [open $status_file w]
puts $fp "=====\r\n"
puts $fp " Status: $hostname\r\n"
puts $fp "=====\r\n"
# Configure LLDP Protocol
puts "Configuring LLDP Protocol\r\n"
print_f10do "configure terminal"
print_f10do "protocol lldp"
print_f10do "no disable"
print_f10do "end"
# Check for Protocl LLDP
set lldp_output [ exec f10do "show runn | grep lldp" ]
if {[regexp "lldp" $lldp_output]} {
puts "LLDP is configured\r\n"
# Write the Result to Status File
puts $fp "LLDP is configured\n"
} else {
puts "ERROR: LLDP is not configured\r\n"
# Write the Result to Status File
puts $fp "ERROR: LLDP is not configured\r\n"
}
# Configure Interfaces
foreach intf_slot $interface_list_slot {
set intf "TenGigabitEthernet $intf_slot"
puts "Configuring $intf ... \n"
puts $fp "Configuring $intf ... \n"
print_f10do "configure terminal"
print_f10do "interface $intf"
print_f10do "no ip address"
print_f10do "no shutdown"
print_f10do "end"
after 200
}
# Wait for 2 mins for the Neighbor to come-up
puts "Wait for 1 min for the Neighbor to come-up\r\n"
after [expr {60 * 1000}]
```

```

puts $fp "=====\n"
puts $fp " Checking Conectivity thru LLDP\n"
puts $fp "=====\n"
# Check LLDP Configurations
foreach intf_slot $interface_list {
set min 0
set result 0
while {$result == 0 && $min < 5} {
set result_str [exec f10do "show lldp neighbors | grep
\"$intf_slot\""]
set tmp_str [string map {\n \r\n} $result_str]
puts $tmp_str
if {[regexp "$intf_slot" $result_str]} {
set result 1
if {[regexp "$remote_intf($intf_slot)" $result_str]} {
puts "Interface $intf_slot is Connected to
$remote_intf($intf_slot)\r\n"
puts $fp "Interface $intf_slot is Connected to
$remote_intf($intf_slot)\r\n"
} else {
puts "ERROR: Interface $intf_slot is Not Connected to Interface
$remote_intf($intf_slot)\r\n"
puts $fp "ERROR: Interface $intf_slot is Not Connected to
Interface$remote_intf($intf_slot) \r\n"
puts "LLDP Output for $intf_slot :\r\n $result_str \r\n"
puts $fp "LLDP Output for $intf_slot :\r\n $result_str \r\n"
}
}
continue
}
}
# Wait for 1 minute
puts "Interface is Not Connected\r\n"
puts "Wait for 1 min for the Neighbor to come-up\r\n"
after [expr {60 * 1000}]incr min
}
if {$result == 1} {
puts "Interface $intf_slot is Connected\r\n"
# Write Result to Status File
puts $fp "Interface $intf_slot is Connected\r\n"
} else {
puts "ERROR: Interface $intf_slot is Not Connected\r\n"
# Write the Result to Status File
puts $fp "ERROR: Interface
$intf_slot is Not Connected\r\n"
}
}
# Close & FTP Status File
puts $fp "=====\n"
close $fp
# Configure FTP - Interface
print_f10do "configure terminal"
print_f10do "interface TenGigabitEthernet 0/22"
print_f10do "ip address 20.0.0.34/16"
print_f10do "no shutdown"
print_f10do "end"
puts "Uploading Status File($status_file) to $ftp_ip ...\n"
spawn ftp "$ftp_ip"
expect "Name .*: "
send "lab\n"
expect "Password:"
send "lab\n"
expect "ftp>"
send "cd scripts\n"
expect "ftp>"
send "ls\n"
expect "ftp>"
send "put $status_file\n"

```

```

expect "ftp>"
send "ls\n"
expect "ftp>"
send "bye\n"
expect eof
print_f10do "configure terminal"
print_f10do "interface TenGigabitEthernet 0/22"
print_f10do "no ip address"
print_f10do "shutdown"
print_f10do "end"
puts "Post-Config Script Execution Successfull !!!!\r\n"
exit 0

```

BMP Operations on Servers Overview

The following sections describe how to prepare the different servers for BMP functionality:

- [DHCP Server](#)
- [File Server Settings](#)
- [Domain Name Server Settings](#)

DHCP Server

To configure the DHCP server use the following information.

DHCP Server Settings

Before you can use BMP mode on a switch, first configure a DHCP server.

Configure the DHCP server with the following set of parameters for each client switch. For more information, refer to the *Dell Networking OS Configuration Guide, Dynamic Host Configuration Protocol* chapter. To assign an IP to the system and other parameters, configure the DHCP server.

Update the following parameters on the appropriate DHCP server:

- **Boot File Name** — the Dell Networking OS image loaded on the system. Option 67 in the DHCP offer is the boot file name; the filename is BOOTP payload. If both are specified, option 67 is used. The system supports the TFTP, HTTP, HTTPS, SFTP, SCP, FTP, FLASH and USBFLASH protocols.
- **Configuration File Name** — the configurations applied to the system. The configuration file name is expected to use option 209. A preconfiguration script can also be given in option 209 to configure the device by itself for a download configuration file.

- **File Server Address** — the server where the Image and Configurations files are placed. The address is assumed to be a TFTP address unless it is given as a URL. The system supports the TFTP, HTTP, HTTPS, SFTP, SCP, and FTP protocols, as well as files stored in Flash.
- **Domain Name Server (Optional)** — the DNS server contacted to resolve the host name.
- **IP Address** — dynamic IP address for the system. Use this IP address only for file transfers.

The following lists the DHCP option codes.

6	Domain Name Server IP
60	Vendor class identifier
61	Class identifier
66	TFTP Server name
67	Boot filename
150	TFTP server IP address
209	Configuration file
230	User port stacking



Note: BMP eventually exits when a timeout occurs.

In the following scenarios, BMP requests a different DHCP offer.

- If the offer contains only a boot image that cannot be downloaded, BMP requests another DHCP offer.
- If you enable the **reload-type config-scr-download enable** command and the configuration file in the offer cannot be downloaded, BMP requests another DHCP offer.

DHCP Server IP Blacklist

If the process does not complete successfully, the DHCP server IP is blacklisted and the BMP process is re-initiated.

A DHCP server IP is maintained in the blacklist for 10 minutes. If a DHCP offer is received from the blacklisted DHCP server, the offer is rejected until the IP is alive in the blacklist (10 minutes).

MAC-Based Configuration

To configure the DHCP server to assign a fixed IP address, Dell Networking OS image, and configuration file based on the system's MAC address, use the BMP mode.

Using BMP, the same IP address is assigned to the system even on repetitive reloads and the same configuration file is retrieved when using the DNS server or the network-config file to determine the hostname.

The assigned IP address is only used to retrieve the files from the file server. It is discarded after the files are retrieved.

Following is a configuration example of DHCP server included on the most popular Linux distribution. The dhcpd.conf file shows the MAC-based IP and configuration file assignment are fixed.

Table 3-1. MAC-Based Configuration

Parameter Example	
<pre>option configfile code 209=text; option bootfile-name code 67=text; host HOST1{</pre>	
#####MAC to IP mapping	
<pre>hardware ethernet 00:01:e8:8c:4d:0e; fixed-address 30.0.0.20;</pre>	
#####Config file name could be given in the following way	FTP URL with IP address
<pre>option configfile "ftp://admin:admin@30.0.0.1/ pt-s4810-12";</pre>	HTTP URL with DNS
<pre>option configfile "http://Guest-1/pt-s4810-12";</pre>	TFTP
<pre>option configfile "pt-s4810-12";</pre>	
#####bootfile-name could be given in the following way	FTP URL with DNS
<pre>option bootfile-name "ftp:// admin:admin@Guest-1/Dell-SE-8.3.10.1.bin";</pre>	HTTP URL with IP address
<pre>option bootfile-name "http://30.0.0.1/ Dell-SE-8.3.10.1.bin";</pre>	TFTP URL with IP address

MAC-Based IP Address Assignment

To assign a fixed IP address and configuration file based on the system's MAC address, configure the DHCP server to deploy in BMP mode. In this way, the same IP address is assigned and the same configuration file is retrieved when the switch reloads.

Using a dynamic IP address assignment may cause the desired configuration to not load on the system because the IP address changes each time the system is reloaded.

For example, on a DHCP3 server, you can configure the assignment of a fixed MAC-based IP address and configuration file by entering the following lines of configuration parameters in the *dhcpd.conf* file on the server:

```
host S4810 {
    hardware ethernet 00:01:e8:81:e2:39;
    fixed-address 20.0.0.48;
    option configfile "customer.conf";
}
```

Class-Based Configuration

By matching a part of the string from the vendor class identifier (option 60) string, the image, configuration file, or the script file can be sent in the DHCP offer.

For example:

```
host dt-maa-z9000-11 {
    hardware ethernet 00:01:e8:a9:81:a3;
    fixed-address 10.16.151.175;
    option tftp-server-address 10.16.151.209;
    match if substring (option vendor-class-identifier,0,17) =
"TY=DELLNTW-Z9000 "; {
        filename "tftp://10.16.151.209/FTOS-ZB-9.3.0.0.bin";
        option configfile "ftp://anvltest:force10@10.16.151.209//
tftpboot/basic-l3-z9k <ftp://10.16.151.209//tftpboot/
basic-l3-z9k> ";
    }
    match if substring (option vendor-class-identifier,0,17) =
"TY=DELLNTW-s4810 "; {
        filename "tftp://10.16.151.209/FTOS-SE-9.3.0.0.bin";
        option configfile "ftp://anvltest:force10@13.4.4.4//
tftpboot/basic-l3-across-s6410-s4810_z9k.cfg";
    }
}
```

The option 60 is also called as vendor class identifier. It has five fixed fields and all the fixed fields have fixed offsets. The fixed fields are as follows:

- TY denotes the type of the device
- HW denotes the version of the box
- SN denotes the serial number of the device
- ST denotes the service tag of the device
- OS refers to the Dell Networking OS version in the device
- US refers to the user defined string (can be a string of 64 characters)

File Server Settings

Set up a file server and ensure connectivity.

To allow file transfers to the switch, configure the file server that holds the boot and configuration files. The system recognizes HTTP, HTTPS, SFTP, TFTP, FTP, USB, and Flash URLs.

For example:

- `tftp://server ip or name/filename`
- `ftp://user:passwd@serverip or name//mypath/Dell-A.B.C.D.bin`
- `flash://filename`
- `http://host[:port]/file-path`
- `https://[<user:pass>@]host[:port]/file-path`
- `sftp://user:passwd@server//path/filename`
- `usbflash://path/filename`

When loading the Dell Networking OS image, if the Dell Networking OS image on the server is different from the image on the local Flash, the system downloads the image from the server onto the local Flash and reloads using that image. If the Dell Networking OS image on the server is the same image, the system loads the configuration file, if present, or the startup-config without downloading a new image.

Domain Name Server Settings

To determine the host name applied in the switch startup configuration, when no configuration file is retrieved from the DHCP server, set up a domain name server (DNS).

The DNS server is contacted only when no configuration file is contained in a DHCP server response and the host name is not resolved from the network-config file on the switch.

Bare Metal Provisioning CLI

Overview

[Bare Metal Provisioning CLI](#) is supported on the **S4810, S4820T, S6000, Z9000** and **MXL** switch platforms.

In a data center network, Bare Metal Provisioning (BMP) automates the configuration and updating the switches, ensuring standard configurations across the installed devices.

To set up a single switch or a stack of switches with minimal effort, use the auto-configuration function.

BMP eases configuration in the following key areas:

- On S4810, S4820T, S6000, Z9000, and MXL Switch platforms running BMP:
 - Boot images and configuration files are specified in a dynamic host configuration protocol (DHCP) server.
 - Supports pre-configuration and post-configuration scripts to automatically load predefined configurations and Dell Networking OS images using DHCP and common file transfer mechanisms.
 - Switch access is allowed through all ports (management and user ports) with or without DHCP-based dynamic IP address configuration of a switch.
 - Configuration files are automatically downloaded from a file server and applied to the switch or stack.
 - Boots up in Layer 3 mode with interfaces already in No Shutdown mode and some basic protocols are enabled to protect the system and the network.

Commands

- [reload conditional nvram-cfg-change](#)
- [reload-type](#)
- [script post-config](#)
- [show boot bmp](#)
- [show reload-type](#)
- [stop bmp](#)

reload conditional nvram-cfg-change

Z **S4810**
S4820T

S6000, MXL
Switch

Syntax

reload conditional nvram-cfg-change

Defaults

None.

Command Modes

EXEC

Command History

This guide is platform-specific. For command information about other platforms, refer to the relevant *Dell Networking OS Command Line Reference Guide*.

Version 9.3(0.0)	Introduced on the S6000.
Version 9.2(0.0)	Introduced on the MXL Switch.
Version 9.1(0.0)	Introduced on the Z9000 and S4810.
Version 8.3.19.0	Introduced on the S4820T.

Usage Information

To upgrade any NVRAM changes to the chassis caused by the following CLI commands: **stack-unit {unit} stack-group**, **stack-unit {unit} port {fanout-portid} portmode quad**, **asf-mode**, **cam-acl**, **buffer-profile**, use the **reload conditional nvram-cfg-change** command on a switch running BMP.

reload-type

Z **S4810**
S4820T

S6000, MXL
Switch

Configure a switch to reload in Normal mode or as a DHCP client with all ports configured for Layer 3 traffic.

Syntax

```
reload-type {normal-reload | bmp [auto-save {enable | disable}]  
[config-scr-download {enable | disable}] [dhcp-timeout minutes] [retry-count  
number] [vendor-class-identifier description]}
```

To stop the BMP process, use the **stop bmp** command.

Parameters

normal-reload	Enable the normal reload type and disable BMP reload type. After performing the reload, the system retrieves the Dell Networking OS image and startup-configuration files from the flash after performing a reload.
bmp	(Default) Enable the BMP reload type. The system acts as a DHCP client and downloads the Dell Networking OS image and configuration file or script files from a specified DHCP server.
auto-save	To save the downloaded configuration or script file which are not saved by default, configure the auto save option. When you configure auto save, downloaded configurations are automatically saved to the startup configuration. Auto saving the downloaded configurations also requires enabling the config-scr-download parameter. Downloaded scripts are automatically saved to the autoexec script.
config-scr-download {enable}	(Optional) Enable the download of the configuration file or pre-configuration script from the DHCP / file servers.
config-scr-download {disable}	(Optional) Disable the download of the configuration file or pre-configuration script from the DHCP/ file servers.
dhcp-timeout <i>minutes</i>	(Optional) Configure the DHCP timeout (in minutes) after which the BMP exits. The range is from 0 to 50. If you enter a range of 0, the timeout is 0 (no limit). The default is disabled. Note: Dell Networking recommends setting the value to 2 or higher.
retry-count <i>number</i>	Configure the number of times to retry downloading the Dell Networking OS image, configuration file, or script file from the DHCP / file servers, if the servers are not reachable. The retry limit is from 0 to 6. If the retry limit is 0, no retry is performed. The default is 0.
vendor-class-identifier <i>description</i>	(Optional) Enter a brief description for the user defined field in option 60. Maximum is 64 characters long. User string is appended with Type, Hardware, Serial Number, Service Tag and OS Version. Note: This parameter replaces the deprecated user-defined-string parameter.

Defaults

BMP

Command Modes

CONFIGURATION

Command History

Version 9.3(0.0)	Updated the parameters for S6000.
Version 9.2(0.0)	Introduced on the MXL Switch. Updated the parameters for S4810, S4820T, and Z9000.
Version 9.1.0.0	Updated the command mode from EXEC Privilege to GLOBAL CONFIGURATION. Updated the parameter from jumpstart to bmp . Added support for the config-scr-download and user-defined-string commands. Supported platforms are S4810, S4820T, and Z9000.
Version 9.0.2.0	Introduced on the S6000.
Version 9.0.0.0	Introduced in Z9000.
Version 8.3.19.0	Introduced on the S4820T.
Version 8.3.1.0	Introduced on the S4810.

Usage Information

For an initial setup, the **config-scr-download** parameter of the **reload-type** command is enabled. After the configuration file is successfully downloaded, the **config-scr-download** parameter is automatically disabled. You can enable it again using the **reload-type** command.

Set the Auto Configuration mode (BMP or Normal reload) using the **reload-type** command. Next, use the **reload** command to reload the switch in the configured mode.

To copy the running configuration to the startup configuration, configure the reload-type by using the **write memory** command.

When a switch reloads in BMP mode, all ports, including the management port, are automatically configured as Layer 3 physical ports. The switch runs the DHCP client on all interfaces. You can reconfigure the default startup configuration, but you cannot reconfigure the DHCP timeout values.

If the switch attempts to contact a DHCP server and one is not found, it continues to send the DHCP discover messages until the DHCP time out occurs. If the DHCP offer is not received, use the **stop bmp** command to enable a premature timeout of BMP. The startup configuration is then loaded from the local flash on the switch.

To toggle between Normal and BMP Auto Configuration modes, use the **reload-type** CLI. Reload settings for Auto Configuration mode that you configure are stored in memory and retained for future reboots and software upgrades. To reload the switch in the last configured mode, BMP or Normal reload mode, use the **reload** command at any time.

Upgrade any configuration changes that have changed the NVRAM content by performing a reload on the chassis.

While BMP is on, the Dell Networking OS prompt changes to `Dell-BMP`.

Related Commands

show reload-type — displays the current reload mode (BMP or Normal mode).

stop bmp — stops the BMP process and prevents a loop if the DHCP server is not found.

script post-config

Z **S4810**
S4820T
S6000

MXL Switch

To run the post-configuration script after the pre-configuration script is executed during the BMP reload, ensure that this command is present in the startup-configuration.

Syntax **script post-config** {*script-name*}

Parameters

<i>scriptname</i>	Enter the name of the script to be run after the BMP start-up configuration has been applied.
-------------------	---

Defaults None.

Command Modes CONFIGURATION

Command History This guide is platform-specific. For command information about other platforms, refer to the relevant *Dell Networking OS Command Line Reference Guide*.

Version 9.3(0.0)	Introduced on the S6000.
------------------	--------------------------

Version 9.2(0.0)	Introduced on the MXL Switch and S4820T.
------------------	--

Version 9.1.0.0	Introduced on the Z9000 and S4810.
-----------------	------------------------------------

show boot bmp

Z **S4810**
S4820T
S6000

MXL Switch

Displays the current state of the BMP process.

Syntax **show boot bmp**

Defaults None.

Command Modes EXEC

Command History

Version 9.3(0.0)	Introduced on the S6000.
------------------	--------------------------

Version 9.2(0.0)	Introduced on the Z9000, S4810, S4820T, and MXL Switch.
------------------	---

Example

```

Dell# show boot bmp
Config Download
via DHCP: enabled
BMP State : Waiting for boot options
...
BMP State : Received DHCP offer from DHCP server 25.1.1.1

***** SELECTED OFFER DETAILS *****
Server type= DHCP
Acquired IP= 25.1.1.25
Subnet-mask = 255.255.0.0
Image file = tftp://25.1.1.1/boot_file.bin
config file = tftp://25.1.1.1/config_file.cfg
Server IP = 25.1.1.1
TFTP Server IP = NIL
DNS IP = 25.1.1.1
Routers = NIL
*****

...
BMP State : Downloading image boot_file.bin from 25.1.1.1
...
BMP State : Image boot_file.vin successfully downloaded
...
BMP State : BMP process is successfully completed

```

**Related
Commands****reload-type** — Configures the reload mode as Normal or BMP.

show reload-type

Z **S4810****S4820T****S6000**

MXL Switch

Display the reload type currently configured on the system.

Syntax**show reload-type****Defaults**

None

Command Modes

EXEC Privilege

**Command
History**

Version 9.3(0.0)	Updated the parameters for S6000.
Version 9.2(0.0)	Introduced on the MXL Switch. Updated the parameters for S4810, S4820T, and Z9000.
Version 9.1.0.0	Updated the parameters for S4810 and Z9000.
Version 9.0.2.0	Introduced on the S6000.
Version 9.0.0.0	Introduced in Z9000.
Version 8.3.19.0	Introduced on the S4820T.

Version 8.3.11.4 Introduced on the Z9000.

Version 8.3.10.0 Introduced on the S4810.

Usage Information

Check the currently configured Auto-Configuration mode (BMP or Normal reload) on a switch running BMP using the **show reload-type** command.

To display the current reload mode for BMP, use the **show bootvar** or **show system brief** commands. The **show bootvar** command includes the path of the Dell Networking OS image file retrieved from a DHCP server when BMP is running, but not after you exit BMP.


Example

```
Dell#show reload
reload-type           : bmp [Next boot : bmp]
auto-save             : enable
config-scr-download  : enable
dhcp-timeout         : 1
vendor-class-identifier : Device 609
retry-count          : 3
Dell#
```

Related Commands

reload-type — Configure the reload mode as BMP or Normal.

stop bmp

 **S4810**

S4820T

S6000,

MXL Switch

To prevent an infinite loop, stop the switch from reloading in BMP mode.

Syntax

stop bmp

Note: Replaces the **stop jumpstart** command.

Defaults

None.

Command Modes

EXEC

Command History

This guide is platform-specific. For command information about other platforms, refer to the relevant *Dell Networking OS Command Line Reference Guide*.

Version 9.3(0.0) Introduced on the S6000. Replaces the **stop jumpstart** command.

Version 9.2(0.0) Introduced on the MXL Switch and S4820T. Replaces the **stop jumpstart** command.

Version 9.1(0.0) Introduced on the Z9000 and S4810. Replaces the **stop jumpstart** command.

Usage Information

Use the **stop bmp** command on a switch running BMP if the switch enters a loop while reloading in BMP mode. A loop occurs when the switch is continuously trying to contact a DHCP server and a DHCP server is not found. The **stop bmp** command stops the switch from connecting to the DHCP server. After you use the **stop bmp** command, the next default reload type is a normal reload as indicated in the **show reload-type** or **show system-brief** commands.

The **stop bmp** command behaves as below in different circumstances,

- While the Dell Networking OS image upgrade is in-progress, it aborts the BMP process after the Dell Networking OS image is upgraded.
- When applying configurations from the file, it aborts the BMP process after all the configurations are applied in the system.
- When running the pre-config or post-config script, it stops execution of the script and aborts the BMP process immediately.
- When downloading the configuration or script file, it aborts the BMP process after the download, it does not apply the configuration or run the script.

When you enter GLOBAL CONFIGURATION mode during the BMP process, warning / error messages display to avoid configuration conflicts between you and the BMP process.

Related Commands

reload-type — Configures Reload mode as Normal or BMP mode.

Smart Scripting

[Smart Scripting](#) is supported on the **S4810, S4820T, S6000, Z9000** and **MXL** switch platforms.

Smart Scripting allows you to add functionality to Dell Networking switches without requiring updates to the Dell Networking OS release. Smart Scripting is available as a separate installable package that supports SQLite DB, and NET SNMP applications and TCL, Expect, Perl, Python, and Unix scripting languages.

The Smart Scripting package supports smart utility (SmartUtils) application programming interfaces (APIs) providing developers with an easier way to invoke switch operations by creating and running Tcl, Expect, Perl, Python, Unix shell scripts and ZSH scripts on the Dell Networking OS. API library files describe the functions supported in PERL, Python, and UNIX scripts.

A separate package has been extended with HTTP and HTTPS daemons based on CGI scripts and a Web-based graphical user interface. For more information about this package, see [Web Server with HTTP Support](#). For information about the HTTP requests supported by the REST API, see [REST API Commands](#).

Overview

Using Smart Scripting, network administrators can create custom Expect, PERL, Python, Tcl, UNIX and ZSH shell scripts to manage and interact with Dell Networking switches/routers in the network. Smart Scripting provides support for:

- Modules required to run PERL scripts, such as the software development kits (SDKs) for VMware and vCenter/vSphere.
- Modules that implement requested Python features, such as AMQP (message queuing), XML-RPC (arbitrary data exchange), and Twisted (event-driven networking engine).

With Smart Scripting, there is no need to learn proprietary scripting languages, allowing for faster development and deployment of custom scripts.

Smart Scripting also offers solutions in a UNIX environment that are useful to cloud administrators who are familiar with working in a UNIX shell. Smart Script support in a UNIX environment allows you to invoke standard UNIX utilities such as netstat, tcpdump, ls, chmod, chown, and so on.

Smart Scripting includes a convenient set of API function libraries that you can refer to when you create Expect, PERL, Python, Tcl, UNIX and ZSH shell scripts. A representation of CLI functions to retrieve data from the Dell Networking OS and change configuration parameters on Dell Networking switches is provided in the API libraries. Script writers include API function calls made directly on the Dell Networking OS command line interface (CLI) in Expect, PERL, Python, Tcl, UNIX and ZSH shell scripts.

For example, the API functions used in a script include setting up a telnet session, gathering data on the switch, sending information to the CLI, and closing telnet sessions. By using simple function calls, you do not have to include the parsing code required for telnet sessions and retrieving configuration information.

Smart Scripting supports running a script either from the Dell Networking OS CLI or directly from a UNIX shell. Run scripts periodically, based on events, at boot up or after the switch is initialized. You can run scripts in the foreground or background and display the progress of a script. If required, you can cancel, stop, or resume scripts.

Smart Scripting allows you to automate common management and maintenance tasks, such as:

- Building visibility and/or discovery programs.
- Creating custom logging.
- Reporting configuration information.
- Reporting switch memory usage, configured virtual local area networks (VLANs), and other operating and configuration parameters.
- Creating custom APIs for external applications to access the switch.
- Automating custom provisioning of network devices to support server virtualization.

For example, you can automate any of the following tasks:

- Monitor the configuration of switch ports to verify that no change occurs and generate an alarm if a configuration change is detected as part of a cloud-computing deployment.
- Stage CLI command requests received from a customer. If a link flaps, the command completion status is held in the script so you can see when the management plane reconnects.
- Generate time-based reports to receive updates on network status on a periodic basis.
- Query an external, configuration management database on a remote server to retrieve information on port operation, and reconfigure switch ports based on the data received.
- Apply additional time-based access-control lists (ACLs) to limit after hours access.

- Monitor network requests; for example, “find a specified MAC address” or “generate a health-check heartbeat”.
- Create a simple menu of options that a non-network administrator can use to create requests to be sent to the network.

Smart Scripting consolidates management data inside a switch and sends it to management consoles, databases, or applications – reducing polling and network traffic. For example, you can use a script as part of a cloud-computing deployment to detect when the network has changed, query a database server for Configuration Management Database (CMDB) information, and ultimately apply network changes based on the data.

Downloading the Smart Scripting Package

Download the Smart Scripts package from the Dell Networking OS website as a file named:

- **SMARTSCRIPTS-P-3.1.0.0.tar.gz** for S4810 and S4820T
- **SMARTSCRIPTS-I-3.1.0.0.tar.gz** for Z9000 and S6000
- **SMARTSCRIPTS-M-3.1.0.0.tar.gz** for MXL Switch

To download the SmartScript package to a dedicated location, use one of the following method: tftp, ft, flash, or nfs mount.

The Smart Scripting package includes the following files and functionality:

- SQLite database and PERL, Python, TCL interface to SQLite
- NetSNMP client and PERL, Python interface to the client
- PERL interpreter and associated files
- Python interpreter and associated files
- Expanded set of UNIX utilities
- Web-Server with HTTP Support (see [Web Server with HTTP Support](#))

Installing Smart Scripting

After you download the Smart Scripts package, install the file from local flash memory on a switch or from an external drive on a network server. Because the installation takes time, it is performed in the background. When the download is complete, a message displays on the console. The package installation updates the running configuration file.



CAUTION:

You can modify (for example, edit or rename) the files downloaded with the Smart Scripting package only in the directory in which you install the package. Never modify the files in other system directories.

To install the Smart Scripting package, download it from the Dell Networking web portal:

1. On a PC or other network device, go to the Dell Networking web portal at <https://www.force10networks.com/CSPortal20/Main/SupportMain.aspx>. Click **Login**, enter your user ID and password, and click the **Login** button.
2. On the Customer Support page, click the **Software Center** tab.
3. In the left hand column, click **Automation Software**.
4. At the bottom of the Terms and Conditions page, click **I agree**.
5. On the Automation Software page, under Software, select the file for the switch from the following list:
 - **SMARTSCRIPTS-P-3.1.0.0.tar.gz** for S4810 and S4820T
 - **SMARTSCRIPTS-I-3.1.0.0.tar.gz** for Z9000 and S6000
 - **SMARTSCRIPTS-M-3.1.0.0.tar.gz** for MXL Switch
6. In the dialog box, select the path for the local flash on the switch or a directory path on a network server where you want to download the **SMARTSCRIPTS** package for your switch.
7. When the download is complete, use the **package install** command from the Dell Networking OS CLI on a switch to install the Smart Scripting package.

Command Syntax	Command Mode	Task
package install { flash://filename ftp://userid:password@host-ipaddress/dir-path tftp:// host-ipaddress/dir-path }	EXEC Privilege	Install the Smart Scripting package from local flash memory or a network server to a dedicated location on your server for script storage.
Where:		
<ul style="list-style-type: none"> • flash://filename installs the Smart Scripting file stored in flash memory on the switch. • ftp://userid:password@host-ipaddress/filepath logs in and installs Smart Scripting from a file stored on an FTP server. • tftp://host-ipaddress/filepath installs Smart Scripting from a file stored on a TFTP server. • nfsmount://filepath copies from the nfs mount file system 		

To remove an installed Open Automation package, such as Smart Scripting, use the **package uninstall** command.

To follow the progress of a package installation (or removal), use the **show packages** command.

Displaying Installed Packages

To view the Open Automation packages currently installed on a switch, including version numbers and content, use the **show packages** command.

Command Syntax	Command Mode	Task
show packages	EXEC Privilege	View package information.

Uninstalling Smart Scripting



Caution: Before you uninstall the Smart Scripting package, stop all scripts that are currently running using the **no script *script-name*** command. You must also manually stop the http server daemon.



Note: If installed, uninstall the VSN Agent package before uninstalling Smart Scripting package.

Uninstalling the Smart Scripting package removes it from the internal flash memory.

Command Syntax	Command Mode	Task
package uninstall <i>package-name</i> Enter the name of the Smart Scripting package, exactly as it appears in the show packages list.	EXEC Privilege	Uninstall the Smart Scripting package stored on the switch.

Limits on System Usage

Smart Scripting establishes limits on system processes for the following attributes (regardless of the user-privilege level or scripting method) to restrict CPU and memory usage:

Table 5-1. Limits on System Attributes

System Attribute	Value	Description
cputime	unlimited	Maximum amount of time used by a process.
filesize	unlimited	Largest file size (in bytes) that can be created.
datasize	131,072	Maximum size (in bytes) of the data segment for a process; this value defines how far a program may extend its break with the sbrk(2) system call.

Table 5-1. Limits on System Attributes

System Attribute	Value	Description
stacksize	2,048	Maximum size (in bytes) of the stack segment for a process; this value defines how far a program's stack segment may be extended. Stack extension is performed automatically by the system.
coredumpsize	unlimited	Largest size (in bytes) of a core file that may be created
memory use	233,244	Maximum size (in bytes) to which a process's resident set size may grow. This value imposes a limit on the amount of physical memory to be given to a process; if memory is tight, the system prefers to take memory from processes that are exceeding their declared resident set size.
memorylocked	77,741	Maximum size (in bytes) which a process may lock into memory using the mlock(2) function.
maxproc	160	Maximum number of simultaneous processes allowed for the user ID.
openfiles	64	Maximum number of open files for this process.

Supported UNIX Utilities

Smart Scripting supports the invocation of the following UNIX utilities in the scripts you run:

Table 5-2. Supported UNIX Utilities

UNIX Utility	Function
arp	Address resolution display and control.
awk	Pattern scanning and processing language.
basename	Return filename or directory portion of pathname.
bc	An arbitrary precision calculator language.
cat	Concatenate and print files.
chmod	Change file modes.
chown	Change file owner and group.
cksum	Display file checksums and block counts.
cut	Select portions of each line of a file.
date	Display or set date and time.
dd	Convert and copy a file.
df	Display free disk space.
env	Set and print environment.
expr	Evaluate expression.
fc	List the history of commands on the computer.

Table 5-2. Supported UNIX Utilities (continued)

fg	Change the background process to foreground.
file	Determine file type.
find	Walk a file hierarchy.
ftp	Internet file transfer program.
getopts	Called each time you want to process an argument.
grep	Print lines matching a pattern.
hostname	Set or print name of current host system.
ifconfig	Configure network interface parameters.
iostat	Report I/O statistics.
ln	Make links.
ls	List directory contents.
md5	Calculates and verifies 128-bit MD5 hashes.
more	A filter for browsing text files.
netstat	Show network status
nice	Execute a utility with an altered scheduling priority.
nohup	Invoke a command immune to hangups.
ping	Send ICMP ECHO_REQUEST packets to network hosts.
ps	Process status.
pwd	Return working directory name.
sed	Stream editor.
sleep	Suspend execution for an interval of time.
sort	Sort or merge text files.
ssh	Open SSH client (remote login program).
stty	Used for changing the settings of a UNIX computer terminal.
tail	Display the last part of a file.
test	Condition evaluation utility.
ulimit	Get and set process limits.
umask	Set file creation mode mask.
vmstat	Report virtual memory statistics.
wait	Await process completion.
wc	Word, line, and byte count.
who	Display the users who are currently logged in.

Smart Utils

When you install the Smart Scripting package, sample PERL and Python scripts are installed in the `/usr/pkg/scripts/sample_scripts` directory. You can also create your own customized scripts and store them anywhere on the switch, such as in a `/f10/flash_scripts` directory.

In addition, you can use the PERL, Python, and UNIX APIs to create scripts that invoke function calls directly in the Dell Networking OS CLI that are collectively called smart utils. These APIs provide a shortcut when writing scripts. For more information, refer to the following sections:

- [Using the PERL API](#)
- [Using the Python API](#)
- [Using UNIX Shell Scripting](#)

For instructions about how to run a PERL, Python, or UNIX script from the Dell Networking OS CLI, refer to [Scheduling Time / Event-based Scripts](#).

For information about how to run a PERL, Python, or UNIX script directly from a UNIX shell, refer to [Running a Script from the UNIX Shell](#).

Creating a User Name and Password for Smart Scripting

Before you run a script from the Dell Networking OS CLI, you may want to configure an additional user name and password to be used only to run scripts on a switch. Use the user name and password to log into a UNIX shell and apply the read-write privileges assigned to the user name when a script is run with the **script** command from the Dell Networking OS CLI.

The user name is an optional keyword in the **script** command (refer to [Scheduling Time / Event-based Scripts](#)). To satisfy the requirements for a UNIX BSD login, the username must be less than 16 characters. A username used to run scripts cannot contain special characters.

Command Syntax	Command Mode	Task
username <i>name</i> password <i>password</i>	CONFIGURATION	Create an additional user name and password that are used to log in to a shell and apply read-write privileges when a script is run.

Logging in to a NetBSD UNIX Shell

To log into the NetBSD UNIX shell on a switch to directly use any of the UNIX commands described in [Supported UNIX Utilities](#) or to run a script, use the **start shell** command. You are prompted to enter a user name and password before you can access the shell. Login is performed using SSHv2.

Command Syntax	Command Mode	Task
start shell	EXEC Privilege	Access the shell to run UNIX commands or a script (refer to Running a Script from the UNIX Shell).

Downloading Scripts to a Switch

Download a script to the switch using TFTP, FTP, or FLASH. Save the script to the dedicated script storage location, /usr/pkg/ss-scripts.

Command Syntax	Command Mode	Task
mount nfs <i>nfs-server-ip:</i> remote_dir <i>mount_name</i> [username <i>username</i> password <i>password</i>]	CONFIGURATION	Configure the folders to mount a remote directory in the local Dell Networking OS path through a network file system (NFS). Enter the name of the remote directory to be mounted through the network file system and the name of the folder in the local system.
script get url	EXEC	Copy a script to a switch. Downloaded files are stored in the following path: /usr/pkg/ss-scripts
script remove { <i>file_name</i> <i>file-name</i> all }	EXEC	Remove a script from a switch.

Setting a Search Path for Scripts

Create a path to the location where scripts are stored to be used by Dell Networking OS when searching for scripts. This negates the need to specify a specific path when executing a script.

Command Syntax	Command Mode	Task
script path <i>path-name</i>	CONFIGURATION	Set a search path for a script in Dell Networking OS. The script path can contain a network file system mounted directory (defined in the mount nfs command). The path is added to a script search list allowing the system to search all locations for the script name. If the script is in multiple locations, the system uses the first instance of the script found.

Scheduling Time / Event-based Scripts

Schedule scripts to execute periodically, based on an event, at a specific time, at boot up, or after you configure the switch. Manage scripts to stop executing after a set period of time or configured to run at optimal times for critical resources such as switch CPU load or packet loss.

To access the Dell Networking OS CLI via scripts, the Dell Networking OS provides an utility called **f10do**, which is also a system command. For more information about the properties of **f10do**, please refer to [Preconfiguration Process for Scripts](#).

Triggering a Script to Run

To trigger scripts to run periodically or based on an event, use the following commands.

Command Syntax	Command Mode	Task
Schedule a script to run in EXEC mode.		
script execute <i>script-name</i> start { now <i>time-date</i> <i>time</i> } [stop { <i>at time-date</i> after time }] [args arguments] [username username] [bg]	EXEC	Schedule a script to execute at a specific time and optionally stop after a specified time. By default, the script runs in the foreground. To run the script in the background, use the bg parameter.

Command Syntax	Command Mode	Task
----------------	--------------	------

This example shows how you can schedule the script named “hello.txt” to start the execution immediately and stop after 20 minutes:

