

Open Automation Guide

Configuration and Command Line Reference

February 2013



Force10

Notes, Cautions, and Warnings



NOTE: A NOTE indicates important information that helps you make better use of your computer.



CAUTION: A CAUTION



WARNING: A WARNING indicates a potential for property damage, personal injury, or death.

Information in this publication is subject to change without notice.

© 2013 Dell Force10. All rights reserved.

Reproduction of these materials in any manner whatsoever without the written permission of Dell Inc. is strictly forbidden.

Trademarks used in this text: Dell™, the DELL logo, Dell Precision™, OptiPlex™, Latitude™, PowerEdge™, PowerVault™, PowerConnect™, OpenManage™, EqualLogic™, KACE™, FlexAddress™ and Vostro™ are trademarks of Dell Inc. Intel, Pentium, Xeon, Core™ and Celeron are registered trademarks of Intel Corporation in the U.S. and other countries. AMD is a registered trademark and AMD Opteron™, AMD Phenom™, and AMD Sempron™ are trademarks of Advanced Micro Devices, Inc. Microsoft®, Windows®, Windows Server®, MS-DOS® and Windows Vista® are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Red Hat Enterprise Linux® and Enterprise Linux® are registered trademarks of Red Hat, Inc. in the United States and/or other countries. Novell® is a registered trademark and SUSE™ is a trademark of Novell Inc. in the United States and other countries. Oracle® is a registered trademark of Oracle Corporation and/or its affiliates. Citrix®, Xen®, XenServer® and XenMotion® are either registered trademarks or trademarks of Citrix Systems, Inc. in the United States and/or other countries. VMware®, Virtual SMP®, vMotion®, vCenter®, and vSphere® are registered trademarks or trademarks of VMware, Inc. in the United States or other countries.

Other trademarks and trade names may be used in this publication to refer to either the entities claiming the marks and names or their products. Dell Inc. disclaims any proprietary interest in trademarks and trade names other than its own.

1	About this Guide	5
	Objectives	5
	Audience	5
	Supported Platforms and Required FTOS Versions	5
	Conventions	6
	Information Symbols	6
	Related Documents	7
2	Open Automation Framework	9
	Bare Metal Provisioning	10
	Smart Scripting	11
	Virtual Server Networking	11
	Programmatic Management	12
	Web Graphical User Interface and HTTP Server	12
3	Smart Scripting	13
	Overview	13
	Use Cases	14
	Downloading the Smart Scripting Package	15
	Installing Smart Scripting	16
	Displaying Installed Packages	17
	Uninstalling Smart Scripting	17
	Limits on System Usage	17
	Supported UNIX Utilities	18
	Creating PERL, Python and UNIX Scripts	20
	Creating a User Name and Password for Smart Scripting	20
	Running a Script from the FTOS CLI	21
	Logging in to a NetBSD UNIX Shell	22
	Running a Script from the UNIX Shell	22
	Using the PERL API	23
	Creating a PERL API Script	23
	Running a PERL API Script	26
	Using the Python API	27
	Creating a Python API Script	27
	Running a Python API Script	30
	Using UNIX Shell Scripting	31
	Creating a UNIX API Script	31
	Running a UNIX API Script	33
4	Smart Scripting CLI	35
	Overview	35
	Commands	35

5	Virtual Server Networking	43
	Overview	43
	Hypervisor Modes	45
	VSN Persistency	45
	VLAN configuration	45
	Management VLAN	45
	Data VLANS	45
	Hypervisor-unaware VLANs	46
	Installing VSN	46
	Enabling VSN in a Hypervisor Session	48
	Discovery	50
	Connectivity	50
	Running VSN Scripts	51
	Stopping a Hypervisor Session	52
	Disabling a Session	52
	Removing a Session	52
	Uninstalling VSN	53
	Viewing VSN information	53
6	Virtual Server Networking CLI	57
	Overview	57
	Commands	57
7	Programmatic Management	69
	Overview	69
	Using the REST API	70
	Plug-In Modules	73
8	Web GUI and HTTP Server	75
	HTTP Server	75
	Web Graphical User Interface	76
	Getting Started	76
	Menu Options	77
9	Web Graphical User Interface	79
10	Index	97

About this Guide

Objectives

This document describes the components and uses of the Open Automation Framework designed to run on the Force10 Operating System (FTOS), including:

- Smart Scripting
- Virtual Server Networking (VSN)
- Programmatic Management
- Web graphic user interface (GUI) and HTTP Server

Information another feature, Bare Metal Provisioning (BMP) that interacts with the Open Automation Framework can be found in the *FTOS Configuration Guide*.

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 on page 7](#) and the IETF Requests for Comment (RFCs).

Supported Platforms and Required FTOS Versions

The Open Automation 2.0 release is supported on the following Dell Force10 switches and minimum FTOS versions:

- S55 switches require FTOS version 8.3.5.2 or later.
- S60 switches require FTOS version of 8.3.3.7 or later.

- S4810 switches require FTOS version 8.3.10.1 or later.
- Z9000 switches require FTOS version 9.0.0.0 or later. (SmartScripts and SmartUtil support only).

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.
	FTOS Behavior	Informs you about an FTOS behavior. These behaviors are inherent to the Dell Force10 system or FTOS feature and are non-configurable.
   	Platform-specific Feature	<p>Informs you of the platform supporting the Open Automation features.</p> <p>For example, the S55, S60, and S4810 platforms support all Open Automation 2.0 features. The Z9000 platform supports the SmartScripts and SmartUtil features.</p>
	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 Force10 Networks switches discussed in this document, refer to the following documents:

- S55
 - *FTOS Command Line Reference Guide for the S55 System*
 - *FTOS Configuration Guide for the S55 System*
 - *Installing the S55 System*
- S60
 - *FTOS Command Line Reference Guide for the S60 System*
 - *FTOS Configuration Guide for the S60 System*
 - *Installing the S60 System*
- S4810
 - *FTOS Command Line Reference Guide for the S4810 System*
 - *FTOS Configuration Guide for the S4810 System*
 - *Installing the S4810 System*
- Z9000
 - *FTOS Command Line Reference Guide for the Z9000 System*
 - *FTOS Configuration Guide for the Z9000 System*
 - *Installing the Z9000 System*
- [FTOS Release Notes](#) for the platform and version you are using.

Open Automation Framework

Open Automation Framework is supported on platforms:    

Dell Force10'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 Force10 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 interoperate. 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).