```
Dell#script execute hello.tcl start now stop after 00:20 args "hi" username test
```

- Scheduled scripts can be un-configured/stopped/killed/resumed only by the configured user (test) or higher privileged user.
- In general, the Privilege level 15 (highest privilege user in Dell Networking OS CLI) user can configure/reconfigure the explicit username in “script execute” CLI.

The following example shows how you can schedule the script named “sample.zsh” to start after one hour and 12 minutes from now and runs the script in the background:

```
Dell#script execute sample.zsh start 01:12 bg
```

Define a trigger event.

script trigger-event <i>event-name</i> { log-event { tag <i>tag</i> } severity <i>severity level</i> } time-event <i>time</i> cpu-usage <i>percentage</i> mem-usage <i>percentage</i> }	CONFIGURATION	Define an event to use to trigger scripts to run by a log event (for example, SYSLOG in Dell Networking OS), a time-based event or when CPU or memory usage reaches a pre-determined percentage. The script runs in the background.
---	---------------	---

This example shows how you can define the event to trigger a script when a SYSLOG event with pattern “OSTATE_UP” occurs:

```
Dell(conf)#script trigger-event event1 log-event tag "OSTATE_UP"
```

Schedule a script to run based on defined events.

script execute <i>script-name</i> [concurrent] triggered-by <i>event-name</i> [args <i>arguments</i>] [username <i>username</i>]	CONFIGURATION	Schedule a script to run triggered by a defined trigger event.
---	---------------	--

This example shows how you can schedule the script named “sample.pl” to run concurrently when trigger event named “event1” takes place:

```
Dell(conf)#script execute sample.pl concurrent triggered-by event1 args "hi" username "user1"
```

Where *event1* is defined by a trigger event:

- If the trigger is associated with a script on Concurrent mode, a new instance of the script is spawned at every occurrence of the event. **Note:** This is limited to a maximum of 10 instances at any given time and further events are ignored.
- If the trigger is associated with a script on Singleton mode, a new instance of the script is not created at every occurrence of the event, if it is already running. The default mode is Singleton.

Command Syntax	Command Mode	Task
Schedule a script to run periodically.		
script execute <i>script-name</i> start { now <i>time-date</i> <i>time</i> } repeat { weekdays <i>weekday</i> days <i>day</i> minutes <i>minutes</i> } [stop { at <i>time-date</i> after <i>time</i> }] [args <i>argument</i>] [username <i>username</i>]	CONFIGURATION	Schedule scripts to run at a certain time, to be repeated, or to stop at a specified time or by a string of arguments. The script runs in the background.
<p>This example shows how you can repeat the execution of the script named “HA_script.pl” which starts after two minutes and repeats every 20 minutes:</p> <pre>Dell(conf)# script execute HA_script.pl start 00:02 repeat minutes 20 username "user2"</pre> <p>The following example shows how you can repeat the execution of the script named “clear-logs.tcl” which starts on 05/11/13 at 13:45 and repeats 13th day of every month:</p> <pre>Dell(conf)# script execute clear-logs.tcl start 13:45-05/11/13 repeat days 13</pre> <p>The following example shows how you can schedule the reloads every first month at 00:00</p> <pre>Dell(conf)# script execute reload.zsh start 00:00 03/01/14 repeat days 1</pre> <p>You can use the “f10do” utility to the above script “reload.zsh”as:</p> <pre>#!/usr/bin/zsh f10do -r "write memory" sleep 5 f10do -r "reload" f10do "yes"</pre>		
Assign scripts to execute on boot up.		
script list execute <i>list-name</i> { on-boot network-up }	CONFIGURATION	Assign the order and run levels to a list of boot scripts to execute on boot up or after the switch is completely initialized.
<p>This example shows how you can execute a list of scripts before the configuration file is loaded in the Dell Networking OS:</p> <pre>Dell(conf)#script list execute boot-scripts on-boot</pre> <p>where boot-scripts file contains the following:</p> <ul style="list-style-type: none"> • seq1 script simple_script.py • seq2 script noshut.pl “te 0/0 te 0/1” • seq3 script creat_vlt.zsh “te 0/14” 		
Supervise the scheduled scripts.		

Command Syntax	Command Mode	Task
script execute <i>script-name</i> watch [start { now at <i>date-time</i> after <i>time</i> }] [stop { after <i>time</i> } repeat { weekdays <i>day</i> days <i>day</i> minutes <i>minutes</i> }] [args <i>arguments</i>] [username <i>username</i>]	CONFIGURATION	Monitor the supervised scripts definitely (with stop and repeat keyword) or indefinitely (with just start keyword).

This example shows how you can monitor the script named “sample.tcl” which starts from now onwards:

```
Dell(conf)#script execute sample.tcl watch start now args "sample" username test
```

- All the supervised scripts will be re-started at the maximum of three times, if the script exits (normally or abnormally) within 10 minutes from start.
- If the supervised scripts exits (normally or abnormally) more than three times within the time span of 10 minutes, then the script will be moved to “Blocked” state and no further executions will be scheduled till the network administrator clears / resets it using the “**script clear**” command.

Monitor or supervise the schedule script using watch.

script execute <i>script-name</i> watch start { now <i>date-time</i> <i>time</i> } [args <i>arg-string</i>] [username <i>username</i>] [bg]	EXECUTION	Monitor the supervised scripts using the watch keyword.
---	-----------	---

This example shows how you can monitor or supervise the script named “monitor_interface.tcl” which starts after 15 minutes from now:

```
Dell#script execmonitor_interface.tcl watch start 00:15 bg
```

when the script completes its execution, it is restarted as you monitor the script.

SQLite

The Dell Networking OS supports the SQLite database engine as it is self-contained, server less, zero-configuration, and transactional. This database performs the following tasks:

- Stores all the periodically collected data / statistics from the Dell Networking OS.
- Stores Dell Networking OS Configuration files.
- Other script logs can be stored, updated, deleted, and transferred using this database.

Dell Networking OS provides Perl, Python, and tclsh interface for SQLite. As a result, all the database handling can be done from the user developed custom scripts itself.

NET SNMP Client

To manage the device using SNMP through scripts, Dell Networking OS supports NetSNMP client. You can run scripts that use SNMP to manage the data center/ devices from within the Dell Networking OS itself. Dell Networking OS provides Perl and Python interfaces for NetSNMP client and to execute the SNMP operations, you can also use **tcsh exec** command. As a part of Net SNMP client, the following utility applications are supported:

- snmpwalk
- snmpget
- snmpset
- snmpbulkget
- snmpbulkwalk
- snmpgetnext

Managing Executed Scripts

Manage or control the scripts such as to stop, kill, resume, unscheduled, and clear by using any one of the following commands.

Protect system resources by scheduling scripts to run when resources are within configured threshold parameters. System resources include switch CPU, memory, and file system.

```
Dell#show script process detail
```

JobID	Script Type	Script Status	Username	Script Name	Args
000	TRIGGER	Running	admin	admin.pl	
001	REPEAT	Running	user1	sample.zsh	hello
002	CONF WATCH	Watch Blocked	user2	args.pl	
003	EXEC	Scheduled	user2	args.pl	hi

```
Dell#
```

To monitor the running script, use the **watch** command.

```
Dell#show script watch details
```

JobID	Script Type	Script Status	Reset Count	Watch StartTime	Script NameArgs	Username
002	CONF WATCH	Watch Blocked	3	Tue Dec 3 18:04:28 2013	args.pl	user2

To control the script that is running based on the system resources such as CPU, memory or disk I/O usage, use the **switch script limit cpu <percentage> memory <percentage> diskio <percentage>** command.

The scripting framework is enabled with system resource usage protection. So, when the system is running at a high cpu, memory or disk I/O, it automatically suspends all the user-configured running scripts and also prevents scheduling new scripts further till the system resource usage comes down to nominal levels. The following example is used to suspend the scripts when the cpu crosses 70, memory usage is above 60, or the disk I/O is greater than 70.

```
Dell(conf)#switch script limit cpu 70 mem 60 diskio 70
```

Note: For diskI/O, /usr/pkg partition disk usage is considered.

To suspend a currently running script, use the **script stop** command.

```
Dell#script stop script-name admin.pl
```

To resume suspended script, use the **script resume** command.

```
Dell#script resume job-id 0
```

To clear the supervised script that is in blocked state, use the **script clear** command.

```
Dell#script clear script-name args.pl
```

To clear all the watch blocked state scripts, use the **script clear all** command.

```
Dell#script clear all
```

To terminate a script that is running, use the **script kill** command.

```
Dell#script kill script-name admin.pl
```

To terminate all the scripts that are running, use the **script kill all** command.

```
Dell#script kill all
```

To unschedule an EXEC mode script, that is scheduled to run later, use the **script unschedule** command.

```
Dell#script unschedule job-id 4
```

Note: For information about running a script directly from a UNIX shell without using the Dell Networking OS CLI, refer to [Running a Script from the UNIX Shell](#).

Viewing Script Information

To view information on currently stored, scheduled, and running SmartScripts, use the following **show** commands.

Command Syntax	Command Mode	Task
show script file [detail]	EXEC Privilege	Display the list of stored files in the script path. Enter detail to show the detail output of the file.
show script process [detail]	EXEC Privilege	Display list of scripts that are scheduled or running. Enter detail to display the detailed status of the scripts.
show script watch [detail]	EXEC Privilege	Display list of supervised scripts that are scheduled or running. Enter detail to display the detailed status of the supervised scripts.

Running a Script from the UNIX Shell

You can run any PERL, Python, and UNIX script stored on a switch from either the Dell Networking OS CLI (refer to [Scheduling Time / Event-based Scripts](#)) or directly from a NetBSD shell on the switch.

When you run a script from a UNIX shell, first access the shell by using the **start shell** command. You are prompted to enter a user name and password configured with the **username** command (refer to [Creating a User Name and Password for Smart Scripting](#)).

[Figure 5-1](#) shows examples of how to execute a PERL, Python, and UNIX shell script directly from a NetBSD shell on Dell Networking OS.

Figure 5-1. Execution of a PERL, Python, and Shell Script from a UNIX Shell: Example

```
Dell# start shell <----- Log on to a UNIX shell

4.4 BSD UNIX () (tty0)

login: admin
Password:
Copyright (c) 1996, 1997, 1998, 1999, 2000, 2001, 2002
    The NetBSD Foundation, Inc. All rights reserved.
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
    The Regents of the University of California. All rights reserved.

$ cd /f10/flash/scripts
$ ls <----- List the existing scripts
createVlans.pl  createVlans.py  createVlans.sh

$ ./createVlans.pl 12 14 <----- Execute the PERL script using either command
$ perl createVlans.pl 12 14

$ ./createVlans.py 12 14 <----- Execute the Python script using either command
$ python createVlans.py 12 14

$ ./createVlans.sh 12 14 <----- Execute the UNIX shell script
```

Using the PERL API

The RESTAPI Interface package provides a PERL API library containing the supported functions (described in [Supported UNIX Utilities](#)), which you can use in a PERL script to invoke Dell Networking OS operations on a switch. The PERL API library is stored in the F10SmartUtils.pl file at /usr/pkg/scripts/smartutils. Code Dell Networking OS API functions in a PERL script as shown in the following example:

Figure 5-2. PERL Script with API function call: Example

```
#!/usr/pkg/bin/perl -w

require '/usr/pkg/scripts/smartutils/F10SmartUtils.pl'; <----- Load the PERL API

usage() if ($#ARGV < 1);
($start,$end)=@ARGV;

$startVlan = $start;
$endVlan = $end;

for (my $i=$startVlan;$i<=$endVlan;$i++) {
    my $createvlanId = F10CreateVlanId($i); <----- Invoke a PERL API function
}

sub usage {
    print "usage: createVlans.pl <start> <end>\n";
    exit;
}
```

[Supported Dell Networking OS API Functions in PERL Scripts](#) describes the supported functions and required arguments that you can use in PERL scripts running on a Dell Networking switch to connect through a telnet session and gather information or configure parameters through the CLI.

Table 5-3. Supported Dell Networking OS API Functions in PERL Scripts

PERL API Function	Arguments	Description
F10AddLagIntToVlan	(lagId, vlanId, tagFlag)	Adds a LAG interface to a VLAN as either tagged or untagged. tagFlag values: 1 (tagged) or 0 (untagged).
F10AddPhyIntToVlan	(stackUnitNum, portId, vlanId, tagFlag)	Adds a physical interface to a VLAN as either tagged or untagged. tagFlag values: 1 (tagged) or 0 (untagged).
F10CreateVlanId	(vlanId)	Creates a VLAN on the switch.
F10DeleteVlanId	(vlanId)	Deletes a VLAN on the switch.
F10ExecShowCmd	(command)	Executes a specified show command.
F10MakeLagIntNoShutdown	(lagId)	Enables the specified port channel.
F10MakeLagIntShutdown	(lagId)	Disables the specified port channel.
F10MakeLagIntSwitch	(lagId)	Configures the specified port channel (LAG) as a Layer 2 switchport.
F10MakePhyIntNoShutdown	(stackUnitNum, portId)	Enables the specified port.
F10MakePhyIntShutdown	(stackUnitNum, portId)	Disables the specified port.
F10MakePhyIntSwitch	(stackUnitNum, portId)	Configures the specified port as a Layer 2 switchport.
F10MakeVlanIntNoShutdown	(vlanId)	Enables the specified VLAN interface.
F10MakeVlanIntShutdown	(vlanId)	Disables the specified VLAN interface.

Table 5-3. Supported Dell_Networking OS API Functions in PERL Scripts (continued)

F10Ping	(ipAddress)	Pings (via ICMP) an IP address from the switch.
F10ShowArpTbl	None	Returns the table of learned ARP entries.
F10ShowBGPNeighbors	None	Returns list of BGP neighbors.
F10ShowBGPRoute	None	Returns the table of BGP-learned routes.
F10ShowBGPSummary	None	Returns summary information on BGP sessions.
F10ShowBootVar	None	Returns system boot variables.
F10ShowEnvironment	None	Returns environment-monitoring variable values.
F10ShowIntBrief	None	Returns brief interface status (up/down/admin up/admin down) of all interfaces.
F10ShowIntBriefLag	None	Returns brief interface status (up/down/ admin up/down) of all port-channel interfaces.
F10ShowIntBriefMan	None	Returns brief interface status (up/down/ admin up/down) of all management interfaces.
F10ShowIntBriefPhy	None	Returns brief interface status (up/down/ admin up/down) of all physical interfaces.
F10ShowIntBriefVlan	None	Returns brief interface status (up/down/ admin up/down) of all VLAN interfaces.
F10ShowIPRoute	None	Returns routing table information.
F10ShowISISNeighbors	None	Returns list of ISIS neighbors.
F10ShowISISRoute	None	Returns the table of ISIS-learned routes.
F10ShowLagIntStatus	(lagId)	Returns the detailed status of a specified port-channel interface.
F10ShowLagIntVlanMembers	(lagId)	Returns information on VLAN membership for a specified port-channel interface.
F10ShowLog	None	Returns the switch log buffer.
F10ShowMacAddrTbl	None	Returns the table of learned MAC addresses.
F10ShowMem	(lagId)	Returns switch memory usage.
F10ShowOSPFNeighbors	None	Returns list of OSPF neighbors.
F10ShowOSPFRoute	None	Returns the table of OSPF-learned routes.
F10ShowPhyIntBand	(stackUnitNum, portId)	Returns in/out bandwidth average for a specified port.
F10ShowPhyIntStatus	(stackUnitNum, portId)	Returns the detailed status of a specified physical interface.
F10ShowPhyIntVlanMembers	(stackUnitNum, portId)	Returns information on VLAN membership for a specified physical interface.
F10ShowProcCpu	None	Returns switch CPU usage and running processes.
F10ShowRun	None	Returns the running configuration (in memory).
F10ShowVer	None	Returns software version information.
F10ShowVlan	None	Returns the show vlan output for all VLANs.
F10ShowVlanId	(vlanId)	Returns the show vlan output for a specific vlan.
F10ShowVlanIntStatus	(vlanId)	Returns the detailed status of a specified VLAN interface.
F10ShowVrrp	None	Returns the full VRRP status output.

Table 5-3. Supported Dell_Networking OS API Functions in PERL Scripts (continued)

F10ShowVrrpBrief	None	Returns a brief VRRP session summary.
F10Traceroute	(ipAddress, timeout)	Performs a traceroute operation to an IP address from the switch.
F10WriteMem	None	Write the running configuration to the startup configuration file.

Running a PERL API Script

When you run a PERL script that invokes the API functions in [Supported UNIX Utilities](#), log on credentials are read from the smartutils.cfg file and a telnet session opens on the switch in which function calls are executed in the Dell Networking OS CLI. The script closes the telnet session after running all the CLI commands.

The smartutils.cfg file is the configuration file used by the RESTAPI Interface package. It contains the user name and passwords required to log on to a switch via telnet and access the CLI to execute the function calls in a PERL API script. The smartutils.cfg file downloads with the Rest API package and is stored at /usr/pkg/scripts/smartutils.



Note: Use the user name and passwords contained in the smartutils.cfg file to log in and run only the scripts created using the PERL, Python, and UNIX APIs described in this chapter. A username used to run scripts cannot contain special characters.

To configure the username and passwords located in the smartutils.cfg file that are used to run PERL API scripts, do the following:

- From a UNIX shell, use the UNIX text editor to open the smartutils.cfg file, enter a user name and password, and save the file.

To run a PERL API script:

- From the Dell Networking OS CLI, use the **script** command as described in [Scheduling Time / Event-based Scripts](#).
- From a UNIX shell, follow the procedure described in [Running a Script from the UNIX Shell](#).

Using the Python API

To create a Python script using the Python API and run the script on a Dell Networking switch, use the information in this section. For information about creating and running a PERL script using the PERL API, refer to [Using the PERL API](#).

Creating a Python API Script

To create a Python script to be run on a Dell Networking switch, use the information in this section. For information about how to run a Python script from the Dell Networking OS CLI, refer to [Scheduling Time / Event-based Scripts](#).

F10SmartUtils.py is the Python API library containing the supported functions (described in [Supported Dell Networking OS API Functions in Python Scripts](#)), which you can use in a Python script to invoke Dell Networking OS operations on a switch. This file is stored at /usr/pkg/scripts/smartutils.

Code Dell Networking OS API functions in a Python script as shown in the following example:

Figure 5-3. Python Script with API function call: Example

```
#!/usr/pkg/bin/python

import sys

sys.path.append('/usr/pkg/scripts/smartutils') <----- Load the Python
API

import F10SmartUtils

def create_vlans(startId,endId):
    for vlanId in range(startId,endId+1):
        result = F10SmartUtils.F10CreateVlanId(vlanId) <----- Invoke a Python API
function
        print result

def main(args):
    try:
        startId = int(args[0])
        endId = int(args[1])
        if(startId<=endId):
            create_vlans(startId, endId)
        else :
            print "Invalid range: startId cannot be larger than
endId",startId,endId
    except ValueError:
        print "Invalid arguments",args

if __name__=="__main__":
    if len(sys.argv)>2:
        main(sys.argv[1:])
    else:
        print "Please supply valid arguments"
        print "createVlans.py <startId> <endId>"
```

[Supported Dell Networking OS API Functions in Python Scripts](#) describes the supported functions and required arguments that you can use in Python scripts running on a Dell Networking switch to connect through a telnet session and gather information or configure parameters through the CLI.

Table 5-4. Supported Dell Networking OS API Functions in Python Scripts

Python API Function	Arguments	Description
F10AddLagIntoVlan	(lagId, vlanId, tagFlag)	Adds a LAG interface to a VLAN as either tagged or untagged. tagFlag values: 1 (tagged) or 0 (untagged).
F10AddPhyIntoVlan	(stackUnitNum, portId, vlanId, tagFlag)	Adds a physical interface to a VLAN as either tagged or untagged. tagFlag values: 1 (tagged) or 0 (untagged).
F10CreateVlanId	(vlanId)	Creates a VLAN on the switch.
F10DeleteVlanId	(vlanId)	Deletes a VLAN on the switch.
F10ExecShowCmd	(command)	Executes a specified show command.
F10MakeLagIntNoShutdown	(lagId)	Enables the specified port channel.
F10MakeLagIntShutdown	(lagId)	Disables the specified port channel.
F10MakeLagIntSwitch	(lagId)	Configures the specified port channel (LAG) as a Layer 2 switchport.
F10MakePhyIntNoShutdown	(stackUnitNum, portId)	Enables the specified port.
F10MakePhyIntShutdown	(stackUnitNum, portId)	Disables the specified port.
F10MakePhyIntSwitch	(stackUnitNum, portId)	Configures the specified port as a Layer 2 switch port.
F10MakeVlanIntNoShutdown	(vlanId)	Enables the specified VLAN interface.
F10MakeVlanIntShutdown	(vlanId)	Disables the specified VLAN interface.
F10Ping	(ipAddress)	Pings (via ICMP) an IP address from the switch.
F10ShowArpTbl	None	Returns the table of learned ARP entries.
F10ShowBGPNeighbors	None	Returns list of BGP neighbors.
F10ShowBGPRoute	None	Returns the table of BGP-learned routes.
F10ShowBGPSummary	None	Returns summary information on BGP sessions.
F10ShowBootVar	None	Returns system boot variables.
F10ShowEnvironment	None	Returns environment-monitoring variable values.
F10ShowIntBrief	None	Returns brief interface status (up/down/admin up/admin down) of all interfaces.
F10ShowIntBriefLag	None	Returns brief interface status (up/down/ admin up/down) of all port-channel interfaces.
F10ShowIntBriefMan	None	Returns brief interface status (up/down/ admin up/down) of all management interfaces.
F10ShowIntBriefPhy	None	Returns brief interface status (up/down/ admin up/down) of all physical interfaces.
F10ShowIntBriefVlan	None	Returns brief interface status (up/down/ admin up/down) of all VLAN interfaces.

Table 5-4. Supported Dell Networking OS API Functions in Python Scripts (continued)

F10ShowIPRoute	None	Returns routing table information.
F10ShowISISNeighbors	None	Returns list of ISIS neighbors.
F10ShowISISRoute	None	Returns the table of ISIS-learned routes.
F10ShowLagIntStatus	(lagId)	Returns the detailed status of a specified port-channel interface.
F10ShowLagIntVlanMembers	(lagId)	Returns information on VLAN membership for a specified port-channel interface.
F10ShowLog	None	Returns the switch log buffer.
F10ShowMacAddrTbl	None	Returns the table of learned MAC addresses.
F10ShowMem	(lagId)	Returns switch memory usage.
F10ShowOSPFNeighbors	None	Returns list of OSPF neighbors.
F10ShowOSPFRoute	None	Returns the table of OSPF-learned routes.
F10ShowPhyIntBand	(stackUnitNum, portId)	Returns in/out bandwidth average for a specified port.
F10ShowPhyIntStatus	(stackUnitNum, portId)	Returns the detailed status of a specified physical interface.
F10ShowPhyIntVlanMembers	(stackUnitNum, portId)	Returns information on VLAN membership for a specified physical interface.
F10ShowProcCpu	None	Returns switch CPU usage and running processes.
F10ShowRun	None	Returns the running configuration (in memory).
F10ShowVer	None	Returns software version information.
F10ShowVlan	None	Returns the show vlan output for all VLANs.
F10ShowVlanId	(vlanId)	Returns the show vlan output for a specific vlan.
F10ShowVlanIntStatus	(vlanId)	Returns the detailed status of a specified VLAN interface.
F10ShowVrrp	None	Returns the full VRRP status output.
F10ShowVrrpBrief	None	Returns a brief VRRP session summary.
F10Traceroute	(ipAddress, timeout)	Performs a traceroute operation to an IP address from the switch.
F10WriteMem	None	Write the running configuration to the startup configuration file.

Running a Python API Script

When you run a Python script that invokes the API functions in [Supported Dell Networking OS API Functions in Python Scripts](#), logon credentials are read from the `smartutils.cfg` file, and a telnet session opens on the switch in which function calls are executed in the Dell Networking OS CLI. The script closes the telnet session after running all the CLI commands.

The `smartutils.cfg` file is the configuration file used by the Programmatic Interface package. It contains the user name and passwords required to log on to a switch via telnet and access the CLI to execute the function calls in a Python API script. The `smartutils.cfg` file downloads with the Programmatic Interface package and is stored at `/usr/pkg/scripts/smartutils`.



Note: Use the user name and passwords contained in the `smartutils.cfg` file to log into and run only the scripts created using the PERL, Python, and UNIX APIs described in this chapter. A username used to run scripts cannot contain special characters.

To configure the username and passwords located in the `smartutils.cfg` file that are used to run Python API scripts, do the following:

- From a UNIX shell, use the UNIX text editor to open the `smartutils.cfg` file, enter a user name and password, and save the file.

To run a Python API script:

- From the Dell Networking OS CLI, use the **script** command described in [Scheduling Time / Event-based Scripts](#).
- From a UNIX shell, follow the procedure described in [Running a Script from the UNIX Shell](#).

Using UNIX Shell Scripting

To create a UNIX script using the UNIX API and run the script on a Dell Networking switch, use the information in this section. For information about creating and running a PERL or Python script using the PERL or Python API, see [Using the PERL API](#) or [Using the Python API](#).

Creating a UNIX API Script

To create a UNIX shell script to run on a Dell Networking switch, use the information in this section.

The `F10SmartScriptUtils.py` file is the main API library script that contains the functions that you can include in a UNIX shell script. The `F10SmartScriptUtils.py` script is stored at `/usr/pkg/scripts/smartutils`. [Supported API Functions in UNIX Shell Scripts](#) describes the Dell Networking OS operations that you can invoke from a UNIX shell script, including the supported functions and required arguments.

[Figure 5-4](#) shows an example of how to write a script in the UNIX shell scripting language. You can store a UNIX shell script anywhere on the switch.

Figure 5-4. Script Written in the UNIX Shell Scripting Language: Example

```
#!/bin/sh

i=$1
while [ $i -le $2 ]
do
    echo $i
    /usr/pkg/bin/python /usr/pkg/scripts/smartutils/F10SmartScriptUtils.py createvlanid $i
    (( i++ ))
done
```

Table 5-5. Supported API Functions in UNIX Shell Scripts

Function	Arguments	Description
addlaginttovlan	lagId, vlanId, tagFlag	Adds a port channel (LAG) to a VLAN. tagFlag values: 1 (tagged) or 0 (untagged).
addphyinttovlan	stackunitNum, portId vlanId, tagFlag	Adds an interface to a VLAN. tagFlag values: 1 (tagged) or 0 (untagged).
createvlanid	vlanId	Creates a VLAN with a specified VLAN ID.
deletevlanid	vlanId	Deletes a VLAN with a specified VLAN ID
makelagintnoshutdown	lagId	Enables the specified port channel.
makelagintshutdown	lagId	Disables the specified port channel.
makelagintswitch	lagId	Configures the specified port channel (LAG) as a Layer 2 switchport.
makephyintnoshutdown	stackUnitNum, portId	Enables the specified port.
makephyintshutdown	stackUnitNum, portId	Disables the specified port.
makephyintswitch	stackunitNum, portId	Configures the specified port as a Layer 2 switchport.
makevlanintnoshutdown	vlanId	Enables the specified VLAN interface.
makevlanintshutdown	vlanId	Disables the specified VLAN interface.
ping	ipAddress	Pings (via ICMP) an IP address from the switch.
showarptbl	None	Returns the table of learned ARP addresses.
showbgpneighbors	None	Returns detailed BGP neighbor information.
showbgproute	None	Returns BGP-learned routes.
showbgpsummary	None	Returns BGP peer summary and status.
showbootvar	None	Returns system boot variables.
showcmd	command	Executes a specified show command.
showenvironment	None	Returns environment-monitoring variable values.
showipintbrief	None	Returns full interface list with up/down status.
showipintbriefflag	None	Returns brief interface status (up/down/ admin up/down) of all port-channel interfaces.
showipintbriefman	None	Returns brief interface status (up/down/ admin up/down) of all management interfaces.

Table 5-5. Supported API Functions in UNIX Shell Scripts (continued)

showipintbriefphy	None	Returns brief interface status (up/down/ admin up/down) of all physical interfaces.
showipintbriefvlan	None	Returns brief interface status (up/down/ admin up/down) of all VLAN interfaces.
showiproute	None	Returns switch routing table.
showisisneighbors	None	Returns detailed ISIS neighbor information.
showisisroute	None	Returns ISIS-learned routes.
showlagintstatus	lagId	Returns detailed status information for a specified port channel.
showlagintvlanmembers	lagId	Returns VLAN membership of a specified port channel.
showlog	None	Returns system log output.
showmacaddrtbl	None	Returns the table of learned MAC addresses.
showmem	lagId	Returns switch memory usage.
showospfneighbors	None	Returns detailed OSPF neighbor information.
showospfroute	None	Returns OSPF-learned routes.
showphyintband	stackunitNum, portId	Returns in/out bandwidth average for a specified port.
showphyintstatus	stackunitNum, portId	Returns detailed status information for a specified port
showphyintvlanmembers	stackunitNum, portId	Returns VLAN membership of a specified port.
showproccpu	None	Returns switch CPU usage and running processes.
showrun	None	Returns the running configuration (in memory).
showver	None	Returns software version information.
showvlan	None	Returns information on all VLANs, including membership.
showvlanid	vlanId	Returns detailed interface information for a specified VLAN.
showvlanintstatus	vlanId	Returns VLAN interface status.
showvrrp	None	Returns the full VRRP status output.
showvrrpbrief	None	Returns a brief VRRP session summary.
traceroute	ipAddress, timeout	Performs a traceroute operation to an IP address from the switch.
writemem	None	Write the running configuration to the startup configuration file.

Running a UNIX API Script

When you run a UNIX shell script that invokes the API functions in [Supported API Functions in UNIX Shell Scripts](#), logon credentials are read from the smartutils.cfg file and a telnet session opens on the switch in which function calls are executed in the Dell Networking OS CLI. The script closes the telnet session after running all the CLI commands.

The smartutils.cfg configuration file is used by the Programmatic Interface package. It contains the user name and passwords required to log on to a switch via telnet and access the CLI to execute the function calls in a UNIX API script. The smartutils.cfg file downloads with the Programmatic Interface package and is stored at /usr/pkg/scripts/smartutils.



Note: Use the user name and passwords contained in the smartutils.cfg file to log into and run only the scripts created using the PERL, Python, and UNIX APIs described in this chapter. A username used to run scripts cannot contain special characters.

To configure the username and passwords located in the smartutils.cfg file that are used to run UNIX API scripts, do one of the following:

- From a UNIX shell, use the UNIX text editor to open the smartutils.cfg file, enter a user name and password, and save the file.

To run a UNIX API script:

- From the Dell Networking OS CLI, use the **script** command described in [Scheduling Time / Event-based Scripts](#).
- From a UNIX shell, follow the procedure described in [Running a Script from the UNIX Shell](#).

Running Scripts with User Privileges

Use these scripts to administer any Expect, PERL, Python, Tcl, UNIX and ZSH shell scripts stored on the switch from the Dell Networking OS CLI.

To apply the associated read-write privileges while running a script from the Dell Networking OS CLI, specify an optional username (refer to [Creating a User Name and Password for Smart Scripting](#)). If you do not specify a user name, the script runs with the privileges of the configured user.

To run a PERL, Python, or UNIX script from the Dell Networking OS CLI, use the **script** command. Enter the script name and directory path to start the script. The script can invoke any of the supported UNIX utilities listed in [Supported UNIX Utilities](#). Enter the command multiple times to run more than one script at the same time.

Command Syntax	Command Mode	Task
script [username <i>name</i>] <i>script-path</i> [<i>script-parameter</i> <i>script-parameter</i> ...]	CONFIGURATION	Run an installed script; for examples refer to Figure 5-2 . For <i>script-path</i> , enter the directory path and filename. (Optional) For username <i>name</i> , enter the user name whose read-write privileges are applied when the script runs. A username used to run scripts cannot contain special characters. (Optional) For <i>script-parameter</i> , enter the values of up to three parameters to be applied when the script runs. Enter a blank space between parameter values; for example: script username admin /f10/flash/createVlans.py 1 2

Smart Scripting CLI

Overview

Smart Scripting is supported on the **S4810**, **S4820T**, **S6000**, **Z9000** and **MXL** switch platforms.

Commands

- mount nfs
- package install
- package uninstall
- script (run)
- script (stop/resume/clear/kill/unschedule)
- script execute (EXEC mode)
- script execute (CONFIGURATION mode)
- script execute triggered-by
- script get
- script list execute
- script path
- script remove
- script trigger-event
- show packages
- show script
- start shell
- switch script limit
- username

mount nfs

S4810 S4820T
S6000
 Z9000, MXL
 Switch

Share the network file system to be used by the local Dell Networking OS file system.

Syntax

mount nfs *nfs-server-ip*: **remote_dir** **mount_name** [**username** *username* **password** *password*]

To unmount the network file system, use the **no mount** command.

Parameters

remote_dir	Enter the directory path where the network file system will be mounted.
mount_name	
username <i>username</i>	(Optional) Enter the keyword username followed by a text string up to 40 characters long as the user name.
password <i>password</i>	(Optional) Enter the keyword password followed by a text string up to 40 characters long as the password.

Defaults

None

Command Modes

CONFIGURATION

Command History

Version 9.3(0.0)	Introduced on the S6000.
Version 9.2(0.0)	Introduced on the S4810, S4820T, Z9000, and MXL Switch.

Related commands

script path	Configure the search path to point the local file share of NFS and to run the scripts in the NFS, as unnamed.
-----------------------------	---

Usage Information

- All the mount points are maintained in the /f10/mnt/nfs folder.
- Only the relative path (mount point name) is acceptable. If the path entered is either complete or absolute, an error are thrown.
- If the mount point exists already under the f10/mnt, you can re-use it or it is created under /f10/mnt/nfs and used.

package install

S4810 **S4820T**

S6000 Z9000

MXL Switch

Install the Smart Scripting package. This command downloads the package from the specified location and installs it in the internal flash memory on a switch.

Syntax

package install *location*

Parameters

<i>location</i>	Enter the location from where you will download and install an Open Automation package, where <i>location</i> is one of the following values: <ul style="list-style-type: none">From the local flash: flash://filenameFrom an FTP server: ftp://userid:password@host-ipaddress/filepathFrom a TFTP server: tftp://host-ipaddress/filepathFrom a NFS mount server: nfsmount://filepath
-----------------	--

Defaults

None

Command Modes

EXEC Privilege

Command History

Version 9.2(0.0)	Introduced on the MXL switch.
Version 9.0.2.0.	Introduced on the S6000.
Version 9.0.0.0	Introduced on the Z9000.
Version 8.3.19.0	Introduced on the S4820T.
Version 8.3.8.0	Introduced on the S4810.

Usage Information

Because installing an Open Automation package may take time, the installation is performed in the background when the download finishes. A message displays on the console when the installation is complete.

To follow the progress of a package installation, use the [show packages](#) command.

package uninstall

S55 **S60**

S4810 **S4820T**

S6000 Z9000

MXL Switch

Remove an installed Open Automation package, such as Smart Scripting, from the system.

Syntax

package uninstall *package-name*

Parameters

<i>package-name</i>	Enter the name of an Open Automation automation package, exactly as it appears in the show packages list.
---------------------	---

Defaults

None

Command Modes

EXEC Privilege

Command History

Version 9.2(0.0)	Introduced on the MXL switch.
Version 9.0.2.0.	Introduced on the S6000.
Version 9.0.0.0	Introduced on the Z9000.
Version 8.3.19.0	Introduced on the S4820T.
Version 8.3.8.0	Introduced on the S4810.

Usage Information

When you uninstall an Open Automation package, it is removed from the local flash memory.



Caution: Before you uninstall the Smart Scripting package, first stop all scripts that are currently running using the **no script script-name** command. You must also manually stop the http-server daemon by executing the **no http-server {http | secure-http}**.

To follow the progress when uninstalling an Open Automation package installation, use the [show packages](#) command.

Related commands

show packages	Display all Open Automation packages installed on the switch.
-------------------------------	---

script (run)

S55 S60

S4810 S4820T

S6000 Z9000
MXL Switch

Run a Expect, Perl, Python, Tcl, UNIX and ZSH shell script from the Dell Networking CLI.

Syntax

script [**username name**] *script-name* [*script-parameter script-parameter ...*]

Parameters

username name	(Optional) Enter the user name whose read-write privileges is applied when the script runs. A username used to run scripts cannot contain special characters.
<i>script-name</i>	Enter the name of the script to run, including the directory path and filename; for example: Perl script: /usr/pkg/scripts/sample_scripts/cmd-server.pl Python script: /usr/pkg/scripts/sample_scripts/DisplayAlarms.py UNIX shell script: /usr/pkg/home/admin/test.sh
<i>script-parameter</i>	(Optional) Enter the values of up to three parameters to be applied when the script is run. Enter a blank space between parameter values. For example: script username admin /f10/flash/createVlans.py 1 2

Defaults

None

Command Modes

CONFIGURATION

Command History

Version 9.2(0.0)	Introduced on the MXL switch.
Version 9.0.2.0.	Introduced on the S6000.
Version 9.0.0.0	Introduced on the Z9000.
Version 8.3.19.0	Introduced on the S4820T.
Version 8.3.8.0	Introduced on the S4810.

Usage Information

To run more than one scripts at the same time, you can use the [script \(run\)](#) command multiple times; for example:

```
Dell(conf)#script username root /usr/pkg/scripts/sample_scripts/  
DisplayAlarms.py  
Dell(conf)#script username root /usr/pkg/bin/python /usr/pkg/scripts/  
VSNAgent/Xen/hpAgtMain.py
```

When you run a script from the Dell Networking OS CLI with the [script \(run\)](#) command, you can specify an optional user name to apply the read-write privileges assigned to the user name when the script is run. To configure the username and password, use the [username](#) command. If you do not specify a user name with the [script \(run\)](#) command, the script is run with the privileges of the current user.

For information about how to run a script directly from a UNIX shell, refer to [Running a Script from the UNIX Shell](#).

To stop a running script, use the **no script** *script-name* command

To display the scripts that are currently running, including the scripts you have stopped, use the [show running-config | grep](#) command.

script (stop/resume/clear/kill/unschedule)

S4810 **S4820T**

S6000 Z9000
MXL Switch

Stop, resume, clear, kill, or unschedule a Expect, Perl, Python, Tcl, UNIX and ZSH shell script from the Dell Networking OS CLI.

Syntax

```
script { stop | resume | clear | kill | unschedule } { script-name script-name | job-id job-id | all }
```

Parameters

stop	Enter the keyword stop to stop a script from being run.
resume	Enter the keyword resume to restart a script that has been stopped.
clear	Enter the keyword clear to restart the supervised scripts that has been blocked.
kill	Enter the keyword kill to end a script from executing.
unschedule	Enter the keyword unschedule to delete a script that was scheduled in EXEC mode.
<i>script-name</i>	Enter the keywords script-name followed by the name of the script to be stopped, resumed, cleared, killed or unscheduled.

	<i>job-id</i>	Enter the keywords job-id followed by the job identifier of the specific job to be stopped, resumed, cleared, killed or unscheduled.
	<i>all</i>	Enter the keyword all to stop, resume, clear, kill or unschedule all scripts.
Defaults	None	
Command Modes	EXEC	
Command History	Version 9.3(0.0)	Introduced on the S6000.
	Version 9.2(0.0)	Introduced on the S4810, S4820T, Z9000, and MXL Switch.
Usage Information		Use the keyword unschedule only on scripts that are not currently running and that were scheduled using the script execute command in EXEC mode.

script execute (EXEC mode)

S4810 S4820T

Z9000 **S6000**
MXL Switch

Schedule when to trigger scripts to execute.

Syntax

script execute *script-name* **start** {**now** | *time-date* | *time*} [**stop** {**at** *time-date* | **after** *time*}] [**args** *arguments*] [**username** *username*] [**bg**]

To cancel, use the **script unschedule** command.

Parameters

<i>script-name</i>	Enter the name of the script to be scheduled for execution.
start now	Enter the keywords start now to begin executing the script.
start <i>time-date</i>	Enter the keywords start time-date with the time and date in HH:MM format to begin executing the script at a specific time. The date can be the present or a future date.
start <i>time</i>	Enter the keywords start with the time in HH:MM format to begin executing the script after a set time.
stop at <i>time-date</i>	Enter the keywords stop at with the time and date in HH:MM-MM/DD/YY format to stop executing the script. The date must be a future date.
stop after <i>time</i>	Enter the keywords stop after followed by the time in HH:MM format to stop executing the script.
args <i>arguments</i>	Enter the keyword args followed by the arguments to be scripted. The maximum length is 64 characters. The arguments can be any number of words within quotes (“”) and separated by a space.
username <i>username</i>	Enter the username to be used when the script executes. The maximum length is 16 characters. The default username is the user configuring the CLI.
bg	Enter the keyword bg to schedule scripts to run in the background.

Defaults	The script runs in the foreground.
Command Modes	EXEC
Command History	Version 9.3(0.0) Introduced on the S6000.
	Version 9.2(0.0) Introduced on the S4810, S4820T, Z9000, and MXL Switch.
Usage Information	All times selected follow the system time of the switch. The maximum number of scripts to configure is 100. An error message displays when you exceed the limit.

script execute watch (EXEC mode)

S4810 S4820T
Z9000 **S6000**
MXL Switch

Monitor or supervise the scheduled scripts.

Syntax	script execute <i>script-name</i> watch start { <i>now</i> <i>time-date</i> <i>time</i> } [args <i>arguments</i>] [username <i>username</i>] [bg]
Parameters	<i>script-name</i> Enter the name of the script to be scheduled for execution.
	<i>watch</i> Enter the keyword watch to monitor the script.
	start now Enter the keywords start now to begin executing the script.
	start <i>time-date</i> Enter the keywords start time-date with the time and date in HH:MM-MM/DD/YY format to begin executing the script at a specific time. The date can be the present or a future date.
	start <i>time</i> Enter the keywords start with the time in HH:MM format to begin executing the script after a set time.
	args <i>arguments</i> Enter the keyword args followed by the arguments to be scripted. The maximum length is 64 characters. The arguments can be any number of words within quotes (“”) and separated by a space.
	username <i>username</i> Enter the username to be used when the script executes. The maximum length is 16 characters. The default username is the user configuring the CLI.
	bg Enter the keyword bg to schedule scripts to run in the background.
Defaults	The script runs in the foreground.
Command Modes	EXEC
Command History	Version 9.3(0.0) Introduced on the S6000.
	Version 9.2(0.0) Introduced on the S4810, S4820T, Z9000, and MXL Switch.

script execute (CONFIGURATION mode)

Schedule when to trigger script executes.

S4810 S4820T

S6000 Z9000
MXL Switch

Syntax

```
script execute script-name start {now | time-date | time} [repeat {days day |
minutes minutes | weekdays weekday}] [stop {at time-date | after time}] | watch
[start {now | date-time | time} repeat {weekdays day | days day | minutes minutes}]
[args arguments] [username username]
```

To cancel the script execution, use the **no script execute** command.

Parameters

<i>script-name</i>	Enter the name of the script to be scheduled for execution.
start now	Enter the keywords start now to begin executing the script.
start <i>time-date</i>	Enter the keyword start with the time and date in HH:MM-MM/DD/YY format to begin executing the script at a specific time. The date can be the present or a future date.
start <i>time</i>	Enter the keyword start with the time in HH:MM format to begin executing the script after a set time.
repeat weekdays <i>weekday</i>	Enter the keywords repeat weekdays to schedule the script to repeat executing on specific days of the week. Select one of the following values: mon, tue, wed, thu, fri, sat, sun, and all.
repeat days <i>day</i>	Enter the keywords repeat days followed by the day of the month to schedule how often to repeat the script execution. The range is from 1 to 31. Select one value only. For example, 1 for every first day of the month, 10 for every 10th day of the month.
repeat minutes <i>minutes</i>	Enter the keywords repeat minutes followed by the number of minutes to schedule how often to repeat the script execution. The range is from 1 to 1440. Select one value only. For example, select 20 to repeat every 20 minutes, 60 for repeating once every hour.
stop at <i>time/date</i>	(Optional) Enter the keywords stop at with the time and date in HH:MM-MM/DD/YY format to stop executing the script. The date must be a future date.
stop after <i>time</i>	(Optional) Enter the keywords stop after followed by the time in HH:MM format to indicate the time after which the script stops executing. For example, “stop after 00:30” indicates to stop the script execution 30 minutes after the start time.
watch	Enter the keyword watch to monitor the script.
args <i>arguments</i>	Enter the keyword args followed by the arguments to be scripted. The maximum length is 64 characters. The arguments can be any number of words within quotes (“”) and separated by a space.
username <i>username</i>	Enter the username to be used when the script executes. The maximum length is 16 characters. The default username is the user configuring the CLI.

The script runs in the background.

Command Modes

CONFIGURATION

Command History

Version 9.3(0.0) Introduced on the S6000.

Version 9.2(0.0) Introduced on the S4810, S4820T, Z9000, and MXL Switch.

Usage Information

All times selected follow the system time of the switch. The maximum number of scripts to execute is 100. An error message displays when you exceed the limit.

If the supervised scripts exit (normally or in an unexpected manner):

- Within 10 minutes from the start, the supervised script(s) is restarted (three retries).
- More than three times within 10 minutes, the script is in **Blocked** state, which you can reset using the **script clear** command.
- TACACS/RADIUS user(s) are not supported.
- Only the person who configured the scheduled scripts (or a higher privileged user) can change them.

script execute triggered-by

S4810 S4820T**S6000**Z9000, MXL
Switch

Schedule the script to execute on the occurrence of the configured event.

Syntax

script execute *scriptname* [**concurrent**] **triggered-by** *event-name* [**args** *arguments*] [**username** *username*]

To cancel the executing the script by an event, use the **no script execute triggered-by** command.

Parameters

<i>script-name</i>	Enter the name of the script to be scheduled for execution.
concurrent	Enter the keyword concurrent to execute the concurrent instances of the script on event occurrence.
triggered-by <i>event-name</i>	Enter the keywords triggered-by followed by the event name to associate a script to a defined trigger event.
args <i>arguments</i>	Enter the keyword args followed by the arguments to be scripted. The maximum length is 64 characters. The arguments can be any number of words within quotes (“”) and separated by a space.
username <i>username</i>	Enter the username to be used when the script executes. The maximum length is 16 characters. The default username is the user configuring the CLI.

Defaults

The script runs in the background and as Singleton.

Command Modes

CONFIGURATION

Command History

 Version 9.3(0.0) Introduced on the S6000.

 Version 9.2(0.0) Introduced on the S4810, S4820T, Z9000, and MXL Switch.

Usage Information

The trigger events can be associated with two types of scripts — Singleton and Concurrent.

For singleton scripts, the following operations are performed:

- When the script is running, all the event occurrences are notified by the SIGUSR1 signal and a new instance of the script is not spawned.
- The script uses the **getcurrentvt** utility to retrieve the event name.
- The script uses the **getnextevt** utility to retrieve the last but one event name.
- The script, which contains the script name, args, and username, can be associated with more than one trigger event.
- When multiple users configure a singleton script, the scripts runs separately in each user context.

For concurrent scripts, the following operations are performed:

- A new instance of the script is spawned at every occurrence of the event.
- The maximum instances are restricted to 10.
- Only one trigger can be associated with the script, which contains the script name, args, and username.

script get

S4810 **S4820T**
S6000 Z9000
 MXL Switch

Copy a script to a switch.

Syntax
script get url
Parameters

url	Enter the keyword url followed by the URL location of the script to download to a switch.
------------	--

Defaults

None

Command Modes

EXEC

Command History

 Version 9.3(0.0) Introduced on the S6000.

 Version 9.2(0.0) Introduced on the S4810, S4820T, Z9000, and MXL Switch.

Usage Information

The following formats are supported: FLASH,FTP, TFTP, HTTP, and SCP. To retrieve the script files, use the following formats:

- flash: Copy from the flash file system (flash://filepath)
- ftp: Copy from the remote file system (ftp://userid:password@hostip/filepath)
- http: Copy from the remote file system (http://hostip/filepath)
- scp: Copy from the remote file system (scp://userid:password@hostip/filepath)
- ftp: Copy from the remote file system (tftp://hostip/_filepath)

The downloaded files are stored into a dedicated folder (**/usr/pkg/ss-scripts**).



Note: In case of stack, the scripts from the ss-scripts will be synchronized across the stack for every one hour. When the stack forms first, the script synchronization from the master to members will happen only after 10 minutes.

script list execute

S4810 **S4820T**

S6000 Z9000
MXL Switch

Configure a list of scripts to run when the switch is rebooted. The scripts may be run before or after loading the configuration file.

Syntax

script list execute *list-name* {**on-boot** | **networkup**}

To delete the list of scripts for an event, use the **no script list execute** command.

Parameters

<i>list-name</i>	Enter the file name that contains the list of scripts to be executed and the sequence in which the scripts will execute.
on-boot	Enter the keywords on-boot to execute the list of scripts before the configuration file is loaded.
network-up	Enter the keywords network-up to execute the list of scripts after the configuration file is loaded.

Defaults

None

Command Modes

CONFIGURATION

Command History

Version 9.3(0.0)	Introduced on the S6000.
Version 9.2(0.0)	Introduced on the S4810, S4820T, Z9000, and MXL Switch.

Usage Information

The maximum number of lists to configure is two. In Normal Reload mode, the on-boot list executes before applying configurations; it is not executed in BMP mode. The network list is executed after applying configurations; it executes in BMP and Normal Reload mode.

File contents used in the list name are in the following format:

```
#! list_name
seq 1 script <script_name> [args]
seq 2 script <script_name> [args]
seq 3 script <script_name> [args]
seq 4 script <script_name> [args]
seq 5 script <script_name> [args]
```

script path

S4810 S4820T

S6000 Z9000
MXL Switch

Configure the path for the script on the switch.

Syntax

script path *path-name*

To remove the path for the script, use the **no script path** command.

Parameters

<i>path-name</i>	Enter the full path of the location of the script. Specify multiple paths using a colon (:). When initialized, the default path is /user/pkg/ss-scripts .
------------------	--

Defaults

None

Command Modes

CONFIGURATION

Command History

Version 9.3(0.0)	Introduced on the S6000.
Version 9.2(0.0)	Introduced on the S4810, S4820T, Z9000, and MXL Switch.

Usage Information

The path can contain a network file system (NFS) mounted directory (defined in the [mount nfs](#) CLI command). The path is added to a script search list allowing the system to search all locations for the script name. If the script is in multiple locations, the system uses the first instance of the script found.

Related commands

mount nfs	Set up the folders in the NFS mounted directory.
---------------------------	--

script remove

S4810 S4820T

S6000 Z9000
MXL Switch

Remove a script from a switch.

Syntax

script remove { *file-name* | **all** }

Parameters	<i>file-name</i>	Enter the file name of the script to be removed from the switch.
	all	Enter the keywords all to remove all files from the dedicated folder on the switch.
Defaults	None	
Command Modes	EXEC	
Command History	Version 9.3(0.0)	Introduced on the S6000.
	Version 9.2(0.0)	Introduced on the S4810, S4820T, Z9000, and MXL Switch.
Usage Information	The files are removed from the dedicated folder <code>/usr/pkg/ss-scripts</code> on the switch.	

script trigger-event

S4810 S4820T

S6000 Z9000
MXL Switch

Configure the event that causes the script to execute.

Syntax

script trigger-event *event-name* {**log-event** {*tag tag*} | **severity** *severity level* | **time-event** *time* {*day* | *date*} | **cpu-usage** *percentage* | **mem-usage** *percentage*}

To delete the trigger event, use the **no script trigger-event** command.

Parameters	event-name	Enter the name of the script event to be triggered.
	log-event { <i>tag tag</i> }	Enter the keywords log-event tag followed by the pattern (tag) in the syslog message to define the trigger event based on the pattern (tag) in the syslog. A maximum three tags can be given. Each tag must be separated by spaces “ ”.
	log-event severity <i>severity level</i>	Enter the keywords log-event severity followed by the severity level to define a trigger even based on the syslog message severity level. Valid message security levels are from 1 to 6.
	time-event <i>time day</i>	Enter the keywords time-event followed by the time in HH:MM (hour-minute) format and then the day of the week as <i>mon, tue, wed, thu, fri, sat, sun, and all</i> to define the trigger event based on the time and day of the week. Scripts associated with this event runs on a particular day at the specified time.
	time-event <i>time date</i>	Enter the keywords time-event followed by the time in HH:MM format and then the date in MM/DD/YY format. The associated script is triggered when the event occurs.
	cpu-usage <i>percentage</i>	Enter the keywords cpu-usage followed by a percentage value between 20 and 90 to define the trigger event based on CPU usage.
	mem-usage <i>percentage</i>	Enter the keywords mem-usage followed by a percentage value between 20 and 90 to define the trigger event based on memory usage.

Defaults	None
Command Modes	CONFIGURATION
Command History	<hr/> Version 9.3(0.0) Introduced on the S6000. <hr/> Version 9.2(0.0) Introduced on the S4810, S4820T, Z9000 and MXL Switch. <hr/>
Usage Information	<p>The maximum number of events you can configure is 20. The maximum number of allowed tag patterns in a log event trigger is three. The tags of the log events must be an exact comparison. The three tags work in a logical OR operation format (match with any one of the tags occurs).</p> <p> Note: Before deleting a trigger event, the script <i>must</i> be unlinked from the event.</p>
Related Commands	<hr/> script execute triggered-by Configure the event that causes the script to execute. <hr/>
Example	<p>The following examples define triggers for the subsequent log messages: link down pattern, failure pattern, severity 1 logs, and 85% memory consumption.</p>
Example “link down” pattern in log messages	Dell(conf)# script trigger-event ma_event log-event tag "Admin state to down: Ma"
Example “failure” pattern in log messages	Dell(conf)# script trigger-event fail_event log-event tag "failure"
Example severity 1 log messages	Dell(conf)# script trigger-event severity1_event log-event severity 1
Example 85% memory consumption	Dell(conf)# script trigger-event memory_event mem-usage 85

show packages

S4810 S4820T
S6000 Z9000
 MXL Switch

Display the installed Open Automation packages, including version number and contents.

Syntax	show packages [system]	
Parameters	system	(Optional) Enter the keyword brief to display system information about the package, version, and status of the package in all stack-units. Note: This option is only available on switches that allow stacking.
Defaults	None	

Command Modes

EXEC

EXEC Privilege

Command History

Version 9.2(0.0)	Introduced on the MXL Switch.
Version 9.0.2.0	Introduced on the S6000.
Version 9.0.0.0	Introduced on the Z9000.
Version 8.3.19.0	Introduced on the S4820T.
Version 8.3.8.0	Introduced on the S4810.

Usage Information

In a stack setup, Dell Networking OS automatically synchronizes the software version installed on the master stack unit with the other switches in the stack. To view the installed software versions on the stack units, use the **show packages system** command.

Example (show packages)

```
Dell# show packages
* Package Name: SMARTSCRIPTS Version: 2.0.0
  Python 2.6.5
  Perl 5.8.8
    Data::Dumper 2.126
    Class::MethodMaker 2.16
    ExtUtils::MakeMaker 6.56
    XML::NamespaceSupport 1.11
    XML::SAX 0.96
    XML::LibXML 1.70
    Compress::Raw::Bzip2 2.027
    Compress::Raw::Zlib 2.027
    IO::Compress 2.027
    URI 1.54
    HTML::Tagset 3.20
    HTML::Parser 3.65
    LWP 5.836
    Net::Telnet 3.03
    OSSP::uuid 1.0602
    UUID 0.02
    version 0.82
    Class::Inspector 1.24
    Task::Weaken 1.03
    Algorithm::Diff 1.1902
    Text::Diff 1.37
    SOAP::Lite 0.712
    Crypt::SSLeay 0.57
    URI::urn::uuid 0.03
    UUID 0.03
    Crypt::SSLeay 0.57
    Net::SNMP 6.0.0
    Net::Telnet::Cisco 1.10

HTTP Server
  mini_httpd 1.19
  Perl and Python function library for Force10 SmartScripts
  smartutils 2.0.0
  WebConnect Web UI and CGI scripts
  htdocs 2.0.0
```

Example (show packages system)

In the following example, Unit 0 is the master stack unit and Unit 1 is the standby unit.