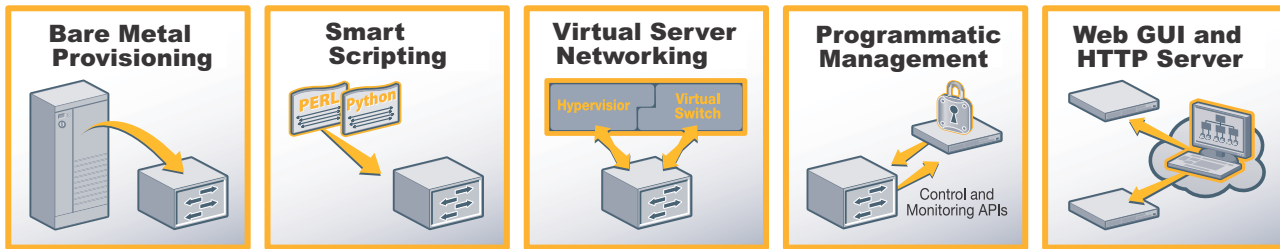
The Open Automation Framework consists of the following network management tools:

- Bare Metal Provisioning
- Smart Scripting
- Virtual Server Networking

- Programmatic Management
- Web GUI and HTTP server

You can use these components together or independently to extend and add functionality to the FTOS operating system without requiring updates to an FTOS 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 Force10 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 FTOS, the operating system for Dell Force10 switches. The switch automatically configures itself by loading and installing an embedded FTOS image with the startup configuration file.

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 Force10 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 FTOS command-line interface (CLI) can be performed with smart scripting.

The scripting environment provided by Smart Scripting (Perl, Python, and UNIX 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 Force10 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

Programmatic Management

Programmatic Management provides application programming interfaces (APIs) so that FTOS switches can 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 command.

Programmatic management greatly improves network manageability by allowing Dell Force10 switches to be managed by third-party system management tools via standard programmatic 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 Graphical User Interface and HTTP Server

The Open Automation Framework supports Web connectivity through its Web interface and HTTP server:

- The Web-based GUI allows you to retrieve and update switch attributes and characteristics.
- The HTTP Server consists of both HTTP and HTTPS daemons running on a switch and communicating with the Web GUI.

Smart Scripting

[Smart Scripting](#) is supported on platforms: **S60** **S55** **S4810** **Z**

Smart Scripting allows you to add functionality to the FTOS operating system without requiring updates to the FTOS release. Smart Scripting is available as a separate installable package that supports TCL, ZCL, Expect, PERL, Python, and UNIX scripting and various FTOS functions.

The Smart Scripting package supports smart utility APIs (SmartUtils) to provide developers with an easier way to invoke switch operations by creating and running PERL, Python, and UNIX shell scripts on the FTOS operating system. API library files describe the functions supported in TCL, ZSH, Expect, PERL, Python, and UNIX scripts.

A separate package has been extended with HTTP and HTTPS daemons to support a REST-like API based on CGI scripts and a Web-based graphical user interface. For information on the HTTP Get requests supported by the REST API, see [Chapter 7, Programmatic Management](#).

Overview

Using Smart Scripting, network administrators can create custom TCL, ZCL, Expect, PERL, Python, and UNIX shell scripts to manage and interact with Dell Force10 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).
- BMP module that implement scripts written in TCL, ZSH, or Expect. For more information, see the *FTOS Configuration Guide*, [Chapter 9, Bare Metal Provisioning 3.0 \(BMP 3.0\)](#).

With Smart Scripting, there is no need to learn proprietary scripting languages, allowing for the 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 directly in a UNIX shell. 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 to which script developers can refer when they create PERL, Python, and UNIX scripts. A representation of CLI functions to retrieve data from the FTOS operating system and change configuration parameters on Dell Force10 switches is provided in the API libraries. Script writers include API function calls made directly on the FTOS command-line interface in their PERL, Python, and UNIX 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, script writers 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 FTOS CLI or directly from a UNIX shell.

This chapter includes the following sections:

- [Use Cases](#)
- [Downloading the Smart Scripting Package](#)
- [Installing Smart Scripting](#)
- [Limits on System Usage](#)
- [Supported UNIX Utilities](#)
- [Creating a User Name and Password for Smart Scripting](#)
- [Running a Script from the FTOS CLI](#)
- [Logging in to a NetBSD UNIX Shell](#)
- [Running a Script from the UNIX Shell](#)
- [Using the PERL API](#)
- [Using the Python API](#)
- [Using UNIX Shell Scripting](#)

Use Cases

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 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

The SmartScripts package can be downloaded from the Dell Force10 website as a file named **SmartScripts2.0.x.tar.gz** for platforms such as S4810, S55, and S60; for Z9000 the file name is **SmartScripts-Z.2.0.x.tar.gz**. The Smart Scripting package is downloaded with the following files and functionality:

- PERL interpreter and associated files
- Python interpreter and associated files
- Expanded set of UNIX utilities
- REST-like API based on CGI scripts (see [Using the REST API](#))
- Web-based graphical user interface (see [Web Graphical User Interface](#))
- HTTP and HTTPS daemons (see [HTTP Server](#))

Installing Smart Scripting

You install the Smart Scripting file in the same way as you install an FTOS 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.

 **CAUTION** You can modify (e.g. 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, you must download it from the Dell Force10 web portal:

1. On a PC or other network device, go to the Dell Force10 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, click the **SMARTSCRIPTS2.0.x.tar.gz** file for S55, S60 and S4810 switches. Click the **SMARTSCRIPTS-Z.2.0.x.tar.gz** file for Z9000 switches.
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 **SMARTSCRIPTS2.0.x.tar.gz** file for S55, S60 and S4810 switches or the **SMARTSCRIPTS-Z.2.0.x.tar.gz** file for Z9000 switches.
7. When the download is complete, enter the **package install** command from the FTOS 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.
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. 		

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

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

Displaying Installed Packages

To view the Open Automation packages currently installed on a switch, including version numbers and content, enter 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, you must first stop all scripts that are currently running using the **no script script-name** command. You must also manually stop the http server daemon.

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 3-1. Limits on System Attributes

System Attribute	Value	Description
cputime	unlimited	Maximum amount of time used by a process.
filesize	unlimited	Largest file size that can be created.

Table 3-1. Limits on System Attributes

System Attribute	Value	Description
datasize	131,072 KB	Maximum size of the data segment for a process; this value defines how far a program may extend its break with the sbrk(2) system call.
stacksize	2,048 KB	Maximum size 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 of a core file that may be created
memory use	233,244 KB	Maximum size 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 will prefer 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 3-2. 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.
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.

Table 3-2. Supported UNIX Utilities (continued)

fc	List the history of commands on the computer.
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.

Creating PERL, Python and UNIX Scripts

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 FTOS CLI. These APIs provide a shortcut when writing scripts. Refer to the following sections for more information:

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

For instructions on how to run a PERL, Python, or UNIX script from the FTOS CLI, see [Running a Script from the FTOS CLI](#).