```
Dell#show packages system

Package Information
-----
Unit Package Name          Version      Status
-----
  0 SMARTSCRIPTS           2.9.9.2     Installed
    nano                   2.2.6nb1    Installing
  1 SMARTSCRIPTS           2.9.9.2     Installing
    nano                   2.2.6nb1    Installing
  2 not present
  3 not present
  4 not present
  5 not present
  6 not present
  7 not present
  8 not present
  9 not present
 10 not present
 11 not present
```

show script

S4810 S4820T**S6000** Z9000
MXL Switch

Display the stored, scheduled, and running scripts.

Syntax

show script {file | process | watch} | [detail]

Parameters

file	Enter the keyword file to list the stored files in the script path. Enter the optional keyword detail to show detailed output of the scripts including job-id, script type, and script status.
process	Enter the keyword process to list the scripts that are scheduled or running. Enter the optional keyword detail to show detailed output of the relevant arguments that are scheduled or running.
watch	Enter the keyword watch to list the supervised scripts and their relevant details.
detail	(Optional) Enter the keyword detail to show detailed output of the file including CPU %, memory %, next scheduled time, and any script name or relevant arguments.

Defaults

None

Command Modes

EXEC Privilege

Command History

Version 9.3(0.0)	Introduced on the S6000.
Version 9.2(0.0)	Introduced on the S4810, S4820T, Z9000, and MXL Switch.

start shell

S4810 **S4820T**

S6000 Z9000

MXL Switch

Start a NetBSD UNIX shell.

Syntax **start shell**

Defaults None

Command Modes EXEC Privilege

Command History

Version 9.2(0.0) Introduced on the MXL Switch.

Version 9.0.2.0 Introduced on the S6000.

Version 9.0.0.0 Introduced on the Z9000.

Version 8.3.19.0 Introduced on the S4820T.

Version 8.3.8.0 Introduced on the S4810.

Usage Information

You must start an NetBSD shell on a switch before you can enter UNIX commands (Table 5-2) or run a script directly from the shell to invoke Dell Networking OS operations (refer to [Running a Script from the UNIX Shell](#)).

After you start a shell, you are prompted to enter a user name and password.

Related commands

show packages Display all Open Automation packages installed on the switch.

switch script limit

S4810 **S4820T**

S6000 Z9000

MXL Switch

To control the script that is running based on CPU, memory, or disk IO usage, use the switch script limit.

Syntax **switch script limit {cpu percentage mem percentage disk percentage}**

To return to the default value, use the **no switch script limit** command.

Parameters

cpu percentage Enter the keyword **cpu** and the maximum percentage limit to suspend and hold scripts for execution. The range is from 20 to 90 percent.

mem percentage Enter the keyword **mem** and the maximum percentage limit to suspend and hold scripts for execution. The range is from 20 to 90 percent.

disk percentage Enter the keyword **disk** and the maximum percentage limit to suspend and hold scripts for execution. The range is from 20 to 90 percent. Only the /usr/pkg is monitored.

Defaults	None
Command Modes	CONFIGURATION
Command History	Version 9.3(0.0) Introduced on the S6000.
	Version 9.2(0.0) Introduced on the S4810, S4820T, Z9000 and MXL Switch.
Usage Information	When any maximum percentage limit is reached, all running scripts are suspended; additional scripts are not be scheduled. Scripts begin running when any of the resource limits reaches 10% less than the configured value. Details are entered into the syslog.

username

S4810 **S4820T**

S6000 Z9000
MXL Switch

Configure an additional user name and password to be used only to run scripts on a switch. The user name and password are used to log in to a UNIX shell and apply the read-write privileges assigned to the user name when a script is run.

Syntax **username** *name* **password** *password*

To remove the user name and password, use the **no username** command.

Defaults none

Parameters	<i>name</i>	Enter a username to access the UNIX shell. The user name must be less than 16 characters to satisfy the BSD UNIX login requirements. A username used to run scripts cannot contain special characters.
	password <i>password</i>	Enter a password to access the UNIX shell.

Command Modes CONFIGURATION

Command History	Version 9.2(0.0)	Introduced on the MXL Switch.
	Version 9.0.2.0	Introduced on the S6000.
	Version 9.0.0.0	Introduced on the Z9000.
	Version 8.3.19.0	Introduced on the S4820T.
	Version 8.3.8.0	Introduced on the S4810.

Usage Information When you run a script from the Dell Networking OS CLI with the [script \(run\)](#) command, you can specify an optional user name to apply the read-write privileges assigned to the user name when the script is run (see [Scheduling Time / Event-based Scripts](#)).

Virtual Server Networking

[Virtual Server Networking](#) is supported on platforms: S4810, S4820T, and MXL switch.

As a part of the Open Automation package, Virtual Switch Networking (VSN) provides real-time communication between the Dell Network fabric and virtual servers to automate network management and configuration tasks throughout the data center. VSN provides a closed-loop provisioning system to enable, for example, the automatic re-provisioning of VLANs and port profiles across multiple switches simultaneously, thereby increasing employee productivity and minimizing human error.

Because Open Automation supports hypervisors from multiple vendors, data center managers can use a single mechanism to simultaneously support multiple hypervisors and their current management tools.

VSN is installed as a self-contained package, and requires the [Smart Scripting](#) package.



Note: VSN is not supported in stacked configurations; it is supported only on standalone switches.

This chapter includes the following:

- [Hypervisor Modes](#)
- [VLAN configuration](#)
- [Installing VSN](#)
- [Enabling VSN in a Hypervisor Session](#)
- [Running VSN Scripts](#)
- [Stopping a Hypervisor Session](#)
- [Uninstalling VSN](#)
- [Viewing VSN information](#)

Overview

Virtual Server Networking is an Open Automation tool that enables Dell Networking switch/routers in a data center network to retrieve configuration information from hypervisors. VMware vSphere and Citrix Xen hypervisors are supported.

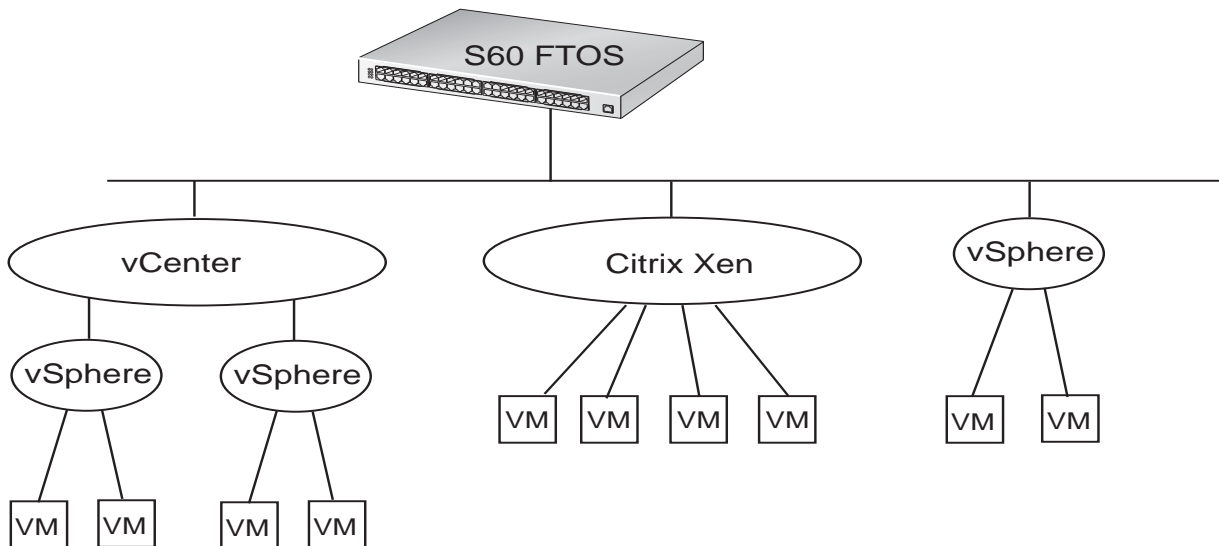
Both VMware and Citrix Xen provide SDKs and APIs for accessing their configuration objects. VSN requires Layer 3 connectivity to access a hypervisor.

Figure 7-1 shows an example of the network architecture in which a Dell Networking OS switch is connected to multiple servers, each of which may run a different type of hypervisor. The vCenter hypervisor from VMware is a centralized server management system that can manage multiple vSphere operating systems on which multiple virtual machines (VMs) can run. The VMware ESX server is a single unit, that may be managed by the hypervisor or act as an independent unit. The Citrix Xen hypervisor uses a distributed management methodology under which a number of XenServers are grouped in a management domain, with a master server managing the other units in the domain.

Minimal packet drops may be seen when migrating VMS from one server to another. The drops may vary from one second or higher, depending on the load on the server and network.

Dell Networking OS supports up to eight hypervisor sessions. A hypervisor session can consist of a single hypervisor unit (ESX, ESXi, XenServer) or a centralized hypervisor (vCenter, Xenpool). A vSphere client is used to manage a single VMware hypervisor. A vCenter server is a centralized management server for managing multiple VMware hypervisors.

Figure 7-1. Virtual Server Networking example



VSN subscribes to hypervisor for any change to be notified to switch. Depending on the hypervisor mode configured, Dell Networking OS may automatically update its configuration, provide provisioning for configuration changes, or require system administrator intervention.

Hypervisor Modes

There are two modes for retrieving configuration information from a hypervisor on a virtual server:

- **Check:** VSN retrieves configuration information from a hypervisor and notifies the system administrator when there is a change in the network configuration; for example, when a VLAN is added or removed. A system administrator must make manual updates to the Dell Networking OS configuration.
- **Config:** VSN retrieves configuration information from a hypervisor and automatically makes the required configuration changes in Dell Networking OS on the switch.

VSN Persistency

VSN installation and configuration is persistent in the Dell Networking OS configuration and remains after a system reload. However, the configuration information retrieved through a hypervisor is not persistent. If the system reloads, when it boots up the VSN application will retrieve the network configuration of virtual servers again and reconfigure Dell Networking OS accordingly.

VLAN configuration

Management VLAN

The management interface between a switch and a hypervisor can be a single port or VLAN interface. If the connection with a hypervisor is through a VLAN, you must manually configure the VLAN interface on the switch before VSN can establish a connection with the hypervisor and retrieve information from it about virtual-server configuration.

A hypervisor's management interface can also be a data interface, which means both management traffic and data traffic can use the same interface.

Manually configured VLANs are not removed by VSN after application or configuration changes are made in Dell Networking OS on a switch.

Data VLANs

Hypervisor-aware VLANs used for data traffic are automatically configured according to the configuration parameters retrieved from the hypervisor by VSN as part of the VLAN trunk on the switch port.

Use the **show vlan** command to display the VSN hypervisor-learned VLANs on the switch. As shown in [Figure 7-2](#), VSN VLANs that have been automatically configured are displayed with a **G** tag in the left-most column and are associated with ports marked with an **H** tag. If a VSN VLAN has been manually configured on the switch, the VLAN has no tag; the associated ports are displayed with an **H** tag.

Figure 7-2. Display VSN Hypervisor-learned VLANs: show vlan

```
Dell(conf-hypervisor)#show config
!
Codes: * - Default VLAN, G - GVRP VLANs, R - Remote Port Mirroring VLANs,
       P - Primary, C - Community, I - Isolated
Q: U - Untagged, T - Tagged
    x - Dot1x untagged, X - Dot1x tagged
    G - GVRP tagged, M - Vlan-stack, H - VSN tagged
    i - Internal untagged, I - Internal tagged, v - VLT untagged, V - VLT
      tagged

      NUM      Status   Description   Q Ports
*      1        Active   U             Te 0/0,15,25,27,29,42-43
          U             Te 11/35-36
G     4001     Active   H             Te 0/35
G     4002     Active   H             Te 0/35
          4003     Active   H             Te 0/35
          T             Te 0/15
```

Hypervisor-unaware VLANs

VSN cannot discover VLAN configurations from a hypervisor. If an application requires a hypervisor-unaware VLAN, you must configure the VLAN manually. User-configured VLANs are not removed when VSN retrieves and updates a network configuration.

Installing VSN

VSN is installed as a separate Open Automation package, apart from the Dell Networking OS image and the downloaded Smart Scripting package. When you install the VSN package, VSN is loaded into Dell Networking OS.



Note: VSN is not supported in stacked configurations; it is only supported on standalone switches.

You install the VSN package file in the same way as you install an Dell Networking OS release: directly from local flash memory on a switch or from an external drive on a network server. Because the installation takes time, it is performed in the background. When the download is complete, a message is displayed on the console. The package installation updates the running-configuration file.

You must manually configure the interfaces used to connect to hypervisors. Refer to the *Dell Networking OS Configuration Guide, Interfaces* chapter for information on how to configure a VLAN or physical interface.

Prerequisites:

- Smart Scripting is a prerequisite for using Virtual Server Networking. You must first install the Smart Scripting package before you can run the VSN application (see [Installing Smart Scripting](#))

To install the VSN package:

1. On a PC or other network device, go to the Dell Networking web portal at <https://www.force10networks.com/CSPortal20/Main/SupportMain.aspx>. Click **Login**, enter your user ID and password, and click the **Login** button.
2. On the Customer Support page, click the **Software Center** tab.
3. In the left-hand column, click **Automation Software**.
4. At the bottom of the Terms and Conditions page, click **I agree**.
5. On the Automation Software page, under Software, select the file for the switch from the following list:
 - **VSNAGENT2.0.x.tar.gz** file for S55 and S60
 - **VSNAGENT-P-2.2.2.0.tar.gz** for S4810 and S4820T
 - **VSNAGENT-M-2.2.2.0.tar.gz** for MXL switch
6. When the download is complete, enter the **package install** command from the Dell Networking OS CLI on a switch to install the VSN package in the internal flash memory.

Command Syntax	Command Mode	Task
package install {flash://filename ftp://userid:password@host-ipaddress/dir-path tftp://host-ipaddress/dir-path} Where: <ul style="list-style-type: none"> • flash://filename installs the VSN file stored in flash memory on the switch. • ftp://userid:password@host-ipaddress/filepath logs in and installs VSN from a file stored on an FTP server. • tftp://host-ipaddress/filepath installs VSN from a file stored on a TFTP server. • nfsmount://filepath copies from a file stored on an NFS mount file system. 	EXEC Privilege	Install the VSN package in the running configuration of the switch from local flash memory or a network server.

7. Enter the following command to configure the Perl script (VSNAgent.pl) used for VSN operations on VMware hypervisors: `script /usr/pkg/scripts/VSNAgent/VMWare/VSNAgent.pl`.

To follow the progress of a package installation (or removal), use the **show packages** command.

Enabling VSN in a Hypervisor Session

Restrictions:

- VSN is not supported in stacked configurations; it is only supported on standalone units.
- VSN supports connections only with VMware and Xen hypervisors
- You can define up to eight hypervisor sessions on a switch.
- To connect with a VMware hypervisor running on an ESXi 5.0 server, you must configure the server's firewall to allow connections only through the management IP address. You can reconfigure the firewall by using the **esxcli network firewall** command to create a rule set that allows the IP address of a Dell Networking switch to pass the firewall. For detailed information, refer to *How to Create Custom Firewall Rules in ESXi 5.0*.
- When you establish a VSN session with a Citrix Xen hypervisor (**access** command) that operates as a slave in a pool, the connection is established with the master. Configuration and access information is retrieved from the entire pool. If the slave is removed from the pool and operates as a standalone hypervisor, the VSN session is still active with the master. In this case, information is retrieved from the pool and not from the standalone hypervisor.

To enable VSN on an interface and connect to hypervisors on network servers:

Step	Task	Command Syntax	Command Mode
1	Enable VSN on an interface.	vsn enable VSN is disabled by default on switch interfaces.	INTERFACE
2	Specify the name of a hypervisor session and enter hypervisor configuration mode.	hypervisor name Enter up to 40 characters to define the hypervisor session.	CONFIGURATION
3	Define the hypervisor type to which you want to connect. Use the show hypervisor supported command to display the currently supported hypervisor types.	type {vmware xen-citrix} There is no default value.	HYPERVISOR

Step	Task	Command Syntax	Command Mode
4	Establish the connection between the switch and a hypervisor	<p>access <i>url</i> username <i>username</i> password <i>password</i></p> <p>Where <i>url</i> is one of the following values: For a VMware hypervisor: https://[ip-address]/sdk/vimService username [<i>name</i>] password [<i>password</i>]</p> <p>For an Xen hypervisor: http://ip-address username [<i>name</i>] password [<i>password</i>]</p> <p>username <i>name</i>: Username to be used for authentication on the server. password <i>password</i>: Password to be used for authentication shown in clear text.</p>	HYPERVISOR
5	Set the mode for retrieving virtual server configurations and updating Dell Networking OS settings on the switch.	<p>mode { check config}</p> <p>check: Retrieve configuration information from the hypervisor, and notify the system administrator of any configuration changes. The configuration changes need to be entered manually on the switch.</p> <p>config: Retrieve configuration information and automatically update the configuration parameters in Dell Networking OS on the switch.</p> <p>Default: config.</p>	HYPERVISOR
6	Enable the defined hypervisor connection.	no disable	HYPERVISOR

After you enable VSN on an interface and enable a hypervisor session that connects to hypervisors on network servers, you can change the **mode** setting when the session is active. You cannot, however, change the **type** and **access** settings if the session is active. To change these settings, you must:

1. In hypervisor configuration mode, stop the session by entering the **disable** command.
2. Enter the **no type** {*vmware* | *xen-citrix*} or **no access** *url* **username** *username* **password** *password* command to remove a configured setting.
3. Enter the **type** {*vmware* | *xen-citrix*} or **access** *url* **username** *username* **password** *password* command to configure a new setting.

Discovery

The discovery process starts after you enter the **no disable** command on the interface and ends in 10 minutes after connectivity is established between the switch and the hypervisor. If no connectivity is established, the switch attempts to connect for three minutes and then stops. Refer to [Connectivity](#) for more details on this process.

After you enable the link between a switch and a hypervisor, the switch uses a discovery mechanism to learn VMAC and VLAN information from the hypervisor. The discovery process also starts in the following conditions:

- Enter the **shutdown** and **no shutdown** commands on a VSN-enabled port. The discovery process resumes on the individual port only, not on all enabled ports.
- Enter the **disable** and **no disable** commands in hypervisor configuration mode, for a specified type of hypervisor connection. The discovery process is resumed on all enabled ports.
- An update arrives from a hypervisor. The discovery process resumes on all VSN-enabled ports.

In order for a switch to learn VLAN information from a hypervisor:

- Incoming traffic must be received on the VSN-enabled ports.
- There must be at least one VMAC configured on the hypervisor so that the VCAP table can capture the VMAC entries for each VSN-enabled port.

The following log messages are displayed when the discovery process is interrupted and when it starts again.

Message 1

```
Nov 28 11:34:19: %STKUNIT0-M:CP %VSNMGR-5-VSN_DISCOVERY_SUSPENDED:
Hypervisor macs not seen on Te 0/25. Discovery suspended.
```

Message 2

```
Nov 28 11:40:36: %STKUNIT0-M:CP %VSNMGR-5-VSN_DISCOVERY_RESUMED: Detected
config change in Hypervisor. Discovery of Hypervisor macs resumed on Te 0/25.
```

Connectivity

If a network server is not reachable, a log message is displayed and the VSN agent tries periodically to establish the connection with the hypervisor. The initial log message is:

Message 3

```
Xen-Citrix:Connection error for hypervisor testing:LOGIN FAILURE
```

If connectivity to a hypervisor is lost after information is retrieved and used to reconfigure a switch, the following log message is displayed. The VSN agent tries to connect to the hypervisor in the background. The information that was retrieved from the hypervisor is not deleted.

Message 4

```
Xen-Citrix:Lost connection to hypervisor xen217. Retrying...
```

Afterwards, one of the following actions is performed:

- If connectivity with the hypervisor is re-established within three minutes after the loss of connectivity, the following log message is displayed and the retrieved information is retained:

Message 5

```
Xen-Citrix:Reestablished connection with hypervisor xen217.
```

- If connectivity with the hypervisor is not re-established within three minutes after the loss of connectivity, the following log message is displayed. The information retrieved from the hypervisor is deleted and the VLANs from the hypervisor are unconfigured:

Message 6

```
Xen-Citrix:Lost connection to hypervisor xen217. Removing learnt information.
```

Running VSN Scripts

The VSN package contains the SDKs for VMware and Citrix Xen hypervisors. The Perl and Python scripts required for VSN functionality are stored with the VSN package in the **/usr/pkg/scripts/VSNAgent** directory as follows:

- For VMware hypervisors, the Perl script is stored is at **/usr/pkg/scripts/VSNAgent/VMWare/VSNAgent.pl**.
- For Citrix Xen hypervisors, the Python script is stored is at **/usr/pkg/scripts/VSNAgent/Xen/hpAgtMain.py**



Caution: The Dell Open Automation Virtual Server Networking™ software package (the “Product”) may contain the VMware SDK for Perl, which is licensed by VMware, Inc. VMware will not provide technical support for the VMware SDK included in the Product. Users interested in writing scripts for VMware products must obtain the VMware SDK directly from VMware. You may not create scripts for VMware products through use of the VMware SDK included in the Virtual Server Networking package. End Users may use the Dell Virtual Server Networking according to the terms, conditions, and limitation of the pertinent Dell End User License Agreement only.

To run a VSN script (Perl or Python) in all connected hypervisor sessions to retrieve virtual server configurations and update Dell Networking OS settings on the switch, enter the **script** command in configuration mode.

Command Syntax	Command Mode	Task
script <i>script-name</i>	CONFIGURATION	Run a VSN script in active sessions on VMware and Xen hypervisors. For <i>script-name</i> , enter the directory path and filename where the VSN script is stored on the switch; for example: <code>script /usr/pkg/scripts/VSNAgent/VMWare/VSNAgent.pl</code> .

To stop a VSN script that is running, enter the **no** version of the **script script-name** command; for example: **no script /usr/pkg/scripts/VSNAgent/VMWare/VSNAgent.pl**.

Stopping a Hypervisor Session

Disabling a Session

Enter the **disable** command in HYPERVISOR mode to stop VSN in a hypervisor session. The **disable** command does not remove connectivity with the hypervisor or remove the session information from the system configuration.

Command Syntax	Command Mode	Task
disable	HYPERVISOR	Shut down VSN in a hypervisor session.

Removing a Session

Use the **no hypervisor** command in CONFIGURATION mode to delete the configuration of a hypervisor session from the running configuration. The **no hypervisor** command deletes the specified configuration and closes an active hypervisor session, but does not remove the VSN agent from your system.

Command Syntax	Command Mode	Task
no hypervisor <i>name</i>	CONFIGURATION	Delete a session from the system. Enter the name of the hypervisor session that you want to remove.

Uninstalling VSN



Caution: Before you uninstall the VSN package, you must first stop all VSN scripts that are currently running using the **no script *script-name*** command.

Uninstalling the VSN package removes it from the internal flash memory on a switch.

Command Syntax	Command Mode	Task
package uninstall <i>name</i> Enter the name of the VSN package, exactly as it appears in show packages output.	EXEC Privilege	Uninstall the VSN package from the system.

Viewing VSN information

To view the configuration of currently active hypervisor sessions, enter the **show configuration** command in HYPERVISOR mode.

Command Syntax	Command Mode	Task
show configuration	HYPERVISOR	Display configuration of current hypervisor sessions.

Figure 7-3. Display a Hypervisor Session: show configuration

```
Dell(conf-hypervisor)#show config
!
hypervisor LocalNetwork
mode config
access https://10.10.10.10 username admin password 7 1d28e9f33f99cf5c
```

To display a list of currently supported hypervisors, use the **show hypervisors supported** command.

Command Syntax	Command Mode	Task
show hypervisor supported	EXEC Privilege	Display a list of supported hypervisors.

Figure 7-4. Display Supported Hypervisors: show hypervisor supported

```
Dell#show hypervisor supported
vmware
xen-citrix
```

To display the components of current hypervisor sessions, including the link, virtual switch, and hypervisor to which the switch is connected, use the **show virtualswitch** command.

Command Syntax	Command Mode	Task
show virtualswitch [<i>interface</i>] [<i>virtualswitch-name</i>]	EXEC Privilege	Display general information on current hypervisor sessions. To display detailed information on a hypervisor session, enter the VSN interface and/or virtual-switch name generated by the hypervisor as displayed in show virtualswitch output (Figure 7-6).

Figure 7-5. Display All Hypervisor Sessions: show virtualswitch

```
Dell#show virtualswitch
Interface      VSwitch      Hypervisor
Gi 0/32       vSwitch3     VMWare_vmware207
Po 7          vSwitch1     VMWare_vmware206
```

Figure 7-6. Display a Specified Hypervisor Sessions: show virtualswitch

```
Dell#show virtualswitch GigabitEthernet 0/32 vSwitch3
Interface                :Gi 0/32
Hypervisor Type          :vmware
Hypervisor Name          :vmware207
Hypervisor Version       :4.1.0
Virtual Switch           :vSwitch3
Port groups              :
  Name                    :VLAN 3
  Vlan Id                 :138
VIFs:
  MAC                     MTU
  00:50:56:92:00:77      8000
  Name                    :VM Network 4
  Vlan Id                 :-
VIFs:
  MAC                     MTU
  00:0c:29:4f:66:19      8000
PIFs:
  MAC                     MTU
  00:26:55:dd:01:4f      8000
```

To display information on the virtual machines accessed on a switch interface, including the virtual machine name, VMAC address, and corresponding VLAN ID, enter the **show vmmmap** command.

Command Syntax	Command Mode	Task
show vmmmap <i>interface</i>	EXEC Privilege	Display the virtual machines accessed on a switch interface.

Figure 7-7. Display Virtual Machines Accessed on an Interface: show vmmmap

```
Dell#show vmmmap gigabitethernet 0/32
VM Name          VIF          Vlan ID
Redhat_207_03_nfs 00:0c:29:4f:66:19 -
Redhat_207_03_nfs 00:50:56:92:00:77 138
```



Note: In **show vmmmap** and **show virtualswitch** output, VLAN 1 is displayed as VLAN ID 1; VLAN 4095 is displayed without a VLAN ID as "- "

Virtual Server Networking CLI

Overview

Virtual Server Networking CLI is supported on the following platforms: S4810, S4820T, and MXL switch.



Note: VSN is not supported in stacked configurations; it is only supported on standalone switches.

Commands

- access
- disable
- hypervisor
- mode
- package install
- package uninstall
- script
- show hypervisor supported
- show packages
- show virtualswitch
- show vmmmap
- type
- vsn enable

access

S4820T

Configure the connection to access a hypervisor.

S4810] and
MXL

Syntax

[no] access url username name password password

Parameters

<i>url</i>	Enter the URL location of the desired hypervisor. For a VMware hypervisor, enter: https://[ip-address]/sdk/vimService username [name] password [password] For a Xen hypervisor, enter: http://ip-address username [name] password [password]
username name	Enter the user name to be used for authentication.
password password	Enter the password to be used for authentication in clear text.

Defaults

None

Command Modes

HYPERVISOR

Command History

Version 9.2(0.0)	Introduced on the MXL switch.
Version 8.3.19.0	Introduced on the S4820T.
Version 8.3.8.0	Introduced on the S4810.

Usage Information

VSN tries to establish a connection with a hypervisor only after the user credentials (user name and password) are configured with the **access** command.

disable

S4810] and

S4820T and
MXL

Stop a hypervisor session.

Syntax

[no] disable

Defaults

disable

Command Modes

HYPERVISOR

Command History

Version 9.2(0.0)	Introduced on the MXL switch.
Version 8.3.19.0	Introduced on the S4820T.
Version 8.3.8.0	Introduced on the S4810.

Usage Information

Entering the **disable** command in hypervisor configuration mode disables VSN in the current hypervisor session. It does not remove connectivity with the hypervisor or remove the session information from the system configuration.

Enter **no disable** to re-enable a configured hypervisor session.

hypervisor

S4810]
S4820T and
MXL

Specify the name of a hypervisor session with which VSN will connect.

Syntax [no] hypervisor *name*

Parameters

<i>name</i>	Enter up to 40 characters to specify the name of a hypervisor session to which you want to connect on network servers.
-------------	--

Defaults None

Command Modes CONFIGURATION

Command History

Version 9.2(0.0)	Introduced on the MXL switch.
Version 8.3.19.0	Introduced on the S4820T.
Version 8.3.8.0	Introduced on the S4810.

Usage Information

After you enter the command, you are placed in hypervisor configuration mode to configure settings for the session.

Enter the **no hypervisor *name*** command to remove the configuration of a specified hypervisor session from the running configuration and close active hypervisor sessions without removing the VSN agent from the system.

mode

S4810]
S4820T and
MXL

Set the hypervisor mode used to retrieve configuration information on virtual servers.

Syntax [no] mode { **check** | **config** }

Defaults **config**

Parameters

check	VSN retrieves configuration information about virtual servers from a hypervisor and notifies the system administrator if the configuration has changed (for example, a VLAN has been added or removed). Changes in Dell Networking OS configuration parameters must be entered manually on the switch.
config	VSN retrieves configuration information from the Hypervisor and implements any necessary configuration changes automatically.

Command Modes HYPERVISOR

Command History

Version 9.2(0.0)	Introduced on the MXL switch.
Version 8.3.19.0	Introduced on the S4820T.
Version 8.3.8.0	Introduced on the S4810.

Usage Information

You can use the **mode** command to change the way in which virtual-server information is retrieved in an existing hypervisor session.

The following log messages are displayed when the hypervisor mode **check** is used to retrieve configuration information on virtual servers:

Message 1