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

Creating a User Name and Password for Smart Scripting

Before you run a script from the FTOS CLI, you may want to 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 with the **script** command from the FTOS CLI.

The user name is an optional entry in the **script** command (see [Running a Script from the FTOS CLI](#)). 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
<code>username name password password</code>	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.

Running a Script from the FTOS CLI

You can run any PERL, Python, and UNIX script that is stored on the switch from the FTOS CLI.

When you run a script from the FTOS CLI, you can specify an optional user name to apply the associated read-write privileges when the script is run (see [Creating a User Name and Password for Smart Scripting](#)). If you do not specify a user name, the script is run with the privileges of the current user.

To run a PERL, Python, or UNIX script from the FTOS CLI, enter the **script** command. You must enter the script name and directory path to start the script. The script can invoke any of the supported UNIX utilities listed in [Table 3-2](#). You can 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, see Figure 3-1 . 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 will be applied when the script is run. 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 is run. Enter a blank space between parameter values; for example: script username admin /f10/flash/createVlans.py 1 2

To stop a script that is running, enter the **no** version of the **script** command; for example: **no script admin.pl**.

To display the scripts that are currently running, including the scripts you have stopped, enter the **show running-config | grep** command.

Figure 3-1. Starting and Stopping PERL and Python Scripts: Examples

```
f)# script /usr/pkg/scripts/sample_scripts/cmd-server.pl
f)# no script /usr/pkg/scripts/sample_scripts/cmd-server.pl

f)# script username admin /usr/pkg/scripts/sample_scripts/DisplayAlarms.py
f)# no script username admin /usr/pkg/scripts/sample_scripts/DisplayAlarms.py
```

Tip: For information on how to run a script directly from a UNIX shell without using the FTOS CLI, see [Running a Script from the UNIX Shell](#).

Logging in to a NetBSD UNIX Shell

To log in to the NetBSD UNIX shell on a switch to directly enter any of the UNIX commands described in [Table 3-2](#) or to run a script, enter 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 (see Running a Script from the UNIX Shell).

Running a Script from the UNIX Shell

You can run any PERL, Python, and UNIX script stored on a switch from either the FTOS CLI (see [Running a Script from the FTOS CLI on page 21](#)) or directly from a NetBSD shell on the switch.

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

[Figure 3-2](#) shows examples of how to execute a PERL, Python, and UNIX shell script directly from a NetBSD shell on the FTOS operating system.

Figure 3-2. Execution of a PERL, Python, and Shell Script from a UNIX Shell: Example

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

4.4 BSD UNIX () (ttyp0)

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

Use the information in this section to create a PERL script using the PERL API and run the script on a Dell Force10 switch. For information on how to create and run a Python script using the Python API, see [Using the Python API](#).

Creating a PERL API Script

The Programmatic Management package provides a PERL API library containing the supported functions (described in [Table 3-2](#)), which can be used in a PERL script to invoke FTOS operations on a switch. The PERL API library is stored in the F10SmartUtils.pl file at /usr/pkg/scripts/smartutils.

You code FTOS API functions in a PERL script as shown in the following example:

Figure 3-3. 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;
}
```

[Table 3-2](#) describes the supported functions and required arguments that you can use in PERL scripts run on a Dell Force10 switch to connect through a telnet session and gather information or configure parameters through the CLI.

Table 3-3. Supported FTOS 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.
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.

Table 3-3. Supported FTOS API Functions in PERL Scripts (continued)

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 PERL API Script

When you run a PERL script that invokes the API functions in [Table 3-2](#), logon credentials are read from the smartutils.cfg file, and a telnet session is opened on the switch in which function calls are executed in the FTOS 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 Management 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 is downloaded with the Programmatic Management package and is stored at /usr/pkg/scripts/smartutils.



Note: The user name and passwords contained in the smartutils.cfg file are used 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 in the smartutils.cfg file that are used to run PERL API scripts, do one of the following:

- From the Web user interface, select **Settings > SmartUtils Credentials** (see [Menu Options](#)).
- 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 FTOS CLI, use the **script** command as described in [Running a Script from the FTOS CLI](#).
- From a UNIX shell, follow the procedure described in [Running a Script from the UNIX Shell](#).

Using the Python API

Use the information in this section to create a Python script using the Python API and run the script on a Dell Force10 switch. For information on how to create and run a PERL script using the PERL API, see [Supported UNIX Utilities](#).

Creating a Python API Script

Use the information in this section to create a Python script to be run on a Dell Force10 switch. For information on how to run a Python script from the FTOS CLI, see [Running a Script from the FTOS CLI](#).

F10SmartUtils.py is the Python API library containing the supported functions (described in [Table 3-4](#)), which can be used in a Python script to invoke FTOS operations on a switch. This file is downloaded with the Programmatic Management package and is stored at **/usr/pkg/scripts/smartutils**.

You code FTOS API functions in a Python script as shown in the following example:

Figure 3-4. 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>"
```

[Table 3-4](#) describes the supported functions and required arguments that you can use in Python scripts run on a Dell Force10 switch to connect through a telnet session and gather information or configure parameters through the CLI.

Table 3-4. Supported FTOS API Functions in Python Scripts

Python 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.

Table 3-4. Supported FTOS API Functions in Python Scripts (continued)

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.
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.

Table 3-4. Supported FTOS API Functions in Python Scripts (continued)

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 [Table 3-4](#), logon credentials are read from the smartutils.cfg file, and a telnet session is opened on the switch in which function calls are executed in the FTOS 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 Management 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 is downloaded with the Programmatic Management package and is stored at /usr/pkg/scripts/smartutils.



Note: The user name and passwords contained in the smartutils.cfg file are used 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 in the smartutils.cfg file that are used to run Python API scripts, do one of the following:

- From the Web user interface, select **Settings > SmartUtils Credentials** (see [Menu Options](#)).
- 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 FTOS CLI, use the **script** command as described in [Running a Script from the FTOS CLI](#).
- From a UNIX shell, follow the procedure described in [Running a Script from the UNIX Shell](#).

Using UNIX Shell Scripting

Use the information in this section to create a UNIX script using the UNIX API and run the script on a Dell Force10 switch. For information on how to create and run a PERL or Python script using the PERL or Python API, see [Supported UNIX Utilities](#) and [Using the Python API](#).

Creating a UNIX API Script

Use the information in this section to create a UNIX shell script to be run on a Dell Force10 switch.

The F10SmartScriptUtils.py file is the main API library file that contains the functions that you can include in a UNIX shell script. The F10SmartScriptUtils.py file is downloaded with the Programmatic Management package and is stored at /usr/pkg/scripts/smartutils. [Table 3-5](#) describes the FTOS operations that you can invoke from a UNIX shell script, including the supported functions and required arguments.

[Figure 3-5](#) 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 3-5. 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 3-5. 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.

Table 3-5. Supported API Functions in UNIX Shell Scripts (continued)

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.
showipintbrieflag	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.
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.

Table 3-5. Supported API Functions in UNIX Shell Scripts (continued)

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.
tracert	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 [Table 3-5](#), logon credentials are read from the **smartutils.cfg** file, and a telnet session is opened on the switch in which function calls are executed in the FTOS 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 Management 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 is downloaded with the Programmatic Management package and is stored at **/usr/pkg/scripts/smartutils**.



Note: The user name and passwords contained in the smartutils.cfg file are used 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 in the **smartutils.cfg** file that are used to run UNIX API scripts, do one of the following:

- From the Web user interface, select **Settings > SmartUtils Credentials** (see [Menu Options](#)).
- 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 FTOS CLI, use the **script** command as described in [Running a Script from the FTOS CLI](#).
- From a UNIX shell, follow the procedure described in [Running a Script from the UNIX Shell](#).

Smart Scripting CLI

Overview

Smart Scripting CLI is supported on platforms: **S55** **S60** **S4810** **Z**

Commands

- [package install](#)
- [package uninstall](#)
- [script](#)
- [show packages](#)
- [start shell](#)
- [username](#)

package install

S55 **S60**
S4810 **Z**

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://filename • From an FTP server: ftp://userid:password@host-ipaddress/filepath • From a TFTP server: tftp://host-ipaddress/filepath
-----------------	--

Defaults

None

Command Modes

EXEC Privilege

Command History

Version 9.0.0.0	Introduced on the Z9000.
Version 8.3.8.0	Introduced on the S4810.

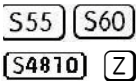
Version 8.3.5.1	Introduced on the S55.
Version 8.3.3.4	Introduced on the S60.

Usage Information

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

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

package uninstall



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 9.0.0.0	Introduced on the Z9000.
Version 8.3.8.0	Introduced on the S4810.
Version 8.3.5.1	Introduced on the S55.
Version 8.3.3.4	Introduced on the S60.

Usage Information

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



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

memory.

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

Related commands

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

script

S55 S60
S4810 Z

Run a Perl, Python, or UNIX shell script from the FTOS CLI.

Syntax

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

Parameters

username <i>name</i>	(Optional) Enter the user name whose read-write privileges will be applied when the script is run. A username used to run scripts cannot contain special characters.
script-name	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
script-parameter	(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 9.0.0.0	Introduced on the Z9000.
Version 8.3.8.0	Introduced on the S4810.
Version 8.3.5.1	Introduced on the S55.
Version 8.3.3.4	Introduced on the S60.

Usage Information

You can enter the [script](#) command multiple times to run more than one script at the same time; for example:

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

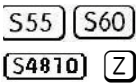
When you run a script from the FTOS CLI with the [script](#) command, you can specify an optional user name to apply the read-write privileges assigned to the user name when the script is run (see [Running a Script from the FTOS CLI](#)). You configure the username and password with the [username](#) command. If you do not specify a user name with the [script](#) command, the script is run with the privileges of the current user.

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

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

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

show packages



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

Syntax

show packages

Defaults

None

Command Modes

EXEC

EXEC Privilege

Command History

Version 9.0.0.0	Introduced on the Z9000.
Version 8.3.8.0	Introduced on the S4810.
Version 8.3.5.1	Introduced on the S55.
Version 8.3.3.4	Introduced on the S60.

Example

```
FTOS# 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

```

FTOS# show packages
* Package Name:SMARTSCRIPTS-Z 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

```

start shell

S55 S60
S4810 Z

Start a NetBSD UNIX shell.

Syntax	start shell
Defaults	None
Command Modes	EXEC Privilege

Command History	Version 9.0.0.0	Introduced on the Z9000.
	Version 8.3.8.0	Introduced on the S4810.
	Version 8.3.5.1	Introduced on the S55.
	Version 8.3.3.4	Introduced on the S60.

Usage Information	You must start an NetBSD shell on a switch before you can enter UNIX commands (Table 3-2) or run a script directly from the shell to invoke FTOS operations (see 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.
------------------	-------------------------------	---

username



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`

Enter **no username** to remove the user name and password.

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 9.0.0.0	Introduced on the Z9000.
	Version 8.3.8.0	Introduced on the S4810.
	Version 8.3.5.1	Introduced on the S55.
	Version 8.3.3.4	Introduced on the S60.

Usage Information When you run a script from the FTOS CLI with the [script](#) 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 [Running a Script from the FTOS CLI](#)).

Virtual Server Networking

[Virtual Server Networking](#) is supported on platforms: **S60** **S55** **S4810**

As a part of the Open Automation package, Virtual Switch Networking (VSN) provides real-time communication between the Dell Force10 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 Force10 switch/routers in a data center network to retrieve configuration information from hypervisors. VMware vSphere and Citrix Xen hypervisors are supported.

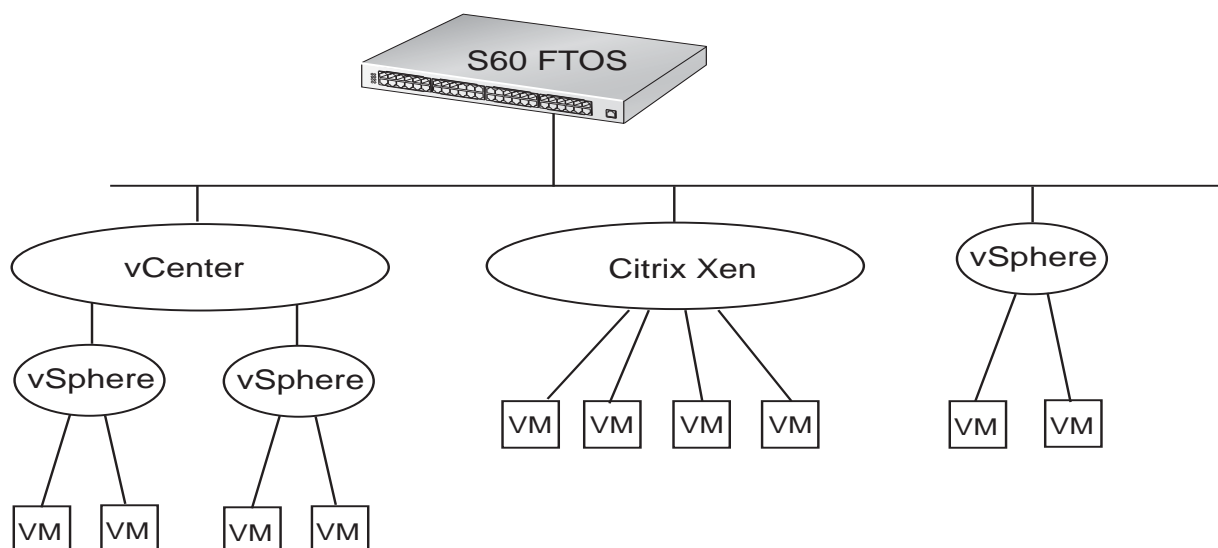
Both VMware and Citrix Xen provide SDKs and APIs for accessing their configuration objects. VSN requires Layer 3 connectivity to access a hypervisor.

Figure 5-1 shows an example of the network architecture in which a Dell FTOS 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.

FTOS 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 5-1. Virtual Server Networking example



VSN subscribes to hypervisor for any change to be notified to switch. Depending on the hypervisor mode configured, FTOS 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 FTOS configuration.
- **Config:** VSN retrieves configuration information from a hypervisor and automatically makes the required configuration changes in FTOS on the switch.

VSN Persistency

VSN installation and configuration is persistent in the FTOS 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 FTOS 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 FTOS 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.

Enter the **show vlan** command to display the VSN hypervisor-learned VLANs on the switch. As shown in Figure 5-2, 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 5-2. Display VSN Hypervisor-learned VLANs: show vlan