```
Dec 1 04:57:48: %STKUNIT0-M:CP %VSNMGR-5-VSN_VLAN_DISCOVERY: Te 0/35, Vlan: 4001-4008,4011-4012
```

Message 2

```
Dec 1 04:56:46: %STKUNIT0-M:CP %VSNMGR-5-VSN_VLAN_WITHDRAWAL: Te 0/35, Vlan: 4001-4008,4011-4012
```

package install

S4810
S4820T and
MXL

Install an Open Automation package, such as Virtual Server Networking. This command downloads the package from the specified location, and installs it in the internal flash memory on a switch.

Syntax

package install *location*

Parameters

<i>location</i>	Enter the location where you want to install an Open Automation package, where <i>location</i> is one of the following values: <ul style="list-style-type: none"> flash://filename installs the VSN package file stored in flash memory on the switch. ftp://userid:password@host-ip-address/file-path logs in and installs VSN from a file stored on an FTP server. tftp://host-ip-address/file-path installs VSN from a file stored on a TFTP server. nfsmount://filepath copies from a file stored on an NFS mount file system.
-----------------	--

Defaults

None

Command Modes

EXEC Privilege

Command History

Version 9.2(0.0)	Introduced on the MXL switch.
Version 8.3.19.0	Introduced on the S4820T.
Version 8.3.8.0	Introduced on the S4810.

Usage Information

Because the installation of the VSN package takes time, the installation is performed in the background. When the download is complete, a message is displayed on the console.

To follow the progress of a package installation, enter the [show packages](#) command.

package uninstall

S4810
S4820T and
MXL

Remove an installed Open Automation package, such as Virtual Server Networking, from the system.

Syntax

package uninstall *name*

Parameters

<i>name</i>	Enter the name of the Open Automation package, exactly as it appears in the show packages list.
-------------	---

Defaults

None

Command Modes

EXEC Privilege

Command History

Version 9.2(0.0)	Introduced on the MXL switch.
Version 8.3.19.0	Introduced on the S4820T.
Version 8.3.8.0	Introduced on the S4810.

Usage Information

Uninstalling the VSN package removes it from the internal flash memory on the switch. To follow the progress when removing a package from the system, enter the [show packages](#) command.



Caution: Before you uninstall the Virtual Server Networking package, you must first stop all scripts that are currently running using the **no script** *script-name* command.

Related commands

show packages	Display all the packages installed in the system.
-------------------------------	---

script

S4810
S4820T and
MXL

Run an installed VSN script (Perl or Python) on active hypervisor links to retrieve virtual server configurations and update Dell Networking OS settings on the switch.

Syntax

[no] script *script-name*

Enter the **no script** *script-name* to stop a running script.

Parameters	<i>script-name</i>	Enter the directory path and filename of where the VSN script is stored; for example, /usr/pkg/scripts/VSNAgent/VMWare/VSNAgent.pl.
Defaults	None	
Command Modes	CONFIGURATION	
Command History	Version 9.2(0.0)	Introduced on the MXL switch.
	Version 8.3.19.0	Introduced on the S4820T.
	Version 8.3.8.0	Introduced on the S4810.
Usage Information		For VMware hypervisors, the VSNAgent.pl Perl script is stored in the /usr/pkg/scripts/VSNAgent/VMWare directory.
		For Xen Citrix hypervisors, the hpAgtMain.py Python script is stored in the /usr/pkg/scripts/VSNAgent/Xen directory.

show hypervisor supported

S4810
S4820T and
MXL

Display the types of hypervisors currently supported by VSN.

Syntax	show hypervisor supported	
Defaults	None	
Command Modes	EXEC Privilege	
Command History	Version 9.2(0.0)	Introduced on the MXL switch.
	Version 8.3.19.0	Introduced on the S4820T.
	Version 8.3.8.0	Introduced on the S4810.
Usage Information	Use this information when defining types of hypervisor connections with the hypervisor command.	
Related Commands	hypervisor Define a hypervisor instance.	
Example	Dell#show hypervisor supported vmware xen-citrix	

show packages

S4810
S4820T and
MXL

Display all Open Automation packages installed on a switch.

Syntax **show packages**

Defaults None

Command Modes EXEC Privilege

Command History

Version 9.2(0.0)	Introduced on the MXL switch.
Version 8.3.19.0	Introduced on the S4820T.
Version 8.3.8.0	Introduced on the S4810.

Example

```

Dell#show packages
*****
* Package Name:SMARTSCRIPTS          Version: 2.0.0

Python 2.6.5

Perl 5.8.8
  Data::Dumper 2.126
  Class::MethodMaker 2.16
  ExtUtils::MakeMaker 6.56
  XML::NamespaceSupport 1.11
  XML::SAX 0.96
  XML::LibXML 1.70
  Compress::Raw::Bzip2 2.027
  Compress::Raw::Zlib 2.027
  IO::Compress 2.027
  URI 1.54
  HTML::Tagset 3.20
  HTML::Parser 3.65
  LWP 5.836
  Net::Telnet 3.03
  OSSP::uuid 1.0602
  UUID 0.02
  version 0.82
  Class::Inspector 1.24
  Task::Weaken 1.03
  Algorithm::Diff 1.1902
  Text::Diff 1.37
  SOAP::Lite 0.712
  Crypt::SSLeay 0.57
  URI::urn::uuid 0.03
  UUID 0.03
  Crypt::SSLeay 0.57
  Net::SNMP 6.0.0
  Net::Telnet::Cisco 1.10

HTTP Server
  mini_httpd 1.19

Perl and Python function library for Force10 SmartScripts
  smartutils 2.0.0

WebConnect Web UI and CGI scripts
  htdocs 2.0.0
*****
*****
* Package Name:VSNAGENT              Version: 2.0.0

Python 2.6.5
  XenAPI

Perl 5.8.8
  VIPerlToolkit 4.1

VSNAgent Scripts
*****

```

show virtualswitch

S4810
S4820T and
MXL

Display the components of current hypervisor sessions, including the virtual switch and name of the hypervisor session to which a switch interface is connected,

Syntax `show virtualswitch [interface] [virtualswitch-name]`

Defaults None

Parameters

<i>interface</i>	Display information on the hypervisor session established on a specified interface. Enter one of the following interface types: <ul style="list-style-type: none">For a 100/1000 Ethernet interface or 1-Gigabit Ethernet interface, enter: GigabitEthernet <i>slot/port</i>For a 10-Gigabit Ethernet interface, enter: TenGigabitEthernet <i>slot/port</i>For a port-channel interface, enter: port-channel <i>number</i> Where the valid port-channel numbers are 1 to 128.
<i>virtualswitch-name</i>	Display information on a specified virtual switch by entering the name generated by the hypervisor.

Command Modes EXEC Privilege

Command History

Version 9.2(0.0)	Introduced on the MXL switch.
Version 8.3.19.0	Introduced on the S4820T.
Version 8.3.8.0	Introduced on the S4810.

Usage Information

Use the **show virtualswitch** command to display the interface, virtual-switch name, and hypervisor-session name for all current hypervisor connections on the switch.

To display detailed information on a hypervisor session, re-enter the command with the interface and virtual-switch name for the session from the **show virtualswitch** output as shown in the example below.

Example

The following command output displays information on the hypervisor sessions established on all virtual switches on network servers connected to switch interfaces.

```
Dell#show virtualswitch
Interface      VSwitch      Hypervisor
Gi 0/32       vSwitch3     VMWare_vmware207
Po 7          vSwitch1     VMWare_vmware206
```

The following command output displays information on the hypervisor session established on virtual switch vSwitch3 on a VMware server connected to the interface 0/32.

```
Dell#show virtualswitch Gigabitethernet 0/32 vSwitch3
Interface           :Gi 0/32
Hypervisor Type    :vmware
Hypervisor Name    :vmware207
Hypervisor Version :4.1.0
Virtual Switch     :vSwitch3
Port groups       :
  Name             :VLAN 3
  Vlan Id          :138
  VIFs:
    MAC            :
    MTU            :
    00:50:56:92:00:77  8000
  Name             :VM Network 4
  Vlan Id          :-
  VIFs:
    MAC            :
    MTU            :
    00:0c:29:4f:66:19  8000
PIFs:
  MAC            :
  MTU            :
  00:26:55:dd:01:4f  8000
```



Note: In `show virtualswitch` output, VLAN 1 is displayed as VLAN ID 1; VLAN 4095 is displayed without a VLAN ID as "- "

show vmmmap

S4870
S4820T and
MXL

Display the virtual machines accessed on a switch interface.

Syntax

show vmmmap *interface*

Defaults

None

Parameters

interface

Display information on the virtual machines accessed on a specified interface. Enter one of the following interface types:

- For a 100/1000 Ethernet interface or 1-Gigabit Ethernet interface, enter: **GigabitEthernet** *slot/port*
- For a 10-Gigabit Ethernet interface, enter the keyword **TenGigabitEthernet** *slot/port*
- For a port-channel interface, enter: **port-channel** *number*
Where the valid port-channel numbers are 1 to 128.

Command Modes

EXEC Privilege


Command History	Version 9.2(0.0)	Introduced on the MXL switch.
	Version 8.3.19.0	Introduced on the S4820T.
	Version 8.3.8.0	Introduced on the S4810.

Usage Information The `show vmmmap` command displays information on the virtual machines accessed on a switch interface, including the virtual machine name, VMAC address, and corresponding VLAN ID

Related Commands	hypervisor	Define a hypervisor instance.
-------------------------	----------------------------	-------------------------------

Example

```
Dell#show vmmmap gigabitethernet 0/32
VM Name                VIF                Vlan ID
Redhat_207_03_nfs     00:0c:29:4f:66:19  -
Redhat_207_03_nfs     00:50:56:92:00:77  138
```

 **Note:** In `show vmmmap` output, VLAN 1 is displayed as VLAN ID 1; VLAN 4095 is displayed without a VLAN ID as "- "

type

S4810
S4820T and
MXL

Set the hypervisor type to which you want to connect.

Syntax `[no] type { vmware | xen-citrix }`

Defaults None

Parameters	vmware	Set the hypervisor type as VMware.
	xen-citrix	Set the hypervisor type as Xen-Citrix.

Command Modes HYPervisor

Usage Information You must configure a hypervisor type in order to enable VSN connections with virtual servers. Use the `show hypervisor supported` command to display the currently supported hypervisor types.

Command History	Version 9.2(0.0)	Introduced on the MXL switch.
	Version 8.3.19.0	Introduced on the S4820T.
	Version 8.3.8.0	Introduced on the S4810.

vsn enable

S4810
S4820T and
 MXL

Enable VSN on an interface.

Syntax

[no] vsn enable

Defaults

VSN is disabled by default on switch interfaces.

Command Modes

INTERFACE

Command History

Version 9.2(0.0)	Introduced on the MXL switch.
------------------	-------------------------------

Version 8.3.19.0	Introduced on the S4820T.
------------------	---------------------------

Version 8.3.8.0	Introduced on the S4810.
-----------------	--------------------------

Usage Information

Enter the **vsn enable** command only on hypervisor-facing interfaces. DO NOT enter this command on an interface used for inter-switch links.

Enter the **no vsn enable** command to remove the VSN configuration from the system. You must reconfigure VSN to re-enable a hypervisor session.

REST API

Representational state transfer (REST) application programming interface (API) is an integrated part of the Dell Networking Operating System (OS). The supported platforms in 9.4(0.0) release are **S4810**, **S4820T**, **S6000** and **Z9000** platforms.

HTTP and HTTPS

Use REST API to configure and monitor a Dell Networking switch over hyper text transfer protocol (HTTP) and hyper text transfer protocol secure (HTTPS).

HTTP and HTTPS are the common protocols that support read, create, update, and delete operations with the actions using methods such as GET, POST, PUT, PATCH, and DELETE. For more details, refer to RFC 2616, *Hypertext Transfer Protocol - HTTP/1.1*.

XML

Extensible markup language (XML) is a standardized, easy-to-read and easy-to-parse method to represent data. The XML protocol data unit (PDU) is used to exchange data between the Dell Networking switch and HTTP/HTTPS client. XML-based implementation used is based on RFC 4741, *NETCONF Configuration Protocol*, and RFC 6020, *YANG - a Data Modeling Language for NETCONF* standards. The system handles up to four simultaneous REST API requests.

Developers can use REST API on switches without having to code individual CLI commands and open telnet/SSH/Console connections for each command.

In addition to REST API, you can use the third-party management tools and other industry-standard management protocols to manage Dell Networking switches.

Important Points to Remember

- Internet Protocol (IP) reachability is required from REST client to Dell Networking switch.
- For enabling and disabling the REST API, refer to [REST API CLI](#).
- You cannot delete physical interfaces using REST API.
- Port 8008 is a non-secure port used for HTTP; port 8888 is a secure port used for HTTPS.

REST Authentication

To enable the Accounting, Authentication, and Authorization (AAA) accounting model, the REST API must be authenticated on Dell Networking OS. For more information about the AAA accounting mode, refer to the *Dell Networking OS Configuration Guide, Security* chapter.

The Dell Networking OS CLI user level privilege dictates the level of the REST API access. Users with privilege level 0 or 1 have read-only access; the allowed REST API method is GET. Users with privilege levels 2 through 15 have read-write access in REST API. The allowed methods are GET, PATCH, PUT, POST, and DELETE.

For information about the **privilege level** command, refer to the *Dell Networking OS Command Line Reference Guide, Security* chapter.

POST and GET Request Examples

The following output displays a POST request to create BGP 65009:

```
linux:~/REST$ cat HTTP_SEND_post_bgp
<bgp>
  <as-name>65009</as-name>
  <maximum-paths>
    <ebgp>64</ebgp>
    <ibgp>32</ibgp>
  </maximum-paths>
</bgp>
linux:~/REST$ curl -v -u admin:admin -X POST -T HTTP_SEND_post_bgp http://
10.43.48.2:8008/api/running/ftos/router
* About to connect() to 10.43.48.2 port 8008 (#0)
* Trying 10.43.48.2... connected
* Server auth using Basic with user 'admin'
> POST /api/running/ftos/router HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.22.0 (i686-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1
zlib/1.2.3.4 libidn/1.23 librtmp/2.3
> Host: 10.43.48.2:8008
> Accept: */*
> Content-Length: 118
> Expect: 100-continue
>
< HTTP/1.1 100 Continue
< Server: ConfD
< Allow: GET, POST, OPTIONS, HEAD
< Content-Length: 0
* We are completely uploaded and fine
< HTTP/1.1 204 No Content
< Server: ConfD
< Cache-control: private, no-cache, must-revalidate, proxy-revalidate
< Date: Thu, 01 Aug 2013 21:42:46 GMT
< Allow: GET, POST, OPTIONS, HEAD
< Content-Length: 0
< Content-Type: text/html
<
* Connection #0 to host 10.43.48.2 left intact
* Closing connection #0
```

The following output displays a GET request after running the POST request using CURL:

```
linux:~/REST$ curl -v -u admin:admin http://10.43.48.2:8008/api/running/ftos/router/bgp
* About to connect() to 10.43.48.2 port 8008 (#0)
*   Trying 10.43.48.2... connected
* Server auth using Basic with user 'admin'
> GET /api/running/ftos/router/bgp HTTP/1.1
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/7.22.0 (i686-pc-linux-gnu) libcurl/7.22.0 OpenSSL/1.0.1
zlib/1.2.3.4 libidn/1.23 librtmp/2.3
> Host: 10.43.48.2:8008
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: ConfD
< Cache-control: private, no-cache, must-revalidate, proxy-revalidate
< Date: Thu, 01 Aug 2013 21:50:48 GMT
< Content-Type: application/vnd.yang.data+xml
< Transfer-Encoding: chunked
<

<bgp>
  <as-name>65009</as-name>
  <timers>
    <bgp>
      <keepalive>60</keepalive>
      <hold-time>180</hold-time>
    </bgp>
  </timers>
  <maximum-paths>
    <ebgp>64</ebgp>
    <ibgp>32</ibgp>
  </maximum-paths>
</bgp>
* Connection #0 to host 10.43.48.2 left intact
* Closing connection #0
```

HTTP Status Error Codes

The REST API server returns the following HTTP status error codes:

200 OK	The request was successfully completed. A response body is returned containing a representation of the resource.
201 Created	A resource was created and the new resource URI is returned in the “Location” header.

204 No Content	The request was successfully completed, but no response body is returned.
400 Bad Request	The request could not be processed because it contains missing or invalid information (such as validation error on an input field, a missing required value, and so on).
401 Unauthorized	The request requires user authentication. The response includes a “WWW-Authenticate” header field for basic authentication.
403 Forbidden	Access to the resource was denied by the server due to authorization rules.
404 Not Found	The requested resource does not exist.
405 Method Not Allowed	The HTTP method specified in the request (DELETE, GET, HEAD, PATCH, POST, PUT) is not supported for this resource.
406 Not Acceptable	The resource identified by this request is not capable of generating the requested representation, specified in the “Accept” header or in the “format” query parameter.
409 Conflict	This code is used if a request tries to create a resource that already exists.
415 Unsupported Media Type	The format of the request is not supported.
500 Internal Error	The server encountered an unexpected condition which prevented it from fulfilling the request.
501 Not Implemented	The server does not (currently) support the functionality required to fulfill the request.
503 Unavailable	The server is currently unable to handle the request due to the resource being used by someone else or is temporarily overloaded.

REST API - Protocol Data Unit (PDU) Structure

The following features are supported to configure the REST API:

- Physical interface
- Logical interface
- BGP
- Infrastructure
- Miscellaneous

Each node is printed as:

<status> <flags> <name> <opts> <type>

- `<status>` is one of the following:
 - + for current
 - x for deprecated
 - o for obsolete
- `<flags>` is one of the following:
 - rw for configuration data
 - ro for non-configuration data
 - -x for rpcs
 - -n for notifications
- `<name>` is the name of the node.
 - (`<name>`) refers that the node is a choice node
 - `:(<name>)` refers that the node is a case node
 - -x for rpcs.



Note: If the node is augmented into the tree from another module, the name is printed as `<prefix>:<name>`.

- `<opts>` is one of the following:
 - ? for an optional leaf or presence container
 - * for a leaf-list
 - [`<keys>`] for a list's keys

`<type>` is the name of the type for leafs and leaf-lists

Configurations

TenGigabitEthernet

The following definition is for configuring and displaying the properties of a **TenGigabitEthernet**.

Module: tengigabitethernet

URI: /api/running/ftos/interface/tengigabitethernet

Methods Supported: GET and PATCH

```
rw tengigabitethernet [name]
+--rw name                    ftos:slot-port
+--rw ip
|  +--rw address?
|     +--rw primary
```

```

|   |   +--rw address?   ftos:ipv4-address-with-prefix
|   +--rw secondary [address]
|       +--rw address   ftos:ipv4-address-with-prefix
+--rw rate-interval?      uint32
+--rw port-channel-protocol?
|   +--rw lacp?
|       +--rw port-channel?   uint32
|       +--rw mode?           enumeration
|       +--rw port-priority?  uint32
+--rw ipv6?
|   +--rw address [ipv6-address]
|       +--rw ipv6-address   ftos:ipv6-address
|       +--rw eui64?        boolean
+--rw portmode?
|   +--rw hybrid? boolean
+--rw switchport?
|   +--rw backup
|       +--rw interface
|           +--rw (intf-type)?
|               +--:(tengigabitethernet)
|                   |   +--rw tengigabitethernet
|                   |       +--rw name   ftos:slot-port
|                   +--:(fortyGigE)
|                       |   +--rw fortyGigE
|                       |       +--rw name   ftos:slot-port
|                       +--:(port-channel)
|                           +--rw port-channel
|                               +--rw name   port-channel
+--rw flowcontrol?
|   +--rw rx?   enumeration
|   +--rw tx?   enumeration
+--rw mac?
|   +--rw access-group?
|   |   +--rw in?
|   |   |   +--rw WORD   access-list-name
|   |   |   +--rw vlan*  vlan-range
|   |   +--rw out?
|   |       +--rw WORD   access-list-name
|   +--rw learning-limit?
|       +--rw count?           uint32
|       +--rw (operations)?

```

```

| | +--:(sticky)
| | | +--rw sticky?          empty
| | +--:(non-sticky)
| |   +--rw dynamic?        boolean
| |   +--rw station-move?   boolean
| +--rw learn-limit-violation?
| | +--rw (action)?
| |   +--:(log)
| |     | +--rw log          boolean
| |     +--:(shutdown)
| |       +--rw shutdown    boolean
| +--rw station-move-violation?
|   +--rw (action)?
|     +--:(log)
|       | +--rw log          boolean
|       +--:(shutdown-both)
|         | +--rw shutdown-both    boolean
|         +--:(shutdown-offending)
|           | +--rw shutdown-offending    boolean
|           +--:(shutdown-original)
|             +--rw shutdown-original    boolean
+--rw speed?          enumeration
+--rw negotiation?
| +--rw auto?    boolean
+--rw description?    string
+--rw mtu?           uint32
+--rw shutdown?     boolean

```

FortyGigabitEthernet

The following definition is for configuring and displaying the properties of a **FortyGigabitEthernet**.

Module: fortyGigE

URI: /api/running/ftos/interface/fortyGigE

Methods Supported: GET and PATCH

```

rw fortyGigE [name]
+--rw name          ftos:slot-port
+--rw ip
| +--rw address?

```



```

|     +--rw primary
|     |   +--rw address?   ftos:ipv4-address-with-prefix
|     +--rw secondary [address]
|           +--rw address   ftos:ipv4-address-with-prefix
+--rw rate-interval?          uint32
+--rw port-channel-protocol?
|   +--rw lacp?
|     +--rw port-channel?   uint32
|     +--rw mode?          enumeration
|     +--rw port-priority?  uint32
+--rw ipv6?
|   +--rw address [ipv6-address]
|     +--rw ipv6-address   ftos:ipv6-address
|     +--rw eui64?        boolean
+--rw portmode?
|   +--rw hybrid?         boolean
+--rw switchport?
|   +--rw backup
|     +--rw interface
|       +--rw (intf-type)?
|         +--:(tengigabitethernet)
|         |   +--rw tengigabitethernet
|         |   |   +--rw name   ftos:slot-port
|         |   +--:(fortyGigE)
|         |   |   +--rw fortyGigE
|         |   |   +--rw name   ftos:slot-port
|         +--:(port-channel)
|         |   +--rw port-channel
|         |   +--rw name     port-channel
+--rw flowcontrol?
|   +--rw rx?  enumeration
|   +--rw tx?  enumeration
+--rw mac?
|   +--rw access-group?
|     |   +--rw in?
|     |   |   +--rw WORD   access-list-name
|     |   |   +--rw vlan*  vlan-range
|     |   +--rw out?
|     |   |   +--rw WORD   access-list-name
|     +--rw learning-limit?
|       +--rw count?          uint32

```

```

|      +--rw (operations)?
|      |      +--:(sticky)
|      |      |      +--rw sticky?                empty
|      |      +--:(non-sticky)
|      |      +--rw dynamic?                    boolean
|      |      +--rw station-move?              boolean
|      +--rw learn-limit-violation?
|      |      +--rw (action)?
|      |      +--:(log)
|      |      |      +--rw log                boolean
|      |      +--:(shutdown)
|      |      +--rw shutdown                boolean
|      +--rw station-move-violation?
|      |      +--rw (action)?
|      |      +--:(log)
|      |      |      +--rw log                boolean
|      |      +--:(shutdown-both)
|      |      |      +--rw shutdown-both        boolean
|      |      +--:(shutdown-offending)
|      |      |      +--rw shutdown-offending    boolean
|      |      +--:(shutdown-original)
|      |      +--rw shutdown-original          boolean
+--rw speed?                enumeration
+--rw negotiation?
| +--rw auto?              boolean
+--rw description?         string
+--rw mtu?                 uint32
+--rw shutdown?           boolean

```

Port-Channel

The following definition is for configuring and displaying the properties of a **Port-channel** interface.

Module: port-channel

URI: /api/running/ftos/interfaces/port-channel

Methods Supported: GET, POST, PUT, PATCH, and DELETE

```

rw port-channel [name]
+--rw name                port-channel
+--rw channel-members

```

```

| +--rw tengigabitethernet [name]
| | +--rw name      ftos:slot-port-range
| +--rw fortyGigE [name]
|   +--rw name      ftos:slot-port-range
+--rw ip
| +--rw address?
|   +--rw primary
|     | +--rw address?  ftos:ipv4-address-with-prefix
|     +--rw secondary [address]
|       +--rw address  ftos:ipv4-address-with-prefix
+--rw min-links?      uint32
+--rw lacp?           enumeration
+--rw rate-interval? uint32
+--rw description?   string
+--rw mtu?            uint32
+--rw shutdown?      boolean
rw vlan [vlan-id]

```

VLAN

The following definition is for configuring and displaying the properties of **Vlan** interface.

Module: vlan

URI: /api/running/ftos/interfaces/vlan

Methods Supported: GET, POST, PUT, PATCH, and DELETE

```

+--rw vlan-id      vlan-id-num
+--rw name?        vlan-name
+--rw tagged
| +--rw tengigabitethernet [name]
| | +--rw name      ftos:slot-port
| +--rw fortyGigE [name]
| | +--rw name      ftos:slot-port
| +--rw port-channel [name]
|   +--rw name      uint32
+--rw untagged
| +--rw tengigabitethernet [name]
| | +--rw name      ftos:slot-port
| +--rw fortyGigE [name]
| | +--rw name      ftos:slot-port

```

```

|   +--rw port-channel [name]
|     +--rw name      uint32
+--rw description?   string
+--rw mtu?           uint32
+--rw shutdown?     boolean
+--rw ip
|   +--rw address?
|     +--rw primary
|       |   +--rw address?  ftos:ipv4-address-with-prefix
|       +--rw secondary [address]
|         +--rw address    ftos:ipv4-address-with-prefix
+--rw ipv6?
  +--rw address [ipv6-address]
    +--rw ipv6-address  string
    +--rw eui64?       boolean

```

Static Route

The following definition is for configuring and displaying the properties of **Static route**.

Module: ip

URI: /api/running/ftos/ip

Methods Supported: GET, POST, PUT, PATCH, and DELETE

```

rw route [ip-address-prefix exit-interface exit-port next-hop]
+--rw ip-address-prefix  ftos:ipv4-address-with-prefix
+--rw exit-interface     union
+--rw exit-port         union
+--rw next-hop          union
+--rw metric?           uint32
+--rw permanent?       empty
+--rw tag?              uint32

```

BGP

The following definition is for configuring and displaying the properties of **BGP**.

Module: bgp

URI: /api/running/ftos/router/bgp

Methods Supported: GET, POST, PUT, PATCH, and DELETE

```
rw bgp [as-name]
+--rw as-name          ftos:as-number
+--rw network [ipv4-address]
|  +--rw ipv4-address  ftos:ipv4-address-with-prefix
+--rw timers
|  +--rw bgp
|     +--rw keepalive?  uint32
|     +--rw hold-time?  uint32
+--rw maximum-paths
|  +--rw ebgp?  uint32
|  +--rw ibgp?  uint32
+--rw peer-group [peer-group-name]
|  +--rw peer-group-name      string
|  +--rw passive?             boolean
|  +--rw limit?               uint32
|  +--rw match-af?            boolean
|  +--rw remote-as?           ftos:as-number
|  +--rw description?         string
|  +--rw shutdown?            boolean
|  +--rw ebgp-multihop?       uint32
|  +--rw update-source-loopback?  uint32
|  +--rw next-hop-self?       boolean
+--rw neighbor [neighbor-router]
|  +--rw neighbor-router      union
|  +--rw remote-as?           ftos:as-number
|  +--rw peer-group-name?     string
|  +--rw description?         string
|  +--rw shutdown?            boolean
```

Operational

TenGigabitEthernet

The following definition is for displaying the statistics of a **TenGigabitEthernet**.

Module: stats

URI: /api/operational/ftos/stats/interfaces/tengigabitethernet

Methods Supported: GET

```
|  +--ro tengigabitethernet [name]
|  |  +--ro name                               ftos:slot-port
|  |  +--ro description?                       string
|  |  +--ro ifType?                             ianaiftype-mib:IANAifType
|  |  +--ro if-index?                           uint32
|  |  +--ro ifAdminStatus?                       enumeration
|  |  +--ro ifOperStatus?                       enumeration
|  |  +--ro ifOperDownReason?                   string
|  |  +--ro ifMtu?                               int32
|  |  +--ro ifSpeed?                             yang:gauge32
|  |  +--ro ifQueueingStrategy?                 string
|  |  +--ro ifPhysAddress?                       yang:phys-address
|  |  +--ro current-intf-address?                string
|  |  +--ro ipv4-address?                       string
|  |  +--ro ipv4-address-mode?                  enumeration
|  |  +--ro local-ipv6-address?                 ipv6-address-with-prefix
|  |  +--ro global-ipv6-address?               ipv6-address-with-prefix
|  |  +--ro dhcp-client?                       string
|  |  +--ro line-speed?                         enumeration
|  |  +--ro ip-mtu?                             uint32
|  |  +--ro arp-type?                           enumeration
|  |  +--ro arp-timeout?                       uint32
|  |  +--ro last-clear-intf-cters?             uint32
|  |  +--ro last-intf-status-changed?         uint32
|  |  +--ro ifHCInOctets?                       yang:counter64
|  |  +--ro ifHCInUcastPkts?                   yang:counter64
|  |  +--ro ifHCInMulticastPkts?              yang:counter64
|  |  +--ro ifHCInBroadcastPkts?              yang:counter64
|  |  +--ro ifHCIn64BytesPkts?                 yang:counter64
|  |  +--ro ifHCInOver64BytesPkts?            yang:counter64
```

		+++ro ifHCInOver127BytesPkts?	yang:counter64
		+++ro ifHCInOver255BytesPkts?	yang:counter64
		+++ro ifHCInOver511BytesPkts?	yang:counter64
		+++ro ifHCInOver1023BytesPkts?	yang:counter64
		+++ro ifHCInSymbolErrors?	yang:counter64
		+++ro ifHCInRunts?	yang:counter64
		+++ro ifHCInGiants?	yang:counter64
		+++ro ifHCInThrottles?	yang:counter64
		+++ro ifHCInCRC?	yang:counter64
		+++ro ifHCInOverRun?	yang:counter64
		+++ro ifHCInDiscarded?	yang:counter64
		+++ro ifHCOutOctets?	yang:counter64
		+++ro ifHCOutUcastPkts?	yang:counter64
		+++ro ifHCOutMulticastPkts?	yang:counter64
		+++ro ifHCOutBroadcastPkts?	yang:counter64
		+++ro ifHCOutUnicastPkts?	yang:counter64
		+++ro ifHCOut64BytesPkts?	yang:counter64
		+++ro ifHCOutOver64BytesPkts?	yang:counter64
		+++ro ifHCOutOver127BytesPkts?	yang:counter64
		+++ro ifHCOutOver255BytesPkts?	yang:counter64
		+++ro ifHCOutOver511BytesPkts?	yang:counter64
		+++ro ifHCOutOver1023BytesPkts?	yang:counter64
		+++ro ifHCOutUnderRuns?	yang:counter64
		+++ro ifHCOutThrottles?	yang:counter64
		+++ro ifHCOutCollisions?	yang:counter64
		+++ro ifHCOutWredDrops?	yang:counter64
		+++ro ifHCOutDiscarded?	yang:counter64
		+++ro ifRateInterval?	yang:counter32
		+++ro ifInPktsInMbps?	yang:counter32
		+++ro ifInPktsInPktsPerSec?	yang:counter32
		+++ro ifInCentLineRate?	yang:counter32
		+++ro ifOutPktsInMbps?	yang:counter32
		+++ro ifOutPktsInPktsPerSec?	yang:counter32
		+++ro ifOutCentLineRate?	yang:counter32
		+++ro ifCounterDiscontinuityTime?	yang:timestamp

FortyGigabitEthernet

The following definition is for displaying the statistics of a **FortyGigabitEthernet**.

URI: /api/operational/ftos/stats/interfaces/fortyGigE

Methods Supported: GET