```
FTOS(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 FTOS image and the downloaded Smart Scripting package. When you install the VSN package, VSN is loaded into FTOS.



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 FTOS 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 *FTOS 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 Force10 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, click the **VSNAGENT2.0.x.tar.gz** file.
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 VSNAGENT2.0.x.tar.gz file.
7. When the download is complete, enter the **package install** command from the FTOS 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.	EXEC Privilege	Install the VSN package in the running configuration of the switch from local flash memory or a network server.

8. 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), enter 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 Force10 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	access url username username password password Where <i>url</i> is one of the following values: For a VMware hypervisor: https://[ip-address]/sdk/vimService username [name] password [password] For an Xen hypervisor: http://ip-address username [name] password [password] username name: Username to be used for authentication on the server. password password: Password to be used for authentication shown in clear text.	HYPERVISOR
5	Set the mode for retrieving virtual server configurations and updating FTOS settings on the switch.	mode {check config} 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. config: Retrieve configuration information and automatically update the configuration parameters in FTOS on the switch. Default: config .	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 10 minutes after connectivity is established between the switch and 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:

- You 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.
- You 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 2.0.x 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 FTOS 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> .



Note: The **script** command is not supported on S55 switches to run VSN scripts.

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 HYPERVERSOR 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	HYPERVERSOR	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 5-3. Display a Hypervisor Session: show configuration

```
FTOS(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, enter the **show hypervisors supported** command.

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

Figure 5-4. Display Supported Hypervisors: show hypervisor supported

```
FTOS#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, enter 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 5-6).

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

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

Figure 5-6. Display a Specified Hypervisor Sessions: show virtualswitch

```
FTOS#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 vmmap** command.

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

Figure 5-7. Display Virtual Machines Accessed on an Interface: show vmmmap

```
FTOS#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 platforms: **S60** **S55** **S4810**



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 vmmap](#)
- [type](#)
- [vsn enable](#)

access

S55 S60 S4810

Configure the connection to access a hypervisor.

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 <i>name</i>	Enter the user name to be used for authentication.
password <i>password</i>	Enter the password to be used for authentication in clear text.

Defaults

None

Command Modes

HYPERVISOR

Command History

Version 8.3.8.0	Introduced on the S4810.
Version 8.3.5.1	Introduced on the S55.
Version 8.3.3.4	Introduced on the S60.

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

S55 S60 S4810

Stop a hypervisor session.

Syntax

[no] disable

Defaults

disable

Command Modes

HYPERVISOR

Command History

Version 8.3.8.0	Introduced on the S4810.
Version 8.3.5.1	Introduced on the S55.
Version 8.3.3.4	Introduced on the S60.

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

S55 S60 S4810

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 8.3.8.0	Introduced on the S4810.
Version 8.3.5.1	Introduced on the S55.
Version 8.3.3.4	Introduced on the S60.

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

S55 S60 S4810

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 FTOS 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 8.3.8.0	Introduced on the S4810.
Version 8.3.5.1	Introduced on the S55.
Version 8.3.3.4	Introduced on the S60.

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

S55 S60 S4810

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.
-----------------	---

Defaults

None

Command Modes

EXEC Privilege

Command History

Version 8.3.8.0	Introduced on the S4810.
Version 8.3.5.1	Introduced on the S55.
Version 8.3.3.4	Introduced on the S60.

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

S55 S60 S4810

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

Syntax	package uninstall <i>name</i>	
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 8.3.8.0	Introduced on the S4810.
	Version 8.3.5.1	Introduced on the S55.
	Version 8.3.3.4	Introduced on the S60.
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 <i>script-name</i> command.
Related commands	show packages	Display all the packages installed in the system.

script

) S60 S4810

Run an installed VSN script (Perl or Python) on active hypervisor links to retrieve virtual server configurations and update FTOS 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 8.3.8.0	Introduced on the S4810.
	Version 8.3.3.4	Introduced on the S60.

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.



Note: The **script** command used to run VSN scripts is not supported on S55 switches.

show hypervisor supported

S55 **S60** **S4810** Display the types of hypervisors currently supported by VSN.

Syntax	show hypervisor supported	
Defaults	None	
Command Modes	EXEC Privilege	
Command History	Version 8.3.8.0	Introduced on the S4810.
	Version 8.3.5.1	Introduced on the S55.
	Version 8.3.3.4	Introduced on the S60.
Usage Information	Use this information when defining types of hypervisor connections with the hypervisor command.	
Related Commands	hypervisor	Define a hypervisor instance.
Example	FTOS#show hypervisor supported vmware xen-citrix	

show packages

S55 **S60** **S4810** Display all Open Automation packages installed on a switch.

Syntax	show packages	
Defaults	None	
Command Modes	EXEC Privilege	
Command History	Version 8.3.8.0	Introduced on the S4810.
	Version 8.3.5.1	Introduced on the S55.
	Version 8.3.3.4	Introduced on the S60.

Example

```

FTOS#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

S55 S60 S4810

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 8.3.8.0	Introduced on the S4810.
	Version 8.3.5.1	Introduced on the S55.
	Version 8.3.3.4	Introduced on the S60.

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.