```
|  +---ro fortyGigE [name]
|  |  +---ro name                               ftos:slot-port
|  |  +---ro description?                       string
|  |  +---ro ifType?                            ianaiftype-mib:IANAifType
|  |  +---ro if-index?                          uint32
|  |  +---ro ifAdminStatus?                     enumeration
|  |  +---ro ifOperStatus?                      enumeration
|  |  +---ro ifOperDownReason?                  string
|  |  +---ro ifMtu?                             int32
|  |  +---ro ifSpeed?                           yang:gauge32
|  |  +---ro ifQueueingStrategy?                string
|  |  +---ro ifPhysAddress?                     yang:phys-address
|  |  +---ro current-intf-address?              string
|  |  +---ro ipv4-address?                       string
|  |  +---ro ipv4-address-mode?                 enumeration
|  |  +---ro local-ipv6-address?                ipv6-address-with-prefix
|  |  +---ro global-ipv6-address?              ipv6-address-with-prefix
|  |  +---ro dhcp-client?                       string
|  |  +---ro line-speed?                        enumeration
|  |  +---ro ip-mtu?                            uint32
|  |  +---ro arp-type?                          enumeration
|  |  +---ro arp-timeout?                       uint32
|  |  +---ro last-clear-intf-cters?             uint32
|  |  +---ro last-intf-status-changed?         uint32
|  |  +---ro ifHCInOctets?                      yang:counter64
|  |  +---ro ifHCInUcastPkts?                  yang:counter64
|  |  +---ro ifHCInMulticastPkts?              yang:counter64
|  |  +---ro ifHCInBroadcastPkts?              yang:counter64
|  |  +---ro ifHCIn64BytesPkts?                yang:counter64
|  |  +---ro ifHCInOver64BytesPkts?            yang:counter64
|  |  +---ro ifHCInOver127BytesPkts?           yang:counter64
|  |  +---ro ifHCInOver255BytesPkts?           yang:counter64
|  |  +---ro ifHCInOver511BytesPkts?           yang:counter64
|  |  +---ro ifHCInOver1023BytesPkts?          yang:counter64
```


		+++ro ifHCInSymbolErrors?	yang:counter64
		+++ro ifHCInRunts?	yang:counter64
		+++ro ifHCInGiants?	yang:counter64
		+++ro ifHCInThrottles?	yang:counter64
		+++ro ifHCInCRC?	yang:counter64
		+++ro ifHCInOverRun?	yang:counter64
		+++ro ifHCInDiscarded?	yang:counter64
		+++ro ifHCOctets?	yang:counter64
		+++ro ifHCOutUcastPkts?	yang:counter64
		+++ro ifHCOutMulticastPkts?	yang:counter64
		+++ro ifHCOutBroadcastPkts?	yang:counter64
		+++ro ifHCOutUnicastPkts?	yang:counter64
		+++ro ifHCOut64BytesPkts?	yang:counter64
		+++ro ifHCOutOver64BytesPkts?	yang:counter64
		+++ro ifHCOutOver127BytesPkts?	yang:counter64
		+++ro ifHCOutOver255BytesPkts?	yang:counter64
		+++ro ifHCOutOver511BytesPkts?	yang:counter64
		+++ro ifHCOutOver1023BytesPkts?	yang:counter64
		+++ro ifHCOutUnderRuns?	yang:counter64
		+++ro ifHCOutThrottles?	yang:counter64
		+++ro ifHCOutCollisions?	yang:counter64
		+++ro ifHCOutWredDrops?	yang:counter64
		+++ro ifHCOutDiscarded?	yang:counter64
		+++ro ifRateInterval?	yang:counter32
		+++ro ifInPktsInMbps?	yang:counter32
		+++ro ifInPktsInPktsPerSec?	yang:counter32
		+++ro ifInCentLineRate?	yang:counter32
		+++ro ifOutPktsInMbps?	yang:counter32
		+++ro ifOutPktsInPktsPerSec?	yang:counter32
		+++ro ifOutCentLineRate?	yang:counter32
		+++ro ifCounterDiscontinuityTime?	yang:timestamp

Port-channel

The following definition is for displaying the statistics of **Port-channel**.

URI: /api/operational/ftos/stats/interfaces/port-channel

Methods Supported: GET

	+++ro port-channel [name]	
		+++ro name port-channel

		+++ro description?	string
		+++ro ifType?	ianaiftype-mib:IANAifType
		+++ro if-index?	uint32
		+++ro ifAdminStatus?	enumeration
		+++ro ifOperStatus?	enumeration
		+++ro ifOperDownReason?	string
		+++ro ifMtu?	int32
		+++ro ifSpeed?	yang:gauge32
		+++ro ifQueueingStrategy?	string
		+++ro ifPhysAddress?	yang:phys-address
		+++ro current-intf-address?	string
		+++ro ipv4-address?	string
		+++ro ipv4-address-mode?	enumeration
		+++ro local-ipv6-address?	ipv6-address-with-prefix
		+++ro global-ipv6-address?	ipv6-address-with-prefix
		+++ro dhcp-client?	string
		+++ro line-speed?	enumeration
		+++ro ip-mtu?	uint32
		+++ro arp-type?	enumeration
		+++ro arp-timeout?	uint32
		+++ro last-clear-intf-cters?	uint32
		+++ro last-intf-status-changed?	uint32
		+++ro owner?	enumeration
		+++ro memberInterfaces [name]	
			+++ro name string
			+++ro ifOperStatus? enumeration
		+++ro minLinks?	uint32
		+++ro ifHCInOctets?	yang:counter64
		+++ro ifHCInUcastPkts?	yang:counter64
		+++ro ifHCInMulticastPkts?	yang:counter64
		+++ro ifHCInBroadcastPkts?	yang:counter64
		+++ro ifHCIn64BytesPkts?	yang:counter64
		+++ro ifHCInOver64BytesPkts?	yang:counter64
		+++ro ifHCInOver127BytesPkts?	yang:counter64
		+++ro ifHCInOver255BytesPkts?	yang:counter64
		+++ro ifHCInOver511BytesPkts?	yang:counter64
		+++ro ifHCInOver1023BytesPkts?	yang:counter64
		+++ro ifHCInSymbolErrors?	yang:counter64
		+++ro ifHCInRunts?	yang:counter64
		+++ro ifHCInGiants?	yang:counter64
		+++ro ifHCInThrottles?	yang:counter64

	+++ro ifHCInCRC?	yang:counter64
	+++ro ifHCInOverRun?	yang:counter64
	+++ro ifHCInDiscarded?	yang:counter64
	+++ro ifHCOutOctets?	yang:counter64
	+++ro ifHCOutUcastPkts?	yang:counter64
	+++ro ifHCOutMulticastPkts?	yang:counter64
	+++ro ifHCOutBroadcastPkts?	yang:counter64
	+++ro ifHCOutUnicastPkts?	yang:counter64
	+++ro ifHCOut64BytesPkts?	yang:counter64
	+++ro ifHCOutOver64BytesPkts?	yang:counter64
	+++ro ifHCOutOver127BytesPkts?	yang:counter64
	+++ro ifHCOutOver255BytesPkts?	yang:counter64
	+++ro ifHCOutOver511BytesPkts?	yang:counter64
	+++ro ifHCOutOver1023BytesPkts?	yang:counter64
	+++ro ifHCOutUnderRuns?	yang:counter64
	+++ro ifHCOutThrottles?	yang:counter64
	+++ro ifHCOutCollisions?	yang:counter64
	+++ro ifHCOutWredDrops?	yang:counter64
	+++ro ifHCOutDiscarded?	yang:counter64
	+++ro ifRateInterval?	yang:counter32
	+++ro ifInPktsInMbps?	yang:counter32
	+++ro ifInPktsInPktsPerSec?	yang:counter32
	+++ro ifInCentLineRate?	yang:counter32
	+++ro ifOutPktsInMbps?	yang:counter32
	+++ro ifOutPktsInPktsPerSec?	yang:counter32
	+++ro ifOutCentLineRate?	yang:counter32
	+++ro ifCounterDiscontinuityTime?	yang:timestamp

VLAN

The following definition is for displaying the statistics of **vlan**.

URI: /api/operational/ftos/stats/interfaces/vlan

Methods Supported: GET

```
+++ro vlan [name]
  +++ro name                vlan-id-num
  +++ro vlan-name?          vlan-name
  +++ro description?        string
  +++ro ifType?              ianaiftype-mib:IANAifType
  +++ro if-index?           uint32
```

```

+---ro ifAdminStatus?          enumeration
+---ro ifOperStatus?          enumeration
+---ro ifOperDownReason?      string
+---ro ifMtu?                 int32
+---ro ifSpeed?               yang:gauge32
+---ro ifQueueingStrategy?    string
+---ro ifPhysAddress?         yang:phys-address
+---ro current-intf-address?   string
+---ro ipv4-address?          string
+---ro ipv4-address-mode?     enumeration
+---ro local-ipv6-address?    ipv6-address-with-prefix
+---ro global-ipv6-address?   ipv6-address-with-prefix
+---ro dhcp-client?           string
+---ro line-speed?            enumeration
+---ro ip-mtu?                uint32
+---ro arp-type?              enumeration
+---ro arp-timeout?           uint32
+---ro last-clear-intf-cters?  uint32
+---ro last-intf-status-changed? uint32

```

IP Statistics

The following definition is for displaying the statistics of an **ip**.

URI: /api/operational/ftos/stats/ip/interface

Methods Supported: GET

```

+---ro ip
| +---ro interface [name]
| | +---ro name                string
| | +---ro description?       string
| | +---ro type?              ianaiftype-mib:IANAifType
| | +---ro ifAdminStatus?     enumeration
| | +---ro ifOperStatus?     enumeration
| | +---ro ip
| | | +---ro address?
| | |   +---ro primary
| | |     | +---ro address?    ftos:ipv4-address-with-prefix
| | |     +---ro secondary [address]
| | |       +---ro address    ftos:ipv4-address-with-prefix
| | +---ro ifVirtualAddress?  ftos:ipv4-address-with-prefix

```

```

| | +--ro ifBroadCastAddress?  inet:ip-address
| | +--ro ifAddressInput?      enumeration
| | +--ro ifIpMtu?             int32
| | +--ro protocols
| |   +--ro udpHelper [helper]
| |     | +--ro helper      inet:ip-address
| |   +--ro udpBroadcast?    inet:ip-address
| |   +--ro directedBroadcast?  boolean
| |   +--ro proxyArp?         boolean
| |   +--ro splitHorizon?     boolean
| |   +--ro poisonReverse?    boolean
| |   +--ro icmpRedirects?    boolean
| |   +--ro icmpUnreachables?  boolean

```

IP Route

The following definition is for displaying the entries of **ip route**.

URI: /api/operational/ftos/stats/ip/route

Methods Supported: GET

```

+--ro route
| +--ro gatewayOfLastResort?  string
| +--ro route-entry [destination]
| | +--ro destination      inet:ip-prefix
| | +--ro gateway?         inet:ip-address
| | +--ro routeType?       enumeration
| | +--ro routeOwner?      string
| | +--ro metric?          string
| | +--ro lastChange?      yang:timestamp
| | +--ro state?           enumeration
| | +--ro defaultRoute?    boolean
| | +--ro summary?         boolean
| +--ro summary
|   +--ro connectedActive?   uint32
|   +--ro connectedInactive? uint32
|   +--ro dynamicActive?     uint32
|   +--ro dynamicInactive?   uint32
|   +--ro staticActive?      uint32
|   +--ro staticInactive?    uint32
|   +--ro routeSizeActive?   uint32

```

```
|    +--ro routeSizeInactive?    uint32
```

BGP

The following definition is for displaying the operational data of **BGP**.

URI: /api/operational/ftos/stats/ip/bgp

Methods Supported: GET

```
+--ro bgp
  +--ro tableVersion?    uint32
  +--ro localRouterId?   string
  +--ro routes
    | +--ro prefixList [networkPrefix networkPrefixLen nextHopAddress seqNum]
    |   +--ro networkPrefix      inet:ip-address
    |   +--ro networkPrefixLen   inet-address:InetAddressPrefixLength
    |   +--ro nextHopAddress     inet:ip-address
    |   +--ro seqNum             uint32
    |   +--ro metric?           uint32
    |   +--ro localPref?        uint32
    |   +--ro weight?          uint32
    |   +--ro nextHopCostIndex?  uint32
    |   +--ro asPathString?     string
    |   +--ro pathSource?       enumeration
    |   +--ro originCode?       enumeration
    |   +--ro isNlreLocAggtd?   snmpv2-tc:TruthValue
    |   +--ro isStale?          snmpv2-tc:TruthValue
    |   +--ro statusCode?       enumeration
    |   +--ro bestRoute?        snmpv2-tc:TruthValue
```

BGP Neighbors

The following definition is for displaying the operational data of **BGP neighbors**.

URI: /api/operational/ftos/stats/ip/bgp/neighbors

Methods Supported: GET

```
+--ro neighbors
  +--ro neighbor [neighborAddress]
    +--ro neighborAddress      inet:ipv4-address
    +--ro advertisedRoutes
```

```

seqNum] | +--ro prefixList [networkPrefix networkPrefixLen nextHopAddress
|
| +--ro networkPrefix      inet:ip-address
| +--ro networkPrefixLen  inet-address:InetAddressPrefixLength
| +--ro nextHopAddress     inet:ip-address
| +--ro seqNum             uint32
| +--ro metric?            uint32
| +--ro localPref?        uint32
| +--ro weight?           uint32
| +--ro nextHopCostIndex? uint32
| +--ro asPathString?     string
| +--ro pathSource?       enumeration
| +--ro originCode?       enumeration
| +--ro isNlreLocAggtd?   snmpv2-tc:TruthValue
| +--ro isStale?          snmpv2-tc:TruthValue
| +--ro statusCode?       enumeration
| +--ro bestRoute?        snmpv2-tc:TruthValue
+--ro receivedRoutes
seqNum] +--ro prefixList [networkPrefix networkPrefixLen nextHopAddress
|
| +--ro networkPrefix      inet:ip-address
| +--ro networkPrefixLen  inet-address:InetAddressPrefixLength
| +--ro nextHopAddress     inet:ip-address
| +--ro seqNum             uint32
| +--ro metric?            uint32
| +--ro localPref?        uint32
| +--ro weight?           uint32
| +--ro nextHopCostIndex? uint32
| +--ro asPathString?     string
| +--ro pathSource?       enumeration
| +--ro originCode?       enumeration
| +--ro isNlreLocAggtd?   snmpv2-tc:TruthValue
| +--ro isStale?          snmpv2-tc:TruthValue
| +--ro statusCode?       enumeration
| +--ro bestRoute?        snmpv2-tc:TruthValue

```

MAC Address Table

The following definition(s) is for displaying the entries of **mac-address-table**.

URI: /api/operational/ftos/stats/mac-address-table

Methods Supported: GET

```
+---ro mac-address-table
|  +---ro count
|  |  +---ro dynamicMacCount?  uint32
|  |  +---ro staticMacCount?   uint32
|  |  +---ro stickyMacCount?   uint32
|  |  +---ro totalMacInUse?    uint32
|  +---ro agingTime?          uint32
|  +---ro macVlan [vlan-id]
|  |  +---ro vlan-id            vlan-id
|  |  +---ro count
|  |  |  +---ro dynamicMacCount?  uint32
|  |  |  +---ro staticMacCount?   uint32
|  |  |  +---ro stickyMacCount?   uint32
|  |  |  +---ro totalMacInUse?    uint32
|  |  +---ro macAddressList [mac-address]
|  |  |  +---ro mac-address      yang:phys-address
|  |  |  +---ro type?           enumeration
|  |  |  +---ro interface?     string
|  |  |  +---ro state?         enumeration
|  +---ro multicast
|  |  +---ro count?            uint32
|  |  +---ro macVlan [vlan-id]
|  |  |  +---ro vlan-id          vlan-id
|  |  |  +---ro staticMacCount?  uint32
|  |  |  +---ro macAddressList [mac-address]
|  |  |  |  +---ro mac-address    yang:phys-address
|  |  |  |  +---ro state?        enumeration
|  |  |  +---ro L2MCIndex?      uint32
|  |  |  +---ro interface-list?  string
```


System Alarm

The following definition is for displaying **alarm** entries.

URI: /api/operational/ftos/stats/alarms

Methods Supported: GET

```
+--ro alarms
| +--ro major-alarms [index]
| | +--ro index          uint32
| | +--ro alarm-description? string
| | +--ro duration?     yang:timestamp
| +--ro minor-alarms [index]
| | +--ro index          uint32
| | +--ro alarm-description? string
| | +--ro duration?     yang:timestamp
| +--ro alarm-thresholds [unit-number]
|   +--ro unit-number    uint32
|   +--ro minor?        uint32
|   +--ro minor-off?    uint32
|   +--ro major?        uint32
|   +--ro major-off?    uint32
|   +--ro shutdown?     uint32
```

System Inventory

The following definition is for displaying the **inventory** details.

URI: /api/operational/ftos/stats/inventory

Methods Supported: GET

```
+--ro inventory
| +--ro system-type?    string
| +--ro system-mode?    string
| +--ro sw-version?     string
| +--ro parts [index]
| | +--ro index          uint32
| | +--ro stack-unit?    uint32
| | +--ro part-type?     string
| | +--ro serial-number? string
| | +--ro part-number?   string
| | +--ro revision?     string
```

```

| | +--ro piece-part-id?      string
| | +--ro ppid-revision?     string
| | +--ro service-tag?      string
| | +--ro express-service-code? string
| +--ro protocols?          string

```

System Version

The following definition is for displaying the **version-info**.

URI: /api/operational/ftos/stats/version-info

Methods Supported: GET

```

+--ro version-info
| +--ro os-ver?          string
| +--ro sw-ver?          string
| +--ro build-time?     string
| +--ro build-path?     string
| +--ro up-time?        string
| +--ro image-name?     string
| +--ro chassis-type?   string
| +--ro processor-type? string
| +--ro flash-info?     string
| +--ro card-info [index]
| | +--ro index          uint32
| | +--ro card-info-detail? string
| +--ro card-if-info [index]
|   +--ro index          uint32
|   +--ro card-if-info-detail? string

```

VLAN

The following definition is for displaying the entries of **vlan**.

URI: /api/operational/ftos/stats/vlan

Methods Supported: GET

```

+--ro vlan [vlanId]
| +--ro vlanId          vlan-id-num
| +--ro vlanName?      vlan-name
| +--ro defaultVlan?   boolean
| +--ro vlanCodes?     vlanCodes

```

```

| +--ro vlanStatus?          vlanStatus
| +--ro vlanDescription?    string
| +--ro vlanPortList [vlanPortListNum]
|   +--ro vlanPortListNum    int32
|   +--ro vlanPortListPortMode?  vlanPortMode
|   +--ro vlanPortListIntfType?  vlanIntfType
|   +--ro vlanPortListData?    string

```

System

The following definition is for displaying the **system** entries.

URI: /api/operational/ftos/stats/system

Methods Supported: GET

```

+--ro system
  +--ro stack-mac-address?  yang:phys-address
  +--ro reload-type?        reload-type
  +--ro next-boot?         reload-type
  +--ro stack-units [unit-number]
  | +--ro unit-number      uint32
  | +--ro unit-role?       string
  | +--ro unit-status?     string
  | +--ro required-type?   string
  | +--ro current-type?    string
  | +--ro os-ver?          string
  | +--ro num-ports?       uint32
  +--ro power-supplies [index]
  | +--ro index            uint32
  | +--ro stack-unit?     uint32
  | +--ro bay-id?         uint32
  | +--ro bay-status?     string
  | +--ro ps-type?        string
  | +--ro fan-status?     string
  +--ro fans [index]
  | +--ro index            uint32
  | +--ro stack-unit?     uint32
  | +--ro bay-id?         uint32
  | +--ro bay-status?     string
  | +--ro fan-id?         uint32
  | +--ro fan-status?     string

```

```
+--ro fan-speed?      uint32
```

BGP MIB

The following definition is for displaying the entries in **BGP MIB table**.

Module: f10-bgp4-v2

URI: /api/operational/f10BgpM2

Methods Supported: GET

```
+--ro f10BgpM2
|  +--ro f10BgpM2CapabilitySupportAvailable?  snmpv2-tc:TruthValue
|  +--ro f10BgpM2AsSize?                      enumeration
|  +--ro f10BgpM2LocalAs?
inet-address:InetAutonomousSystemNumber
|  +--ro f10BgpM2LocalIdentifier?             inet:ip-address
|  +--ro f10BgpM2RouteReflector?            snmpv2-tc:TruthValue
|  +--ro f10BgpM2ClusterId?                  f10-bgp4:F10BgpM2Identifier
|  +--ro f10BgpM2ConfederationRouter?        snmpv2-tc:TruthValue
|  +--ro f10BgpM2ConfederationId?
inet-address:InetAutonomousSystemNumber
|  +--ro f10BgpM2CfgBaseScalarStorageType?   snmpv2-tc:StorageType
|  +--ro f10BgpM2CfgLocalAs?
inet-address:InetAutonomousSystemNumber
|  +--ro f10BgpM2CfgLocalIdentifier?         inet:ip-address
|  +--ro f10BgpM2CfgRouteReflector?         snmpv2-tc:TruthValue
|  +--ro f10BgpM2CfgClusterId?              f10-bgp4:F10BgpM2Identifier
|  +--ro f10BgpM2CfgConfederationRouter?    snmpv2-tc:TruthValue
|  +--ro f10BgpM2CfgConfederationId?
inet-address:InetAutonomousSystemNumber
|  +--ro f10BgpM2CfgPeerNextIndex?          int32
|  +--ro f10BgpM2PathAttrCount?            yang:counter32
```

Module: f10BgpM2VersionTable

URI: /api/operational/f10BgpM2VersionTable

Methods Supported: GET

```
+--ro f10BgpM2VersionTable
|  +--ro f10BgpM2VersionEntry [f10BgpM2VersionIndex]
|      +--ro f10BgpM2VersionIndex      uint32
|      +--ro f10BgpM2VersionSupported?  snmpv2-tc:TruthValue
```

Module: f10BgpM2SupportedCapabilitiesTable

URI: /api/operational/f10BgpM2SupportedCapabilitiesTable

Methods Supported: GET

```
+--ro f10BgpM2SupportedCapabilitiesTable
|  +--ro f10BgpM2SupportedCapabilitiesEntry [f10BgpM2SupportedCapabilityCode]
|    +--ro f10BgpM2SupportedCapabilityCode      uint32
|    +--ro f10BgpM2SupportedCapability?         snmpv2-tc:TruthValue
```

Module: f10BgpM2PeerTable

URI: /api/operational/f10BgpM2PeerTable

Methods Supported: GET

```
+--ro f10BgpM2PeerTable
|  +--ro f10BgpM2PeerEntry [f10BgpM2PeerInstance f10BgpM2PeerLocalAddr
|  f10BgpM2PeerRemoteAddr]
|    +--ro f10BgpM2PeerInstance                uint32
|    +--ro f10BgpM2PeerIdentifier?              inet:ip-address
|    +--ro f10BgpM2PeerState?                  enumeration
|    +--ro f10BgpM2PeerStatus?                 enumeration
|    +--ro f10BgpM2PeerConfiguredVersion?      uint32
|    +--ro f10BgpM2PeerNegotiatedVersion?      uint32
|    +--ro f10BgpM2PeerLocalAddr               inet:ip-address
|    +--ro f10BgpM2PeerLocalPort?              inet-address:InetPortNumber
|    +--ro f10BgpM2PeerLocalAs?                inet-address:InetAutonomousSystemNumber
|    +--ro f10BgpM2PeerRemoteAddr              inet:ip-address
|    +--ro f10BgpM2PeerRemotePort?             inet-address:InetPortNumber
|    +--ro f10BgpM2PeerRemoteAs?              inet-address:InetAutonomousSystemNumber
|    +--ro f10BgpM2PeerIndex?                  uint32
|    +--ro f10BgpM2PeerLastErrorReceived?      binary
|    +--ro f10BgpM2PeerLastErrorSent?          binary
|    +--ro f10BgpM2PeerLastErrorReceivedTime?  yang:timeticks
|    +--ro f10BgpM2PeerLastErrorSentTime?      yang:timeticks
|    +--ro f10BgpM2PeerLastErrorReceivedText?  snmp-framework:SnmpAdminString
|    +--ro f10BgpM2PeerLastErrorSentText?      snmp-framework:SnmpAdminString
|    +--ro f10BgpM2PeerLastErrorReceivedData?  binary
|    +--ro f10BgpM2PeerLastErrorSentData?      binary
|    +--ro f10BgpM2PeerFsmEstablishedTime?     yang:gauge32
|    +--ro f10BgpM2PeerInUpdatesElapsedTime?  yang:gauge32
```

	+++ro f10BgpM2PeerConnectRetryInterval?	uint32
	+++ro f10BgpM2PeerHoldTimeConfigured?	uint32
	+++ro f10BgpM2PeerKeepAliveConfigured?	uint32
	+++ro f10BgpM2PeerMinASOrigInterval?	uint32
	+++ro f10BgpM2PeerMinRouteAdverInterval?	uint32
	+++ro f10BgpM2PeerHoldTime?	uint32
	+++ro f10BgpM2PeerKeepAlive?	uint32
	+++ro f10BgpM2PeerInUpdates?	yang:counter32
	+++ro f10BgpM2PeerOutUpdates?	yang:counter32
	+++ro f10BgpM2PeerInTotalMessages?	yang:counter32
	+++ro f10BgpM2PeerOutTotalMessages?	yang:counter32
	+++ro f10BgpM2PeerFsmEstablishedTrans?	yang:counter32
	+++ro f10BgpM2PeerInKeepalives?	yang:counter32
	+++ro f10BgpM2PeerOutKeepalives?	yang:counter32
	+++ro f10BgpM2PeerInOpen?	yang:counter32
	+++ro f10BgpM2PeerOutOpen?	yang:counter32
	+++ro f10BgpM2PeerInRteRefresh?	yang:counter32
	+++ro f10BgpM2PeerOutRteRefresh?	yang:counter32
	+++ro f10BgpM2PeerReflectorClient?	enumeration
	+++ro f10BgpM2PeerConfedMember?	snmpv2-tc:TruthValue
	+++ro f10BgpM2CfgPeerConfedMember?	snmpv2-tc:TruthValue
	+++ro f10BgpM2PeerGroupName?	string

Module: f10BgpM2PeerCapsAnnouncedTable**URI:** /api/operational/f10BgpM2PeerCapsAnnouncedTable**Methods Supported:** GET

```
+++ro f10BgpM2PeerCapsAnnouncedTable
|   +++ro f10BgpM2PeerCapsAnnouncedEntry [f10BgpM2PeerIndex
f10BgpM2PeerCapAnnouncedCode f10BgpM2PeerCapAnnouncedIndex]
|       +++ro f10BgpM2PeerIndex                uint32
|       +++ro f10BgpM2PeerCapAnnouncedCode    uint32
|       +++ro f10BgpM2PeerCapAnnouncedIndex   uint32
|       +++ro f10BgpM2PeerCapAnnouncedValue?  binary
```

Module: f10BgpM2/f10BgpM2PeerCapsReceivedTable**URI:** /api/operational/f10BgpM2PeerCapsReceivedTable**Methods Supported:** GET

```
+++ro f10BgpM2PeerCapsReceivedTable
|   +++ro f10BgpM2PeerCapsReceivedEntry [f10BgpM2PeerIndex
f10BgpM2PeerCapReceivedCode f10BgpM2PeerCapReceivedIndex]
```

	+-ro f10BgpM2PeerIndex	uint32
	+-ro f10BgpM2PeerCapReceivedCode	uint32
	+-ro f10BgpM2PeerCapReceivedIndex	uint32
	+-ro f10BgpM2PeerCapReceivedValue?	binary

Module: f10BgpM2PrefixCountersTable

URI: /api/operational/f10BgpM2PrefixCountersTable

Methods Supported: GET

```

+-ro f10BgpM2PrefixCountersTable
| +-ro f10BgpM2PrefixCountersEntry [f10BgpM2PeerIndex
f10BgpM2PrefixCountersAfi f10BgpM2PrefixCountersSafi]
| +-ro f10BgpM2PeerIndex                uint32
| +-ro f10BgpM2PrefixCountersAfi        f10-bgp4:F10BgpM2Afi
| +-ro f10BgpM2PrefixCountersSafi       f10-bgp4:F10BgpM2Safi
| +-ro f10BgpM2PrefixInPrefixes?       yang:gauge32
| +-ro f10BgpM2PrefixInPrefixesAccepted? yang:gauge32
| +-ro f10BgpM2PrefixInPrefixesRejected? yang:gauge32
| +-ro f10BgpM2PrefixOutPrefixes?       yang:gauge32
| +-ro f10BgpM2PrefixWdrawnByPeer?      yang:gauge32
| +-ro f10BgpM2PrefixWdrawnFromPeer?    yang:gauge32

```

Module: f10BgpM2CfgPeerAdminStatusTable

URI: /api/operational/f10BgpM2CfgPeerAdminStatusTable

Methods Supported: GET

```

+-ro f10BgpM2CfgPeerAdminStatusTable
| +-ro f10BgpM2CfgPeerAdminStatusEntry [f10BgpM2PeerIndex]
| +-ro f10BgpM2PeerIndex                uint32
| +-ro f10BgpM2CfgPeerAdminStatus?      enumeration

```

Module: f10BgpM2CfgPeerTable

URI: /api/operational/f10BgpM2CfgPeerTable

Methods Supported: GET

```

+-ro f10BgpM2CfgPeerTable
| +-ro f10BgpM2CfgPeerEntry [f10BgpM2CfgPeerIndex]
| +-ro f10BgpM2CfgPeerIndex                uint32
| +-ro f10BgpM2CfgPeerConfiguredVersion?   uint32
| +-ro f10BgpM2CfgAllowVersionNegotiation? snmpv2-tc:TruthValue
| +-ro f10BgpM2CfgPeerLocalAddr?           inet:ip-address

```