```
FTOS#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.

```
FTOS#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 vmmap

S55 S60 S4810

Display the virtual machines accessed on a switch interface.

Syntax `show vmmap interface`

Defaults None

Parameters	<i>interface</i>	Display information on the virtual machines accessed 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 the keyword 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.
-------------------	------------------	--

Command Modes EXEC Privilege


Command History	Version 8.3.8.0	Introduced on the S4810.
	Version 8.3.5.1	Introduced on the S55.
	Version 8.3.3.4	Introduced on the S60.

Usage Information The [show vmmap](#) 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

```
FTOS#show vmmap 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 vmmap** output, VLAN 1 is displayed as VLAN ID 1; VLAN 4095 is displayed without a VLAN ID as "- "

type

S55 S60 S4810

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

Version 8.3.5.1	Introduced on the S55.
-----------------	------------------------

Version 8.3.3.4	Introduced on the S60.
-----------------	------------------------

vsn enable

S55 S60 S4810

Enable VSN on an interface.

Syntax

[no] vsn enable

Defaults

VSN is disabled by default on switch interfaces.

Command Modes

INTERFACE

Command History

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

Version 8.3.5.1	Introduced on the S55.
-----------------	------------------------

Version 8.3.3.4	Introduced on the S60.
-----------------	------------------------

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.

Programmatic Management

[Programmatic Management](#) is supported on platforms: **S60** **S55** **S4810** and is downloaded with the SmartScripts package (see [Downloading the Smart Scripting Package](#)).



Note: This feature is not currently supported on the Z9000 platform.

Overview

In the Open Automation framework, Programmatic Management allows you to remotely manage Dell Force10 switches by invoking “out-of-the-box” scripts using the Representational State Transfer (REST) application programming interface (API).

The REST API takes advantage of Perl and Python scripts to add switch functionality outside of FTOS. You can run scripts from a remote device to access a switch and perform FTOS operations through a REST-based HTTP call.

Script writers can use the REST API to access the FTOS CLI on switches without having to code individual CLI commands and telnet connections for each command. This API allows Open Automation to remain independent of changes in the FTOS CLI.

In addition to using the REST API, you can use third-party management tools and other industry-standard management protocols, such as SNMP (Get and Set) and XML, to manage Dell Force10 switches.

- For information on the third-party management tools supported to manage Dell Force10 switches, see [Plug-In Modules](#).
- For information on the SNMP and XML functions supported on Dell Force10 switches, refer to the *FTOS Configuration Guides* for the S55, S60, and S4810 systems.

Using the REST API

The script-based REST API allows you to remotely access a switch that supports Open Automation from a network management device through programmatic HTTP requests to directly perform FTOS functions.

The REST API operates by invoking the CGI scripts within the HTTP server on the switch. The HTTP server passes an HTTP request to the backend CGI scripts. For more information on the HTTP server, see [HTTP Server](#).

The CGI scripts are shared with the Web GUI to retrieve data through the FTOS CLI. The CGI script functions are stored on the switch under the main Web GUI directory at /htdocs/cgi-bin.



Note: In Open Automation 2.0, the REST API supports only CGI scripts that perform HTTP read (get) requests. HTTP write (post) requests that make configuration changes will be supported in future releases.

All REST-based API calls return plain text output.

Prerequisites: You must know the IP address of the switch to which you want to connect and there must be network connectivity from your remote device to the switch.

[Table 7-1](#) describes the CGI scripts supported in an HTTP get request to access the REST API and return data from a remote Dell Force10 switch.

The following example shows how to embed a REST-based HTTP get request in a Perl script run from a remote device.

Figure 7-1. Perl Sample with HTTP Get Request that Invokes the REST API

```
#!/usr/bin/perl

require LWP::UserAgent;

# Create new LWP object and set the timeout large because "F10Ping" will take a
# while to return results (5 pings each with 5 second timeout, plus GUI delay).
my $ua = LWP::UserAgent->new;
$ua->timeout(300);

#Send HTTP GET request message and store response data
my $response = $ua->get('http://10.43.3.55/cgi-bin/F10Ping?IpAddress=10.43.0.1');

#Display response texts on success or display status
if ($response->is_success) {
    print $response->decoded_content;
}
else {
    die $response->status_line;
}
```

The following example shows how to embed an HTTP get request in Python script.

Figure 7-2. Python Sample with HTTP Get Request that Invokes the REST API

```
#!/usr/bin/python

import httplib

conn = httplib.HTTPConnection("10.42.51.5")

# Send HTTP GET request
conn.request("GET", "/cgi-bin/F10Ping?IpAddress=10.42.0.13")

# Get response data
response = conn.getresponse()

# Display response texts on success or display status
if(response.status == 200):
    # Handle response data
    data = response.read()
    print data
else :
    # Handle error
    print "Operation failed",response.status,response.reason

conn.close()
```

Table 7-1. Supported Get Requests Invoked through the REST API

HTTP Get Request	FTOS CLI Operation
F10Ping?IpAddress={ <i>ip-address</i> }	Pings a remote host from the switch using HTTP and returns the output.
F10ShowArpTbl	Returns a formatted table of known MAC address-to-IP address bindings.
F10ShowBGPNeighbors	Returns information on currently running BGP instances and discovered (configured or connected) BGP neighbors.
F10ShowBGPSummary	Returns summary information on BGP sessions.
F10ShowBootVar	Returns the FTOS images that are loaded on the switch and the order in which they are used to reload the switch.
F10ShowDate	Returns the current system date and time.
F10ShowIntBrief	Returns brief status (up/down/ admin) of all interfaces.
F10ShowIntBriefLag	Returns brief interface status (up/down/ admin) of all port-channel interfaces.
F10ShowIntBriefMan	Returns brief interface status (up/down/ admin) of all management interfaces.
F10ShowIntBriefPhy	Returns brief interface status (up/down/ admin) of all physical interfaces.
F10ShowIntBriefVlan	Returns brief status (up/down/ admin) of all VLAN interfaces.
F10ShowIPRoute	Returns information from the switch's routing table.
F10ShowISISNeighbors	Returns information on currently running ISIS instances and discovered (configured or connected) ISIS neighbors.
F10ShowLog	Returns the contents of the event log in the switch memory buffer.
F10ShowMacAddrTbl	Returns a table of learned MAC addresses from the switch's forwarding table.
F10ShowMem	Returns information on switch memory consumption.
F10ShowOSPFNeighbors	Returns information on configured OSPF instances and discovered OSPF neighbors.
F10ShowPhyIntBand?StackSlot={ <i>slot-number</i> }&Port={ <i>port-number</i> }	Returns the amount of bandwidth used for a specified port.
F10ShowProcCpu	Returns information on CPU utilization and the processes running on the switch.
F10ShowRun	Returns the contents of the running configuration file.
F10ShowVersion	Returns information on the version of the currently running switch software.
F10ShowVlan	Returns the table of known VLANs and their member interfaces.
F10ShowVlanId?VlanId={ <i>vlan-id</i> }	Returns a list of the member interfaces that belong to a specified VLAN.
F10ShowVrrp	Returns information on the currently configured VRRP instances, sessions, and their status.
F10ShowVrrpBrief	Returns information on VRRP sessions and their status.
F10Traceroute?IpAddress={ <i>ip-address</i> }&Timeout={ <i>timeout</i> }	Performs a traceroute operation to a remote host from the switch and returns output to the client device from which the HTTP request was sent for a specified timeout period.