```

|      +--ro f10BgpM2CfgPeerLocalAs?
inet-address:InetAutonomousSystemNumber
|      +--ro f10BgpM2CfgPeerRemoteAddr?          inet:ip-address
|      +--ro f10BgpM2CfgPeerRemoteAs?
inet-address:InetAutonomousSystemNumber
|      +--ro f10BgpM2CfgPeerEntryStorageType?    snmpv2-tc:StorageType
|      +--ro f10BgpM2CfgPeerError?              enumeration
|      +--ro f10BgpM2CfgPeerBgpPeerEntry?       snmpv2-tc:RowPointer
|      +--ro f10BgpM2CfgPeerRowEntryStatus?     snmpv2-tc:RowStatus
|      +--ro f10BgpM2CfgPeerStatus?             enumeration
|      +--ro f10BgpM2CfgPeerConnectRetryInterval? uint32
|      +--ro f10BgpM2CfgPeerHoldTimeConfigured? uint32
|      +--ro f10BgpM2CfgPeerKeepAliveConfigured? uint32
|      +--ro f10BgpM2CfgPeerMinASOrigInterval?  uint32
|      +--ro f10BgpM2CfgPeerMinRouteAdverInter? uint32
|      +--ro f10BgpM2CfgPeerReflectorClient?    enumeration

```

Module: f10BgpM2NlriTable

URI: /api/operational/f10BgpM2NlriTable

Methods Supported: GET

```

+--ro f10BgpM2NlriTable
|  +--ro f10BgpM2NlriEntry [f10BgpM2PeerIndex f10BgpM2NlriAfi f10BgpM2NlriSafi
f10BgpM2NlriPrefix f10BgpM2NlriPrefixLen f10BgpM2NlriIndex]
|      +--ro f10BgpM2PeerIndex          uint32
|      +--ro f10BgpM2NlriIndex          uint32
|      +--ro f10BgpM2NlriAfi            f10-bgp4:F10BgpM2Afi
|      +--ro f10BgpM2NlriSafi           f10-bgp4:F10BgpM2Safi
|      +--ro f10BgpM2NlriPrefix         inet:ip-address
|      +--ro f10BgpM2NlriPrefixLen     inet-address:InetAddressPrefixLength
|      +--ro f10BgpM2NlriBest?         snmpv2-tc:TruthValue
|      +--ro f10BgpM2NlriCalcLocalPref? uint32
|      +--ro f10BgpM2PathAttrIndex?    uint32
|      +--ro f10BgpM2NlriOpaqueType?   enumeration
|      +--ro f10BgpM2NlriOpaquePointer? snmpv2-tc:RowPointer
|      +--ro f10BgpM2RouteFlag?        enumeration

```

Module: f10BgpM2AdjRibsOutTable

URI: /api/operational/f10BgpM2AdjRibsOutTable

Methods Supported: GET

```

+--ro f10BgpM2AdjRibsOutTable

```



```

| ---ro f10BgpM2AdjRibsOutEntry [f10BgpM2PeerIndex f10BgpM2NlriAfi
f10BgpM2NlriSafi f10BgpM2NlriPrefix f10BgpM2NlriPrefixLen
f10BgpM2AdjRibsOutIndex]
|   +-ro f10BgpM2PeerIndex          uint32
|   +-ro f10BgpM2NlriAfi            f10-bgp4:F10BgpM2Afi
|   +-ro f10BgpM2NlriSafi           f10-bgp4:F10BgpM2Safi
|   +-ro f10BgpM2NlriPrefix         inet:ip-address
|   +-ro f10BgpM2NlriPrefixLen     inet-address:InetAddressPrefixLength
|   +-ro f10BgpM2AdjRibsOutIndex    uint32
|   +-ro f10BgpM2AdjRibsOutRoute?   snmpv2-tc:RowPointer

```

Module: f10BgpM2PathAttrTable

URI: /api/operational/f10BgpM2PathAttrTable

Methods Supported: GET

```

+---ro f10BgpM2PathAttrTable
|   +---ro f10BgpM2PathAttrEntry [f10BgpM2PathAttrIndex]
|     +-ro f10BgpM2PathAttrIndex          uint32
|     +-ro f10BgpM2PathAttrOrigin?        enumeration
|     +-ro f10BgpM2PathAttrNextHop?       inet:ip-address
|     +-ro f10BgpM2PathAttrMedPresent?    snmpv2-tc:TruthValue
|     +-ro f10BgpM2PathAttrMed?          uint32
|     +-ro f10BgpM2PathAttrLocalPrefPresent? snmpv2-tc:TruthValue
|     +-ro f10BgpM2PathAttrLocalPref?    uint32
|     +-ro f10BgpM2PathAttrAtomicAggregate? enumeration
|     +-ro f10BgpM2PathAttrAggregatorAS?
inet-address:InetAddressAutonomousSystemNumber
|     +-ro f10BgpM2PathAttrAggregatorAddr? f10-bgp4:F10BgpM2Identifier
|     +-ro f10BgpM2AsPathCalcLength?      uint32
|     +-ro f10BgpM2AsPathString?          snmp-framework:SnmpAdminString
|     +-ro f10BgpM2AsPathIndex?          uint32
|     +-ro f10BgpM2AsPath4bytePathPresent? snmpv2-tc:TruthValue
|     +-ro f10BgpM2AsPath4byteAggregatorAS?
inet-address:InetAddressAutonomousSystemNumber
|     +-ro f10BgpM2AsPath4byteCalcLength? uint32
|     +-ro f10BgpM2AsPath4byteString?     snmp-framework:SnmpAdminString
|     +-ro f10BgpM2AsPath4byteIndex?     uint32

```

Module: f10BgpM2AsPathTable

URI: /api/operational/f10BgpM2AsPathTable

Methods Supported: GET

```
+--ro f10BgpM2AsPathTable
|  +--ro f10BgpM2AsPathTableEntry [f10BgpM2PathAttrIndex
f10BgpM2AsPathSegmentIndex f10BgpM2AsPathElementIndex]
|    +--ro f10BgpM2PathAttrIndex      uint32
|    +--ro f10BgpM2AsPathSegmentIndex  uint32
|    +--ro f10BgpM2AsPathElementIndex  uint32
|    +--ro f10BgpM2AsPathType?         enumeration
|    +--ro f10BgpM2AsPathElementValue?
inet-address:InetAutonomousSystemNumber
```

Module: f10BgpM2PathAttrUnknownTable

URI: /api/operational/f10BgpM2PathAttrUnknownTable

Methods Supported: GET

```
+--ro f10BgpM2PathAttrUnknownTable
|  +--ro f10BgpM2PathAttrUnknownEntry [f10BgpM2PathAttrIndex
f10BgpM2PathAttrUnknownIndex]
|    +--ro f10BgpM2PathAttrIndex      uint32
|    +--ro f10BgpM2PathAttrUnknownIndex  uint32
|    +--ro f10BgpM2PathAttrUnknownType?  uint32
|    +--ro f10BgpM2PathAttrUnknownValue? binary
```

Module: f10BgpM2PathAttrCommTable

URI: /api/operational/f10BgpM2PathAttrCommTable

Methods Supported: GET

```
+--ro f10BgpM2PathAttrCommTable
|  +--ro f10BgpM2PathAttrCommEntry [f10BgpM2PathAttrIndex
f10BgpM2PathAttrCommIndex]
|    +--ro f10BgpM2PathAttrIndex      uint32
|    +--ro f10BgpM2PathAttrCommIndex  uint32
|    +--ro f10BgpM2PathAttrCommValue? f10-bgp4:F10BgpM2Community
```

Module: f10BgpM2LinkLocalNextHopTable

URI: /api/operational/f10BgpM2LinkLocalNextHopTable

Methods Supported: GET

```
+--ro f10BgpM2LinkLocalNextHopTable
|  +--ro f10BgpM2LinkLocalNextHopEntry [f10BgpM2PathAttrIndex]
```

```

|   +--ro f10BgpM2PathAttrIndex          uint32
|   +--ro f10BgpM2LinkLocalNextHopPresent?  snmpv2-tc:TruthValue
|   +--ro f10BgpM2LinkLocalNextHop?        inet-address:InetAddress

```

Module: f10BgpM2PathAttrOriginatorIdTable

URI: /api/operational/f10BgpM2PathAttrOriginatorIdTable

Methods Supported: GET

```

+--ro f10BgpM2PathAttrOriginatorIdTable
|  +--ro f10BgpM2PathAttrOriginatorIdEntry [f10BgpM2PathAttrIndex]
|  +--ro f10BgpM2PathAttrIndex          uint32
|  +--ro f10BgpM2PathAttrOriginatorId?  inet:ip-address

```

Module: f10BgpM2PathAttrClusterTable

URI: /api/operational/f10BgpM2PathAttrClusterTable

Methods Supported: GET

```

+--ro f10BgpM2PathAttrClusterTable
|  +--ro f10BgpM2PathAttrClusterEntry [f10BgpM2PathAttrIndex
f10BgpM2PathAttrClusterIndex]
|  +--ro f10BgpM2PathAttrIndex          uint32
|  +--ro f10BgpM2PathAttrClusterIndex  uint32
|  +--ro f10BgpM2PathAttrClusterValue?  f10-bgp4:F10BgpM2Identifier

```

Module: f10BgpM2PathAttrExtCommTable

URI: /api/operational/f10BgpM2PathAttrExtCommTable

Methods Supported: GET

```

+--ro f10BgpM2PathAttrExtCommTable
|  +--ro f10BgpM2PathAttrExtCommEntry [f10BgpM2PathAttrIndex
f10BgpM2PathAttrExtCommIndex]
|  +--ro f10BgpM2PathAttrIndex          uint32
|  +--ro f10BgpM2PathAttrExtCommIndex  uint32
|  +--ro f10BgpM2PathAttrExtCommValue?  f10-bgp4:F10BgpM2ExtendedCommunity

```

Module: f10BgpM2FlapStatisticsTable

URI: /api/operational/f10BgpM2FlapStatisticsTable

Methods Supported: GET

```

+--ro f10BgpM2FlapStatisticsTable
  +--ro f10BgpM2FlapStatisticsEntry [NetworkPrefix NetworkPrefixLen
NextHopAddress]

```

```

+--ro NetworkPrefix          inet:ip-address
+--ro NetworkPrefixLen      inet-address:InetAddressPrefixLength
+--ro NextHopAddress         inet:ip-address
+--ro StatusCode?           enumeration
+--ro BestRoute?            snmpv2-tc:TruthValue
+--ro PathSource?           enumeration
+--ro OriginCode?           enumeration
+--ro RouteFlapCount?       uint32
+--ro RouteFlapDuration?    yang:timeticks
+--ro ReuseDuration?        yang:timeticks
+--ro AsPathString?         snmp-framework:SnmpAdminString

```

Notifications

Module: f10BgpM2Established

URI: /api/operational/f10BgpM2Established

Methods Supported: GET

```

---n f10BgpM2Established
+--ro object-1
| +--ro f10BgpM2PeerInstance?    uint32
| +--ro f10BgpM2PeerLocalAddr?   inet:ip-address
| +--ro f10BgpM2PeerRemoteAddr?  inet:ip-address
+--ro object-2
| +--ro f10BgpM2PeerInstance?    uint32
| +--ro f10BgpM2PeerRemoteAddr?  inet:ip-address
| +--ro f10BgpM2PeerLocalAddr?   inet:ip-address
+--ro object-3
| +--ro f10BgpM2PeerInstance?    uint32
| +--ro f10BgpM2PeerLocalAddr?   inet:ip-address
| +--ro f10BgpM2PeerRemoteAddr?  inet:ip-address
+--ro object-4
| +--ro f10BgpM2PeerInstance?    uint32
| +--ro f10BgpM2PeerLocalAddr?   inet:ip-address
| +--ro f10BgpM2PeerRemoteAddr?  inet:ip-address
+--ro object-5
| +--ro f10BgpM2PeerInstance?    uint32
| +--ro f10BgpM2PeerLocalAddr?   inet:ip-address
| +--ro f10BgpM2PeerRemoteAddr?  inet:ip-address
| +--ro f10BgpM2PeerLastErrorReceived?  binary
+--ro object-6

```

```

+--ro f10BgpM2PeerInstance?      uint32
+--ro f10BgpM2PeerLocalAddr?     inet:ip-address
+--ro f10BgpM2PeerRemoteAddr?   inet:ip-address
+--ro f10BgpM2PeerState?        enumeration

```

Module: f10BgpM2BackwardTransition

URI: /api/operational/f10BgpM2BackwardTransition

Methods Supported: GET

```

--n f10BgpM2BackwardTransition
+--ro object-1
| +--ro f10BgpM2PeerInstance?      uint32
| +--ro f10BgpM2PeerLocalAddr?     inet:ip-address
| +--ro f10BgpM2PeerRemoteAddr?   inet:ip-address
+--ro object-2
| +--ro f10BgpM2PeerInstance?      uint32
| +--ro f10BgpM2PeerRemoteAddr?   inet:ip-address
| +--ro f10BgpM2PeerLocalAddr?     inet:ip-address
+--ro object-3
| +--ro f10BgpM2PeerInstance?      uint32
| +--ro f10BgpM2PeerLocalAddr?     inet:ip-address
| +--ro f10BgpM2PeerRemoteAddr?   inet:ip-address
+--ro object-4
| +--ro f10BgpM2PeerInstance?      uint32
| +--ro f10BgpM2PeerLocalAddr?     inet:ip-address
| +--ro f10BgpM2PeerRemoteAddr?   inet:ip-address
+--ro object-5
| +--ro f10BgpM2PeerInstance?      uint32
| +--ro f10BgpM2PeerLocalAddr?     inet:ip-address
| +--ro f10BgpM2PeerRemoteAddr?   inet:ip-address
| +--ro f10BgpM2PeerLastErrorReceived?  binary
+--ro object-6
| +--ro f10BgpM2PeerInstance?      uint32
| +--ro f10BgpM2PeerLocalAddr?     inet:ip-address
| +--ro f10BgpM2PeerRemoteAddr?   inet:ip-address
| +--ro f10BgpM2PeerLastErrorReceivedText?  snmp-framework:SnmpAdminString
+--ro object-7
+--ro f10BgpM2PeerInstance?      uint32
+--ro f10BgpM2PeerLocalAddr?     inet:ip-address
+--ro f10BgpM2PeerRemoteAddr?   inet:ip-address
+--ro f10BgpM2PeerState?        enumeration

```

Forwarding Plane Statistics

The following definition is for displaying the **forwarding plane statistics**.

Module: fp-stats

URI: /api/operational/fp-stats

Methods Supported: GET

```
+--ro fp-stats
|  +--ro fp-stats-entry [stackUnitId]
|    +--ro stackUnitId      int32
|    +--ro rxHandle?        int32
|    +--ro numMsgHdr?        int32
|    +--ro numMsgBuf?        int32
|    +--ro numCluster?      int32
|    +--ro received?        int32
|    +--ro dropped?         int32
|    +--ro rxToNetwork?     int32
|    +--ro rxError?         int32
|    +--ro rxDatapathError? int32
|    +--ro rxPktCOS0?       int32
|    +--ro rxPktCOS1?       int32
|    +--ro rxPktCOS2?       int32
|    +--ro rxPktCOS3?       int32
|    +--ro rxPktCOS4?       int32
|    +--ro rxPktCOS5?       int32
|    +--ro rxPktCOS6?       int32
|    +--ro rxPktCOS7?       int32
|    +--ro rxPktUnit0?      int32
|    +--ro rxPktUnit1?      int32
|    +--ro rxPktUnit2?      int32
|    +--ro rxPktUnit3?      int32
|    +--ro transmitted?     int32
|    +--ro txRequested?     int32
|    +--ro txDescriptor?    int32
|    +--ro txError?         int32
|    +--ro txReqTooLarge?   int32
|    +--ro txInternalError? int32
|    +--ro txDatapathError? int32
|    +--ro txPktCOS0?       int32
```

```

|   +--ro txPktCOS1?           int32
|   +--ro txPktCOS2?           int32
|   +--ro txPktCOS3?           int32
|   +--ro txPktCOS4?           int32
|   +--ro txPktCOS5?           int32
|   +--ro txPktCOS6?           int32
|   +--ro txPktCOS7?           int32
|   +--ro txPktUnit0?          int32
|   +--ro txPktUnit1?          int32
|   +--ro txPktUnit2?          int32
|   +--ro txPktUnit3?          int32

```

Module: fp-stats

URI: /api/operational/fp-cpu-party-bus-stats

Methods Supported: GET

```

+--ro fp-cpu-party-bus-stats
|   +--ro fp-cpu-party-bus-stats-entry [stackUnitId]
|       +--ro stackUnitId          int32
|       +--ro inputPackets?        int32
|       +--ro inputBytes?          int32
|       +--ro inputDropped?        int32
|       +--ro inputError?          int32
|       +--ro outputPackets?       int32
|       +--ro outputBytes?         int32
|       +--ro outputError?         int32

```

Module: fp-stats

URI: /api/operational/fp-drops

Methods Supported: GET

```

+--ro fp-drops
|   +--ro fp-drops-entry [stackUnitId stackPortId]
|       +--ro stackUnitId          int32
|       +--ro stackPortId          int32
|       +--ro inDrops?              yang:counter64
|       +--ro inIBPCBPFfullDrops?  yang:counter64
|       +--ro inPortSTPnotFwdDrops? yang:counter64
|       +--ro inIPv4L3Discards?     yang:counter64
|       +--ro inPolicyDiscards?     yang:counter64
|       +--ro inPktDroppedByFP?    yang:counter64

```

	+++ro inL2L3Drops?	yang:counter64
	+++ro inPortBitMapZeroDrops?	yang:counter64
	+++ro inRxVLANDrops?	yang:counter64
	+++ro inFCSDrops?	yang:counter64
	+++ro inMTUExceeds?	yang:counter64
	+++ro mmuHoldDrops?	yang:counter64
	+++ro mmuTxPurgeCellErr?	yang:counter64
	+++ro mmuAgedDrops?	yang:counter64
	+++ro egressFCSDrops?	yang:counter64
	+++ro egIPv4L3UCAgedDrops?	yang:counter64
	+++ro egTTLThresholdDrops?	yang:counter64
	+++ro egInvalidVLANCounterDrops?	yang:counter64
	+++ro egL2MCDrops?	yang:counter64
	+++ro egPktDropsOfAnyCondition?	yang:counter64
	+++ro egHgMacUnderFlow?	yang:counter64
	+++ro egTxErrPktCounter?	yang:counter64

Module: fp-stats**URI:** /api/operational/fp-packet-buffer**Methods Supported:** GET

```
+++ro fp-packet-buffer
|  +++ro fp-packet-buffer-entry [stackUnitId portPipe]
|    +++ro stackUnitId          int32
|    +++ro portPipe             int32
|    +++ro totalPktBuffer?      int32
|    +++ro currentAvailableBuffer? int32
|    +++ro packetBufferAlloc?   int32
```

Module: fp-stats**URI:** /api/operational/fp-port-stats**Methods Supported:** GET

```
+++ro fp-port-stats
|  +++ro fp-port-stats-entry [stackUnitId stackPortId]
|    +++ro stackUnitId          int32
|    +++ro stackPortId          int32
|    +++ro currentUsagePerPort? int32
|    +++ro defaultPktBufferAlloc? int32
|    +++ro maxLimitPerPort?     int32
```


Module: fp-stats

URI: /api/operational/fp-cos-stats

Methods Supported: GET

```
+--ro fp-cos-stats
  +--ro fp-cos-stats-entry [stackUnitId stackPortId stackPortCOSId]
    +--ro stackUnitId          int32
    +--ro stackPortId          int32
    +--ro stackPortCOSId       int32
    +--ro currentUsagePerCOS?  int32
    +--ro defaultPktBufferAlloc? int32
    +--ro maxLimitPerCOS?      int32
```

IETF Interfaces

The following definition is for configuring and displaying the properties of an interfaces using the **IETF**.

Module: ietf-interfaces

URI: /api/operational/interfaces

Methods Supported:

```
+--rw interfaces
  +--rw interface [name]
    +--rw name          string
    +--rw description?  string
    +--ro type?         ianaift:iana-if-type
    +--ro location?     string
    +--rw enabled?      boolean
    +--ro if-index?     int32
    +--rw mtu?          uint32
    +--rw link-up-down-trap-enable? enumeration
    +--rw rate-interval? uint32
```

IF MIB

The following definition is for displaying the properties of interfaces using **IF-MIB**.

Module: IF-MIB

URI: /api/operational/if-mib:interfaces

Methods Supported: GET

```

+--ro interfaces
|  +--ro ifNumber?   int32
|  +--ro ifEntry [ifIndex]
|      +--ro ifIndex                               if-mib:InterfaceIndex
|      +--ro ifDescr?                               smiv2:DisplayString
|      +--ro ifType?                               ianaiftype-mib:IANAifType
|      +--ro ifMtu?                                 int32
|      +--ro ifSpeed?                              yang:gauge32
|      +--ro ifPhysAddress?                       yang:phys-address
|      +--ro ifAdminStatus?                       enumeration
|      +--ro ifOperStatus?                       enumeration
|      +--ro ifLastChange?                       yang:timeticks
|      +--ro ifInOctets?                          yang:counter32
|      +--ro ifInUcastPkts?                      yang:counter32
|      x--ro ifInNUcastPkts?                    yang:counter32
|      +--ro ifInDiscards?                      yang:counter32
|      +--ro ifInErrors?                       yang:counter32
|      +--ro ifInUnknownProtos?                yang:counter32
|      +--ro ifOutOctets?                       yang:counter32
|      +--ro ifOutUcastPkts?                   yang:counter32
|      x--ro ifOutNUcastPkts?                  yang:counter32
|      +--ro ifOutDiscards?                   yang:counter32
|      +--ro ifOutErrors?                     yang:counter32
|      x--ro ifOutQLen?                        yang:gauge32
|      x--ro ifSpecific?                      yang:object-identifier
|      +--ro ifName?                            smiv2:DisplayString
|      +--ro ifInMulticastPkts?                yang:counter32
|      +--ro ifInBroadcastPkts?               yang:counter32
|      +--ro ifOutMulticastPkts?              yang:counter32
|      +--ro ifOutBroadcastPkts?              yang:counter32
|      +--ro ifHCInOctets?                    yang:counter64
|      +--ro ifHCInUcastPkts?                 yang:counter64
|      +--ro ifHCInMulticastPkts?             yang:counter64
|      +--ro ifHCInBroadcastPkts?             yang:counter64
|      +--ro ifHCOctets?                      yang:counter64
|      +--ro ifHCOUcastPkts?                  yang:counter64
|      +--ro ifHCOmulticastPkts?              yang:counter64
|      +--ro ifHCObroadcastPkts?              yang:counter64
|      +--ro ifLinkUpDownTrapEnable?          enumeration

```

```

|   +--ro ifHighSpeed?                yang:gauge32
|   +--ro ifPromiscuousMode?          smiv2:TruthValue
|   +--ro ifConnectorPresent?         smiv2:TruthValue
|   +--ro ifAlias?                    smiv2:DisplayString
|   +--ro ifCounterDiscontinuityTime? yang:timestamp
|   x--ro ifTestId?                   smiv2:TestAndIncr
|   x--ro ifTestStatus?               enumeration
|   x--ro ifTestType?                smiv2:AutonomousType
|   x--ro ifTestResult?               enumeration
|   x--ro ifTestCode?                yang:object-identifier
|   x--ro ifTestOwner?               if-mib:OwnerString

```

Module: IF-MIB

URI: /api/operational/ifMIBObjects

Methods Supported: GET

```

+--ro ifMIBObjects
  +--ro ifStackEntry [ifStackHigherLayer ifStackLowerLayer]
    | +--ro ifStackHigherLayer   if-mib:InterfaceIndexOrZero
    | +--ro ifStackLowerLayer    if-mib:InterfaceIndexOrZero
    | +--ro ifStackStatus?       smiv2:RowStatus
    +--ro ifRcvAddressEntry [ifIndex ifRcvAddressAddress]
      | +--ro ifIndex            leafref
      | +--ro ifRcvAddressAddress yang:phys-address
      | +--ro ifRcvAddressStatus? smiv2:RowStatus
      | +--ro ifRcvAddressType?  enumeration
      +--ro ifTableLastChange?  yang:timeticks
      +--ro ifStackLastChange?  yang:timeticks

+--ro ifMIBObjects
  +--ro ifStackEntry                [ifStackHigherLayer ifStackLowerLayer]
    | +--ro ifStackHigherLayer      if-mib:InterfaceIndexOrZero
    | +--ro ifStackLowerLayer       if-mib:InterfaceIndexOrZero
    | +--ro ifStackStatus?          smiv2:RowStatus
    +--ro ifRcvAddressEntry          [ifIndex ifRcvAddressAddress]
      | +--ro ifIndex               leafref
      | +--ro ifRcvAddressAddress    yang:phys-address
      | +--ro ifRcvAddressStatus?    smiv2:RowStatus
      | +--ro ifRcvAddressType?     enumeration
      +--ro ifTableLastChange?      yang:timeticks
      +--ro ifStackLastChange?      yang:timeticks

```

Notifications**Module:** IF-MIB**URI:** /api/operational/linkDown**Methods Supported:** GET

```

+---n linkDown
|  +---ro linkDown-ifIndex
|  |  +---ro ifIndex?  leafref
|  +---ro linkDown-ifAdminStatus
|  |  +---ro ifIndex?      leafref
|  |  +---ro ifAdminStatus? enumeration
|  +---ro linkDown-ifOperStatus
|    +---ro ifIndex?      leafref
|    +---ro ifOperStatus? enumeration

```

Module: IF-MIB**URI:** /api/operational/linkUp**Methods Supported:** GET

```

+---n linkUp
  +---ro linkUp-ifIndex
  |  +---ro ifIndex?  leafref
+---ro linkUp-ifAdminStatus
  |  +---ro ifIndex?      leafref
  |  +---ro ifAdminStatus? enumeratio
+---ro linkUp-ifOperStatus
  +---ro ifIndex?      leafref
  +---ro ifOperStatus? enumeration

```

REST API CLI

Overview

The REST API CLI command is supported on the **S4810**, **S4820T**, **S6000**, and **Z9000** platforms.

rest-server

S4810 **S4820T**

S6000 **Z**

Enable the REST API service over a non-secure or secure HTTP.

To disable the REST API over a non-secure or secure HTTP request, use the **no rest-server {http | secure-http}** command.

Syntax

rest-server {http | secure-http}

Defaults

REST API is disabled by default.

Command Modes

CONFIGURATION

Parameters

http	Enable the REST API on HTTP (Port: 8008).
secure-http	Enable the REST API on HTTPS (Port: 8888).

Command History

Version 9.4(0.0)	Introduced on the S4810, S4820T, S6000, and Z9000.
------------------	--



Note: The **rest enable** command is deprecated, however the support is maintained for backward compatibility in version 9.4(0.0) and will be removed in the future release.

Web Server with HTTP Support

Web Server with HTTP Support are downloaded with the SmartScripts package (see [Downloading the Smart Scripting Package](#)). It is supported on the **S4810**, **S4820T**, **Z9000** and **MXL** switch platforms.

This chapter describes the Web-based components in the Open Automation package:

- [Web Server](#)

Web Server

In the Open Automation package, the web server runs on a switch and handles HTTP and HTTPS requests. You can start the web server in a non-secure (HTTP) or secure (HTTPS) mode.

To start the web server in a non-secure (without SSL) mode for receiving HTTP requests and write the configuration to the running configuration, use the **http-server http** command:

Command Syntax	Command Mode	Task
http-server http	CONFIGURATION	Starts the web-server application in non-secure mode to receive HTTP requests.

To start the web server in a secure mode for receiving HTTPS requests and write the configuration to the running configuration, use the **http-server secure-http** command:

Command Syntax	Command Mode	Task
http-server secure-http	CONFIGURATION	Starts the web-server application in secure mode using SSL to receive HTTPS requests.

To stop the web server and remove the configuration from the running-configuration file, use the **no http-server {http | secure-http}** command.

Web Graphical User Interface

This appendix contains examples of the output displayed for each menu option in the Web interface used in the Open Automation Framework for the menus:

- System
- Interfaces
- Protocols
- Diagnostics
- Utilities
- Settings

System Menu

System > Software Version

SOFTWARE VERSION

```
Force10 Networks Real Time Operating System Software
Force10 Operating System Version: 1.0
Force10 Application Software Version: E8-3-10-101
Copyright (c) 1999-2011 by Force10 Networks, Inc.
Build Time: Tue Oct 25 00:45:41 PDT 2011
Build Path: /sites/sjc/work/build/buildSpaces/build09/E8-3-10/SW/SRC/Cp_src/Tacacs
Force10 uptime is 1 hour(s), 0 minute(s)
```

```
System image file is "/tftpboot/arir/FTOS-SE-8-3-10-101.bin"
```

```
System Type: S4810
Control Processor: Freescale QorIQ P2020 with 2147483648 bytes of memory.
```

```
128M bytes of boot flash memory.
```

```
1 52-port GE/TE/FG (SE)
48 Ten GigabitEthernet/IEEE 802.3 interface(s)
4 Forty GigabitEthernet/IEEE 802.3 interface(s)
```

System > Time/Date

CURRENT DATE