Plug-In Modules

[Programmatic Management](#) are third-party management tools and applications that run on host devices in a data center network.

Plug-in modules running on remote hosts work together to provide a framework that may invoke SNMP get and set requests, XML queries, and Telnet CLI commands on Dell Force10 switches. For example, the Oracle OEM plug-in can retrieve status information on network interfaces, and CPU and memory usage via SNMP walks, resulting in timely detection of possible switch problems.

[Table 7-2](#) describes the plug-in modules that are supported to access Dell Force10 switches.

Table 7-2. Supported Plug-In Modules to Access Dell Force10 Switches

Management Tool and Required Version	Supported Devices
CA Spectrum Infrastructure Manager	S55, S60, and S4810
EMC Smarts Ionix	S55, S60, and S4810
Dell AIM	S55, S60, and S4810
HP Network Automation (NA)	S55, S60, and S4810
IBM Systems Director	S55, S60, and S4810
Oracle Enterprise Manager (OEM) version 12c	S55, S60, and S4810

For more information on a plug-in module, refer to the third-party web site for the management tool.

Web GUI and HTTP Server

Web GUI and HTTP Server are supported on platforms **S55** **S60** **S4810**

and are downloaded with the SmartScripts package (see [Downloading the Smart Scripting Package](#)).

This chapter describes the Web-based components in the Open Automation package:

- [HTTP Server](#)
- [Web Graphical User Interface](#)

HTTP Server

In the Open Automation package, the HTTP web server runs on a switch and handles HTTP requests from the Web-based graphical user interface (GUI). You can start the HTTP 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, enter 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 HTTP requests and write the configuration to the running configuration, enter 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 HTTP requests.

Enter the **no http-server {http | secure-http}** command to stop the web server and remove the configuration from the running-configuration file.

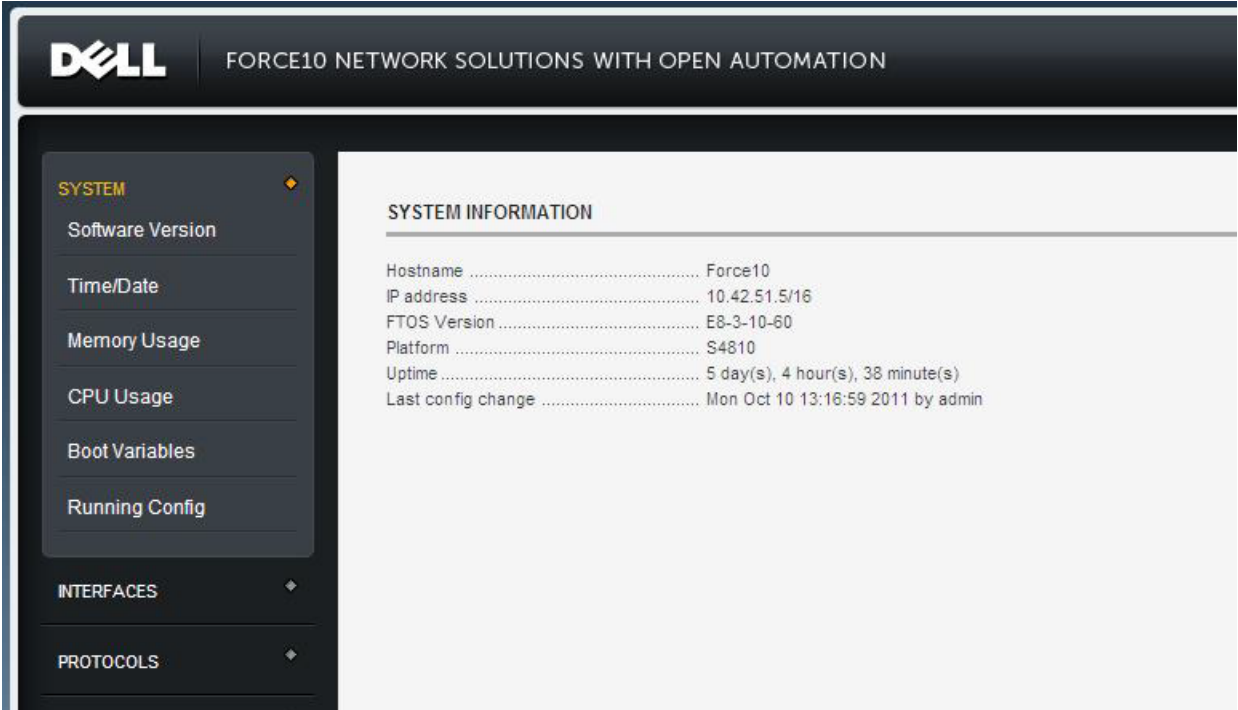
Web Graphical User Interface

In the Open Automation package, the Web graphical user interface (GUI) provides a user-friendly way to retrieve configuration information from a switch by choosing a menu option.

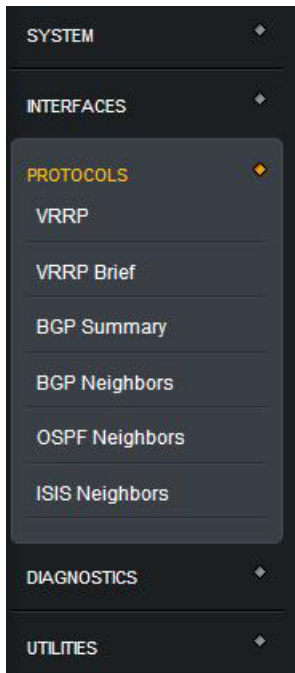
Getting Started

To open the Web interface and get started on switch operations, follow these steps:

Step	Task
	<p>Prerequisites: Only the following web browsers are supported:</p> <ul style="list-style-type: none">Internet Explorer 7.0 or laterFirefox 3.6 or later
1.	Open a web browser and enter the http://ip-address command to access the Open Automation web-based interface for a switch. The main screen of the Web GUI is displayed as shown below:



-
- 2 To retrieve or change configuration parameters on the switch, click a menu name and then click a menu option. You may be prompted to enter more information.
-



Refer to [Table 8-1](#) for a list of the tasks you can perform by choosing each menu option.