```
03:09:46.883 PST Sat Nov 12 2011
```

System > Memory Usage

MEMORY USAGE

```
Statistics On Unit 0 Processor
=====
Total (b)      Used (b)      Free (b)      Lowest (b)    Largest (b)
2147483648    3825202      2143658446    2143641882    2143658446
```

System > CPU Usage

CPU USAGE

CPU Statistics Of Unit 0

=====

CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%

PID	Runtime (ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
0x42caa000	60	6	10000	0.00%	0.00%	0.00%	0	diagagt
0x42c88000	0	0	0	0.00%	0.00%	0.00%	0	debugagt
0x42c67000	0	0	0	0.00%	0.00%	0.00%	0	F10StkMgr
0x42c44000	8240	824	10000	0.00%	0.00%	0.02%	0	lcMgr
0x42c1e000	20	2	10000	0.00%	0.00%	0.00%	0	dla
0x42bf9000	440	44	10000	0.00%	0.00%	0.00%	0	sysAdmTsk
0x42bd8000	3070	307	10000	0.00%	0.00%	0.00%	0	timerMgr
0x42bb5000	5880	588	10000	0.00%	0.00%	0.00%	0	PM
0x42b91000	6200	620	10000	0.00%	0.00%	0.00%	0	KP
0x42b6e000	10	1	10000	0.00%	0.00%	0.00%	0	evagt
0x42b48000	250	25	10000	0.00%	0.00%	0.00%	0	ipc
0x41e1e000	210	21	10000	0.00%	0.00%	0.00%	0	tme
0x41e1c000	0	0	0	0.00%	0.00%	0.00%	0	ttraceIpFlow
0x41e19000	0	0	0	0.00%	0.00%	0.00%	0	linkscan_user_threa
0x41df9000	0	0	0	0.00%	0.00%	0.00%	0	tDDB
0x41df6000	0	0	0	0.00%	0.00%	0.00%	0	GC
0x41df2000	0	0	0	0.00%	0.00%	0.00%	0	isrTask
0x41de9000	30	3	10000	0.00%	0.00%	0.00%	0	bshell_reaper_threa
0x41de0000	0	0	0	0.00%	0.00%	0.00%	0	tSysLog
0x41dde000	420	42	10000	0.00%	0.00%	0.00%	0	tTimerTask
0x41ddc000	7630	763	10000	0.00%	0.00%	0.00%	0	tExcTask
0x41dca000	0	0	0	0.00%	0.00%	0.00%	0	tLogTask
0x41dc4000	43120	4312	10000	0.00%	0.00%	0.00%	0	tUsrRoot
0x41d80000	10	1	10000	0.00%	0.00%	0.00%	0	main
0x43147000	0	0	0	0.00%	0.00%	0.00%	0	tFib6audit
0x42f95000	170	17	10000	0.00%	0.00%	0.00%	0	igmpAgent
0x42f92000	100	10	10000	0.00%	0.00%	0.00%	0	tFib6spf
0x42f60000	16850	1685	10000	0.00%	0.08%	0.03%	0	l2LrnAgeMove
0x42efe000	0	0	0	0.00%	0.00%	0.00%	0	fib6
0x42ed3000	1300	130	10000	0.00%	0.00%	0.00%	0	MacAgent
0x42eb1000	11400	1140	10000	0.00%	0.00%	0.02%	0	frrpagt
0x42e7b000	700	70	10000	0.00%	0.00%	0.00%	0	dsagt
0x42e58000	0	0	0	0.00%	0.00%	0.00%	0	tFib4audit
0x42d7c000	0	0	0	0.00%	0.00%	0.00%	0	ifaDispatch
0x42d62000	5750	575	10000	0.00%	0.00%	0.00%	0	ifagt_1
0x42d25000	130	13	10000	0.00%	0.00%	0.00%	0	tFib4spf
0x42d23000	330	33	10000	0.00%	0.00%	0.00%	0	aclAgent
0x42cf9000	10	1	10000	0.00%	0.00%	0.00%	0	tFib4
0x42c42000	90	9	10000	0.00%	0.00%	0.00%	0	count
0x4336e000	0	0	0	0.00%	0.00%	0.00%	0	frrpaRecv

System > Boot Variables

BOOT VARIABLES

```
PRIMARY IMAGE FILE = tftp://10.42.7.77//tftpboot/arir/FTOS-SE-8-3-10-101.bin
SECONDARY IMAGE FILE = system://A
DEFAULT IMAGE FILE = system://A
LOCAL CONFIG FILE = variable does not exist
PRIMARY HOST CONFIG FILE = variable does not exist
SECONDARY HOST CONFIG FILE = variable does not exist
PRIMARY NETWORK CONFIG FILE = variable does not exist
SECONDARY NETWORK CONFIG FILE = variable does not exist
CURRENT IMAGE FILE = tftp://10.42.7.77//tftpboot/arir/FTOS-SE-8-3-10-101.bin
CURRENT CONFIG FILE 1 = flash://startup-config
CURRENT CONFIG FILE 2 = variable does not exist
CONFIG LOAD PREFERENCE = local first
BOOT INTERFACE GATEWAY IP ADDRESS = 0.0.0.0
Reload Mode = normal-reload
```

System > Running Config

RUNNING CONFIG

```
Current Configuration ...
! Version E8-3-5-58
! Last configuration change at Thu Sep 29 17:58:15 2011 by admin
! Startup-config last updated at Tue Sep 20 00:10:37 2011 by admin
!
boot system stack-unit 0 primary tftp://10.42.7.77/FTOS-SD-8-3-5-58.bin
boot system stack-unit 0 secondary system: A:
boot system stack-unit 0 default tftp://10.42.7.77/FTOS-SD-8-3-5-58.bin
boot system gateway 10.43.0.1
!
redundancy auto-synchronize full
!
hardware watchdog
!
hostname st-s55-0a
!
enable password 7 b125455cf679b208e79b910e85789edf
!
username test password 7 7b56aef7d3a1cce8
username admin password 7 1d28e9f33f99cf5c
username admin1 password 7 3b0067cc6673eaec
!
protocol spanning-tree mstp
no disable
!
stack-unit 0 provision S55
!
interface GigabitEthernet 0/0
no ip address
shutdown
!
interface GigabitEthernet 0/1
no ip address
switchport
no shutdown
!
```

System > Information

SYSTEM INFORMATION

Hostname Force10
IP address 10.42.51.5/16
FTOS Version E8-3-10-101
Platform S4810
Uptime 57 minute(s)
Last config change Tue Nov 8 12:52:54 2011 by admin

Interfaces Menu

Interfaces > All

ALL INTERFACES							
Interface	IP-Address	OK	Method	Status			Protocol
GigabitEthernet 0/0	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/1	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/2	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/3	unassigned	YES	Manual	up		up	
GigabitEthernet 0/4	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/5	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/6	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/7	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/8	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/9	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/10	unassigned	YES	Manual	up		up	
GigabitEthernet 0/11	unassigned	YES	Manual	up		up	
GigabitEthernet 0/12	unassigned	YES	Manual	up		up	
GigabitEthernet 0/13	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/14	unassigned	YES	Manual	up		up	
GigabitEthernet 0/15	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/16	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/17	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/18	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/19	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/20	unassigned	YES	Manual	up		up	
GigabitEthernet 0/21	unassigned	YES	Manual	up		up	
GigabitEthernet 0/22	unassigned	YES	Manual	up		up	
GigabitEthernet 0/23	unassigned	YES	Manual	up		up	
GigabitEthernet 0/24	unassigned	YES	Manual	up		up	
GigabitEthernet 0/25	unassigned	NO	Manual	up		down	
GigabitEthernet 0/26	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/27	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/28	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/29	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/30	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/31	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/32	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/33	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/34	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/35	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/36	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/37	unassigned	YES	Manual	up		up	
GigabitEthernet 0/38	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/39	39.39.39.2	YES	Manual	up		up	
GigabitEthernet 0/40	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/41	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/42	unassigned	YES	Manual	up		up	
GigabitEthernet 0/43	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/44	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/45	70.70.70.1	NO	Manual	up		down	
GigabitEthernet 0/46	unassigned	NO	Manual	administratively	down	down	
GigabitEthernet 0/47	unassigned	NO	Manual	up		down	
ManagementEthernet 0/0	10.43.60.100	YES	Manual	up		up	

Interfaces > Physical

PHYSICAL INTERFACES

Interface	IP-Address	OK	Method	Status	Protocol
GigabitEthernet 0/0	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/1	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/2	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/3	unassigned	YES	Manual	up	up
GigabitEthernet 0/4	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/5	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/6	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/7	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/8	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/9	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/10	unassigned	YES	Manual	up	up
GigabitEthernet 0/11	unassigned	YES	Manual	up	up
GigabitEthernet 0/12	unassigned	YES	Manual	up	up
GigabitEthernet 0/13	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/14	unassigned	YES	Manual	up	up
GigabitEthernet 0/15	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/16	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/17	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/18	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/19	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/20	unassigned	YES	Manual	up	up
GigabitEthernet 0/21	unassigned	YES	Manual	up	up
GigabitEthernet 0/22	unassigned	YES	Manual	up	up
GigabitEthernet 0/23	unassigned	YES	Manual	up	up
GigabitEthernet 0/24	unassigned	YES	Manual	up	up
GigabitEthernet 0/25	unassigned	NO	Manual	up	down
GigabitEthernet 0/26	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/27	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/28	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/29	unassigned	NO	Manual	administratively down	down
GigabitEthernet 0/30	unassigned	NO	Manual	administratively down	down

Interfaces > LAGs

INTERFACES - LAGS

Interface	IP-Address	OK	Method	Status	Protocol
Port-channel 1	unassigned	YES	Manual	up	up
Port-channel 2	unassigned	YES	Manual	up	up

Interfaces > VLANs

INTERFACES - VLANS

Interface	IP-Address	OK	Method	Status	Protocol
Vlan 1	unassigned	NO	Manual	administratively down	down
Vlan 2	unassigned	NO	Manual	administratively down	down
Vlan 3	unassigned	NO	Manual	administratively down	down
Vlan 4	unassigned	NO	Manual	administratively down	down
Vlan 5	unassigned	NO	Manual	administratively down	down
Vlan 1000	5.5.5.3	YES	Manual	up	up
Vlan 2000	16.16.16.2	YES	Manual	up	up
Vlan 2001	16.16.17.2	YES	Manual	up	up
Vlan 3000	24.24.24.3	YES	Manual	up	up
Vlan 3001	unassigned	NO	Manual	administratively down	down
Vlan 3002	unassigned	NO	Manual	administratively down	down
Vlan 3500	unassigned	NO	Manual	administratively down	down
Vlan 4000	unassigned	YES	Manual	up	up
Vlan 4009	unassigned	NO	Manual	administratively down	down
Vlan 4011	unassigned	NO	Manual	administratively down	down
Vlan 4012	unassigned	NO	Manual	administratively down	down
Vlan 4050	42.42.42.3	YES	Manual	up	up

Interfaces > Management

INTERFACES - MANAGEMENT

Interface		IP-Address	OK	Method	Status	Protocol
ManagementEthernet	0/0	10.43.3.55	YES	Manual	up	up
ManagementEthernet	1/0	unassigned	NO	Manual	up	not present
ManagementEthernet	2/0	unassigned	NO	Manual	up	not present
ManagementEthernet	3/0	unassigned	NO	Manual	up	not present
ManagementEthernet	4/0	unassigned	NO	Manual	up	not present
ManagementEthernet	5/0	unassigned	NO	Manual	up	not present
ManagementEthernet	6/0	unassigned	NO	Manual	up	not present
ManagementEthernet	7/0	unassigned	NO	Manual	up	not present
ManagementEthernet	8/0	unassigned	NO	Manual	up	not present
ManagementEthernet	9/0	unassigned	NO	Manual	up	not present
ManagementEthernet	10/0	unassigned	NO	Manual	up	not present
ManagementEthernet	11/0	unassigned	NO	Manual	up	not present

Protocols Menu

Protocols > VRRP
PROTOCOLS - VRRP

```

-----
Vlan 100, IPv4 VRID: 1, Version: 2, Net: 88.1.1.1
State: Master, Priority: 101, Master: 88.1.1.1 (local)
Hold Down: 0 sec, Preempt: TRUE, AdvInt: 1 sec
Adv rcvd: 0, Bad pkts rcvd: 0, Adv sent: 2931984, Gratuitous ARP sent: 12
Virtual MAC address:
  00:00:5e:00:01:01
Virtual IP address:
  88.1.1.4 88.1.1.5 88.1.1.6 88.1.1.7
  88.1.1.8 88.1.1.9 88.1.1.10 88.1.1.11
  88.1.1.12 88.1.1.13 88.1.1.14 88.1.1.15
Authentication: (none)
-----

```

```

-----
Vlan 101, IPv4 VRID: 1, Version: 2, Net: 88.1.2.1
State: Master, Priority: 101, Master: 88.1.2.1 (local)
Hold Down: 0 sec, Preempt: TRUE, AdvInt: 1 sec
Adv rcvd: 0, Bad pkts rcvd: 0, Adv sent: 2930966, Gratuitous ARP sent: 12
Virtual MAC address:
  00:00:5e:00:01:01
Virtual IP address:
  88.1.2.4 88.1.2.5 88.1.2.6 88.1.2.7
  88.1.2.8 88.1.2.9 88.1.2.10 88.1.2.11
  88.1.2.12 88.1.2.13 88.1.2.14 88.1.2.15
Authentication: (none)
-----

```

```

-----
Vlan 102, IPv4 VRID: 1, Version: 2, Net: 88.1.3.1
State: Master, Priority: 101, Master: 88.1.3.1 (local)
Hold Down: 0 sec, Preempt: TRUE, AdvInt: 1 sec
Adv rcvd: 0, Bad pkts rcvd: 0, Adv sent: 2930214, Gratuitous ARP sent: 12
Virtual MAC address:
  00:00:5e:00:01:01
Virtual IP address:
  88.1.3.4 88.1.3.5 88.1.3.6 88.1.3.7
  88.1.3.8 88.1.3.9 88.1.3.10 88.1.3.11
  88.1.3.12 88.1.3.13 88.1.3.14 88.1.3.15
Authentication: (none)
-----

```

```

-----
Vlan 103, IPv4 VRID: 1, Version: 2, Net: 88.1.4.1
State: Master, Priority: 101, Master: 88.1.4.1 (local)
Hold Down: 0 sec, Preempt: TRUE, AdvInt: 1 sec
Adv rcvd: 0, Bad pkts rcvd: 0, Adv sent: 2929610, Gratuitous ARP sent: 12
Virtual MAC address:
  00:00:5e:00:01:01
Virtual IP address:
  88.1.4.4 88.1.4.5 88.1.4.6 88.1.4.7
  88.1.4.8 88.1.4.9 88.1.4.10 88.1.4.11
  88.1.4.12 88.1.4.13 88.1.4.14 88.1.4.15
Authentication: (none)
-----

```


Protocols > VRRP Brief

PROTOCOLS - VRRP BRIEF

Interface	Group	Pri	Pre	State	Master addr	Virtual addr(s)	Description
Vl 100	IPv4 1	101	Y	Master	88.1.1.1	88.1.1.4 88.1.1.5...	
Vl 101	IPv4 1	101	Y	Master	88.1.2.1	88.1.2.4 88.1.2.5...	
Vl 102	IPv4 1	101	Y	Master	88.1.3.1	88.1.3.4 88.1.3.5...	
Vl 103	IPv4 1	101	Y	Master	88.1.4.1	88.1.4.4 88.1.4.5...	
Vl 104	IPv4 1	101	Y	Master	88.1.5.1	88.1.5.4 88.1.5.5...	
Vl 105	IPv4 1	101	Y	Master	88.1.6.1	88.1.6.4 88.1.6.5...	
Vl 106	IPv4 1	101	Y	Master	88.1.7.1	88.1.7.4 88.1.7.5...	
Vl 107	IPv4 1	101	Y	Master	88.1.8.1	88.1.8.4 88.1.8.5...	
Vl 108	IPv4 1	101	Y	Master	88.1.9.1	88.1.9.4 88.1.9.5...	
Vl 109	IPv4 1	101	Y	Master	88.1.10.1	88.1.10.4 88.1.10.5...	
Vl 110	IPv4 1	101	Y	Master	88.1.11.1	88.1.11.4 88.1.11.5...	
Vl 111	IPv4 1	101	Y	Master	88.1.12.1	88.1.12.4 88.1.12.5...	
Vl 112	IPv4 1	101	Y	Master	88.1.13.1	88.1.13.4 88.1.13.5...	
Vl 113	IPv4 1	101	Y	Master	88.1.14.1	88.1.14.4 88.1.14.5...	
Vl 114	IPv4 1	101	Y	Master	88.1.15.1	88.1.15.4 88.1.15.5...	
Vl 115	IPv4 1	101	Y	Master	88.1.16.1	88.1.16.4 88.1.16.5...	
Vl 116	IPv4 1	101	Y	Master	88.1.17.1	88.1.17.4 88.1.17.5...	
Vl 117	IPv4 1	101	Y	Master	88.1.18.1	88.1.18.4 88.1.18.5...	
Vl 118	IPv4 1	101	Y	Master	88.1.19.1	88.1.19.4 88.1.19.5...	
Vl 119	IPv4 1	101	Y	Master	88.1.20.1	88.1.20.4 88.1.20.5...	
Vl 120	IPv4 1	101	Y	Master	88.1.21.1	88.1.21.4 88.1.21.5...	
Vl 121	IPv4 1	101	Y	Master	88.1.22.1	88.1.22.4 88.1.22.5...	
Vl 122	IPv4 1	101	Y	Master	88.1.23.1	88.1.23.4 88.1.23.5...	
Vl 123	IPv4 1	101	Y	Master	88.1.24.1	88.1.24.4 88.1.24.5...	
Vl 124	IPv4 1	101	Y	Master	88.1.25.1	88.1.25.4 88.1.25.5...	
Vl 125	IPv4 1	101	Y	Master	88.1.26.1	88.1.26.4 88.1.26.5...	
Vl 126	IPv4 1	101	Y	Master	88.1.27.1	88.1.27.4 88.1.27.5...	
Vl 127	IPv4 1	101	Y	Master	88.1.28.1	88.1.28.4 88.1.28.5...	
Vl 128	IPv4 1	101	Y	Master	88.1.29.1	88.1.29.4 88.1.29.5...	
Vl 129	IPv4 1	101	Y	Master	88.1.30.1	88.1.30.4 88.1.30.5...	
Vl 130	IPv4 1	101	Y	Master	88.1.31.1	88.1.31.4 88.1.31.5...	
Vl 131	IPv4 1	101	Y	Master	88.1.32.1	88.1.32.4 88.1.32.5...	
Vl 132	IPv4 1	101	Y	Master	88.1.33.1	88.1.33.4 88.1.33.5...	
Vl 133	IPv4 1	101	Y	Master	88.1.34.1	88.1.34.4 88.1.34.5...	
Vl 134	IPv4 1	101	Y	Master	88.1.35.1	88.1.35.4 88.1.35.5...	
Vl 135	IPv4 1	101	Y	Master	88.1.36.1	88.1.36.4 88.1.36.5...	
Vl 136	IPv4 1	101	Y	Master	88.1.37.1	88.1.37.4 88.1.37.5...	
Vl 137	IPv4 1	101	Y	Master	88.1.38.1	88.1.38.4 88.1.38.5...	
Vl 138	IPv4 1	101	Y	Master	88.1.39.1	88.1.39.4 88.1.39.5...	
Vl 139	IPv4 1	101	Y	Master	88.1.40.1	88.1.40.4 88.1.40.5...	
Vl 140	IPv4 1	101	Y	Master	88.1.41.1	88.1.41.4 88.1.41.5...	
Vl 141	IPv4 1	101	Y	Master	88.1.42.1	88.1.42.4 88.1.42.5...	
Vl 142	IPv4 1	101	Y	Master	88.1.43.1	88.1.43.4 88.1.43.5...	
Vl 143	IPv4 1	101	Y	Master	88.1.44.1	88.1.44.4 88.1.44.5...	
Vl 144	IPv4 1	101	Y	Master	88.1.45.1	88.1.45.4 88.1.45.5...	
Vl 145	IPv4 1	101	Y	Master	88.1.46.1	88.1.46.4 88.1.46.5...	
Vl 146	IPv4 1	101	Y	Master	88.1.47.1	88.1.47.4 88.1.47.5...	

Protocols > BGP Summary

PROTOCOLS - BGP SUMMARY

BGP router identifier 222.222.222.222, local AS number 6338
 BGP table version is 0, main routing table version 0
 2 neighbor(s) using 12288 bytes of memory

Neighbor	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/Pfx
5.5.5.1	6338	10083	10070	0	0	0	1w0d	0
70.70.70.2	4383	0	0	0	0	0	never	Active

Protocols > BGP Neighbors

PROTOCOLS - BGP NEIGHBORS

```

BGP neighbor is 5.5.5.1, remote AS 6338, internal link
  BGP version 4, remote router ID 223.223.223.223
  BGP state ESTABLISHED, in this state for 1w0d
  Last read 00:00:41, last write 00:00:47
  Hold time is 180, keepalive interval is 60 seconds
  Received 10083 messages, 0 in queue
    1 opens, 0 notifications, 0 updates
    10082 keepalives, 0 route refresh requests
  Sent 10070 messages, 0 in queue
    1 opens, 0 notifications, 0 updates
    10069 keepalives, 0 route refresh requests
  Minimum time between advertisement runs is 5 seconds
  Minimum time before advertisements start is 0 seconds

Capabilities received from neighbor for IPv4 Unicast :
  MULTIPROTO_EXT(1)
  ROUTE_REFRESH(2)
  CISCO_ROUTE_REFRESH(128)

Capabilities advertised to neighbor for IPv4 Unicast :
  MULTIPROTO_EXT(1)
  ROUTE_REFRESH(2)
  CISCO_ROUTE_REFRESH(128)

For address family: IPv4 Unicast
  BGP table version 0, neighbor version 0
  Prefixes accepted 0 (consume 0 bytes), withdrawn 0 by peer, martian prefixes ignored 0
  Prefixes advertised 0, denied 0, withdrawn 0 from peer

Connections established 1; dropped 0
  Last reset never
Local host: 5.5.5.3, Local port: 55170
Foreign host: 5.5.5.1, Foreign port: 179

```

Protocols > OSPF Neighbors

PROTOCOLS - OSPF NEIGHBORS

Neighbor ID	Pri	State	Dead Time	Address	Interface	Area
223.223.223.223	1	FULL/DR	00:00:39	16.16.16.1	Vl 2000	0
223.223.223.223	1	FULL/DR	00:00:33	16.16.17.1	Vl 2001	0
223.223.223.223	1	FULL/DR	00:00:34	39.39.39.1	Gi 0/39	0

Protocols > ISIS Neighbors

PROTOCOLS - ISIS NEIGHBORS

System Id	Interface	State	Type	Priority	Uptime	Circuit Id
0509.0001.0000	Te 0/10	Up/Up	L1L2	0/0	2w5d/2w5d	0010.0100.1001.01/0010.0100.1001.01
0509.0002.0000	Te 0/10	Up	L2	0	2w5d	0010.0100.1001.01
0509.0003.0000	Te 0/10	Up	L2	0	2w5d	0010.0100.1001.01
0030.0300.3003	Te 0/25	Up/Up	L1L2 (M)	64/64	4w6d/4w6d	0010.0100.1001.02/0010.0100.1001.02
050C.0001.0000	Te 0/46	Up/Up	L1L2	0/0	2w5d/2w5d	0010.0100.1001.04/0010.0100.1001.04
0020.0200.2002	V1 100	Init/Init	L1L2 (M)	64/64	1d15h/1d4h	0020.0200.2002.06/0020.0200.2002.06
0030.0300.3003	V1 100	Init/Init	L1L2 (M)	64/64	2w4d/2w4d	0020.0200.2002.06/0020.0200.2002.06
0030.0300.3003	V1 101	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.07/0010.0100.1001.07
0030.0300.3003	V1 102	Up/Up	L1L2 (M)	64/64	1d10h/1d7h	0010.0100.1001.08/0010.0100.1001.08
0030.0300.3003	V1 103	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.09/0010.0100.1001.09
0030.0300.3003	V1 104	Up/Up	L1L2 (M)	64/64	1d10h/1d10h	0010.0100.1001.0A/0010.0100.1001.0A
0030.0300.3003	V1 105	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.0B/0010.0100.1001.0B
0030.0300.3003	V1 106	Up/Up	L1L2 (M)	64/64	1d7h/1d10h	0010.0100.1001.0C/0010.0100.1001.0C
0030.0300.3003	V1 107	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.0D/0010.0100.1001.0D
0030.0300.3003	V1 108	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.0E/0010.0100.1001.0E
0030.0300.3003	V1 109	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.0F/0010.0100.1001.0F
0030.0300.3003	V1 110	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.10/0010.0100.1001.10
0030.0300.3003	V1 111	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.11/0010.0100.1001.11
0030.0300.3003	V1 112	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.12/0010.0100.1001.12
0030.0300.3003	V1 113	Up/Up	L1L2 (M)	64/64	1d10h/1d10h	0010.0100.1001.13/0010.0100.1001.13
0030.0300.3003	V1 114	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.14/0010.0100.1001.14
0030.0300.3003	V1 115	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.15/0010.0100.1001.15
0030.0300.3003	V1 116	Up/Up	L1L2 (M)	64/64	1d7h/1d10h	0010.0100.1001.16/0010.0100.1001.16
0030.0300.3003	V1 117	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.17/0010.0100.1001.17
0030.0300.3003	V1 118	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.18/0010.0100.1001.18
0030.0300.3003	V1 119	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.19/0010.0100.1001.19
0030.0300.3003	V1 120	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.1A/0010.0100.1001.1A
0030.0300.3003	V1 121	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.1B/0010.0100.1001.1B
0030.0300.3003	V1 122	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.1C/0010.0100.1001.1C
0030.0300.3003	V1 123	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.1D/0010.0100.1001.1D
0030.0300.3003	V1 124	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.1E/0010.0100.1001.1E
0030.0300.3003	V1 125	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.1F/0010.0100.1001.1F
0030.0300.3003	V1 126	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.20/0010.0100.1001.20
0030.0300.3003	V1 127	Up/Up	L1L2 (M)	64/64	1d15h/1d7h	0010.0100.1001.21/0010.0100.1001.21
0030.0300.3003	V1 128	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.22/0010.0100.1001.22
0030.0300.3003	V1 129	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.23/0010.0100.1001.23
0030.0300.3003	V1 130	Up/Up	L1L2 (M)	64/64	1d7h/1d7h	0010.0100.1001.24/0010.0100.1001.24

Diagnostics Menu

Diagnostics > Arp Table

ARP TABLE

Protocol	Address	Age(min)	Hardware Address	Interface	VLAN	CPU
Internet	5.5.5.1	87	00:01:e8:8c:44:aa	Gi 0/3	Vl 1000	CP
Internet	5.5.5.3	-	00:01:e8:9b:00:02	-	Vl 1000	CP
Internet	10.43.0.1	0	00:06:28:5d:4f:c2	Ma 0/0	-	CP
Internet	10.43.254.20	7	00:0c:29:2a:6e:cc	Ma 0/0	-	CP
Internet	16.16.16.1	71	00:01:e8:8c:44:aa	Gi 0/14	Vl 2000	CP
Internet	16.16.16.2	-	00:01:e8:9b:00:02	-	Vl 2000	CP
Internet	16.16.17.1	71	00:01:e8:8c:44:aa	Po 1	Vl 2001	CP
Internet	16.16.17.2	-	00:01:e8:9b:00:02	-	Vl 2001	CP
Internet	24.24.24.3	-	00:01:e8:9b:00:02	-	Vl 3000	CP
Internet	39.39.39.1	71	00:01:e8:8c:44:aa	Gi 0/39	-	CP
Internet	39.39.39.2	-	00:01:e8:9b:00:02	Gi 0/39	-	CP
Internet	42.42.42.3	-	00:01:e8:9b:00:02	-	Vl 4050	CP

Diagnostics > Mac Address Table

MAC ADDRESS TABLE

VlanId	Mac Address	Type	Interface	State
1000	00:01:e8:8c:44:aa	Dynamic	Gi 0/3	Active
2000	00:01:e8:8c:44:aa	Dynamic	Gi 0/14	Active
2001	00:01:e8:8c:44:aa	Dynamic	Po 2	Active

Diagnostics > Routing Table

ROUTING TABLE

Codes: C - connected, S - static, R - RIP,
 B - BGP, IN - internal BGP, EX - external BGP, LO - Locally Originated,
 O - OSPF, IA - OSPF inter area, N1 - OSPF NSSA external type 1,
 N2 - OSPF NSSA external type 2, E1 - OSPF external type 1,
 E2 - OSPF external type 2, i - IS-IS, L1 - IS-IS level-1,
 L2 - IS-IS level-2, IA - IS-IS inter area, * - candidate default,
 > - non-active route, + - summary route

Gateway of last resort is not set

	Destination	Gateway	Dist/Metric	Last Change
C	5.5.5.0/24	Direct, Vl 1000	0/0	1w0d
C	16.16.16.0/24	Direct, Vl 2000	0/0	1w0d
C	16.16.17.0/24	Direct, Vl 2001	0/0	1w0d
C	24.24.24.0/24	Direct, Vl 3000	0/0	1w0d
C	39.39.39.0/24	Direct, Gi 0/39	0/0	1w0d
C	42.42.42.0/24	Direct, Vl 4050	0/0	1w0d
C	222.222.222.222/32	Direct, Lo 0	0/0	1w0d

Diagnostics > VLANs

VLANs

Codes: * - Default VLAN, G - GVRP VLANs, P - Primary, C - Community, I - Isolated
 Q: U - Untagged, T - Tagged
 x - Dot1x untagged, X - Dot1x tagged
 G - GVRP tagged, M - Vlan-stack, H - VSN tagged

NUM	Status	Description	Q Ports
* 1	Inactive		U Gi 0/25,47
2	Inactive		
3	Inactive		
4	Inactive		
5	Inactive		
1000	Active		T Gi 0/3
2000	Active	OSPF	T Gi 0/14
2001	Active		T Po1(Gi 0/21-23) T Po2(Gi 0/10-12)
3000	Active	ISIS	T Gi 0/3
3001	Inactive		
3002	Inactive		
3500	Active	L2	T Gi 0/37
4000	Active	L2	T Gi 0/42
4009	Inactive		
4011	Inactive		
4012	Inactive		
4050	Active	L3	T Gi 0/20

Diagnostics > VLAN Members

VLAN MEMBERS

Select VLAN ID

SELECTED VLAN

Codes: * - Default VLAN, G - GVRP VLANs, P - Primary, C - Community, I - Isolated
 Q: U - Untagged, T - Tagged
 x - Dot1x untagged, X - Dot1x tagged
 G - GVRP tagged, M - Vlan-stack, H - VSN tagged

NUM	Status	Description	Q Ports
2001	Active		T Po1(Gi 0/21-23) T Po2(Gi 0/10-12)

Diagnostics > Int Bandwidth

BANDWIDTH

Enter Stack Unit
Enter Port

BANDWIDTH DATA

```
Rate info (interval 299 seconds):  
  Input 00.00 Mbits/sec,      1 packets/sec, 0.00% of line-rate  
  Output 00.00 Mbits/sec,     0 packets/sec, 0.00% of line-rate  
Time since last interface status change: 1w0d1h
```

Utilities Menu

Utilities > Ping**PING**

Enter IP Address (No Hostnames) **PING DATA**

```
Sending 5, 100-byte ICMP Echos to 10.42.0.13, timeout is 2 seconds:
!!!!
Success rate is 100.0 percent (5/5), round-trip min/avg/max = 0/0/0 (ms)
```

Utilities > Traceroute**TRACE**

Enter IP Address (No Hostnames) Enter Timeout **TRACE DATA**

```
-----
Tracing the route to 10.42.0.13, 30 hops max, 40 byte packets
-----
```

TTL	Hostname	Probe1	Probe2	Probe3
1	10.43.0.1	000.000 ms	000.000 ms	000.000 ms
2	10.42.0.13	000.000 ms	006.000 ms	000.000 ms

Settings Menu

Settings > SmartUtils Credentials

SMARTUTILS USER CREDENTIALS



This username and password must exist on FTOS and have privilege level 15 or enable password.

Configure credentials used by SmartUtils to communicate with FTOS

Enter User Name

Enter Password

Enter Enable Password