Refer to the [Web Graphical User Interface](#) for examples of the output of each menu option.

Menu Options

[Table 8-1](#) describes the switch operation performed by each menu option.

Table 8-1. Web User Interface: Supported Operations

Menu Option	Task
System > Software Version	Displays information on the FTOS version currently running on the switch.
System > Time/Date	Displays the system date and time.
System > Memory Usage	Displays information on memory usage.
System > CPU Usage	Displays information on CPU usage for the processes running on the switch.
System > Boot Variables	Displays the FTOS images loaded on a switch and the order in which they are used when the system boots up.
System > Running Config	Displays the currently running configuration on a switch.
Interfaces > All	Displays brief status information (up, down, administratively up or down) on all interfaces.
Interfaces > Physical	Displays the status and IP address of physical port interfaces.
Interfaces > LAGs	Displays the status and IP address of port-channel interfaces.
Interfaces > VLANs	Displays the status and IP address of VLAN interfaces.

Table 8-1. Web User Interface: Supported Operations (continued)

Interfaces > Management	Displays the status and IP address of management interfaces.
Protocols > VRRP	Displays the currently configured VRRP instances on a switch, including status and session information.
Protocols > VRRP Brief	Displays summary information on BGP sessions and status.
Protocols > BGP Summary	Displays summary information on BGP sessions.
Protocols > BGP Neighbors	Displays detailed information on current BGP sessions, including connected neighbors.
Protocols > OSPF Neighbors	Displays detailed information on current OSPF sessions, including connected neighbors.
Protocols > ISIS Neighbors	Displays detailed information on current ISIS sessions, including connected neighbors.
Diagnostics > ARP Table	Displays the learned MAC address-to-IP address bindings from the ARP table.
Diagnostics > MAC Address Table	Displays the learned MAC addresses from the forwarding table.
Diagnostics > Routing Table	Displays information on learned IP routes from the routing table.
Diagnostics > System Log	Displays the current events from the switch log buffer.
Diagnostics > VLANs	Displays the currently configured VLANs and their port members.
Diagnostics > VLAN Members	Displays the current membership of a specified VLAN ID.
Utilities > Ping	Ping a remote host at the specified IP address via HTTP and display returned output.
Utilities > Traceroute	Trace the route to a remote host at the specified IP address using the specified timeout value (in seconds) and display returned output.
Settings > SmartUtils Credentials	<p>Reconfigure the user name, password, and enable password used to log on to FTOS on a switch and run a script.</p> <p>Important: Use this option to ensure that the user credentials applied by Smart Scripting to run scripts on a switch are the same values as those configured on the FTOS CLI with the username command.</p>

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-oa
!
enable password 7 b125455cf679b208e79b910e85789edf
!
username test password 7 7b56aef7d3alcce8
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 > System Log

SYSTEM LOG

[illegible]

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 2001

Submit

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)10.42.0.13

Submit

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)10.42.0.13

Enter Timeout130

Submit

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

Index

A

APIs

- Perl 25
- Python 28
- REST 72
- UNIX 31

B

Bare Metal Provisioning

- description 10
- switch autoconfiguration 10

D

- displaying installed OA packages 17
- Document conventions 6

E

- ESX hypervisor 44

H

HTTP server

- description 12, 75
- starting in secure mode 75
- starting without SSL 75

hypervisors

- check and config modes 45
- connecting to 51
- discovering VMACs and VLANs 50
- displaying a session 53
- displaying virtual machines 54
- displaying VSN VLANs 46
- enabling a session 48
- removing a session 52
- running a script 51
- stopping a session 52
- supported with VSN 11

M

- menu options, for Web interface 77, 79
- minimum software versions required 5

O

Open Automation

- components 10
- description 9
- display installed packages 17

P

PERL

- application programming interface 23

Perl 13

- creating a script 23
- running a script 26
- supported API functions 25

Plug-in modules

- as third-party management tools 73
- description 73
- supported modules 73

Programmatic Management

- description 12, 69
- protocols supported 12
- REST API 70
- third-party tools supported 12, 69

Python 13

- application programming interface 27
- creating a script 27
- running a script 30
- supported API functions 28

R

REST API

- description 70
- supported CGI scripts 72

S

scripts 52

- adding functionality with Smart Scripting 13
- creating a Perl API script 23
- creating a Python API script 27
- creating a UNIX API script 31
- creating a user name 20
- logging in to a UNIX shell 22
- running a perl API script 26
- running a Python API script 30
- running a script 37
- running a UNIX API script 33
- running from a UNIX shell 22
- running from FTOS CLI 21
- samples installed with Smart Scripting 20
- stopping a running script 21

shell

- logging in to UNIX 22
- starting 41

Smart Scripting

- description 11, 13
- installation 16
- package contents 15
- PERL API 23
- Perl scripts 13

- Python API 27
- Python scripts 13
- REST API 69
- restrictions on CPU and memory usage 17
- running scripts from a UNIX shell 22
- scripting languages supported 11
- supported UNIX utilities 18
- UNIX scripts 13, 31
- use cases 14
- username for running scripts 20
- Smart Scripting commands
 - package install 35
 - package uninstall 36
 - script 37
 - show packages 38
 - start shell 41
 - username 42
- SmartUtils 13
- stopping a hypervisor session 52

T

- third-party management tools supported 73

U

UNIX

- creating a shell script 31
- logging in to shell 22
- running a UNIX script 33
- running scripts from a shell 22
- supported API functions 31
- UNIX scripts 13
- UNIX utilities 18
- user name
 - for running scripts 20

V

- vCenter hypervisor 44
- vCenter server 44
- virtual machines 44
 - displaying in a hypervisor session 54
- Virtual Server Networking
 - connecting to a hypervisor 51
 - description 11, 43
 - discovering hypervisor configuration 50
 - displaying a hypervisor session 53
 - displaying VSN VLANs 46
 - enabling on an interface 48
 - hypervisors supported 11
 - installation 46
 - removing a hypervisor session 52

- running a script in a hypervisor session 51
- stopping a hypervisor session 52
- stopping a script 52
- supported hypervisors 43
- VLAN configuration 45
- Virtual Server Networking commands
 - access 58
 - disable 58
 - hypervisor 59
 - mode 59
 - package install 60
 - package uninstall 61
 - script 62
 - show hypervisor supported 63
 - show packages 63
 - show virtualswitch 65
 - show vmmap 67
 - type 68
 - vsn enable 68
- VLAN configuration
 - displaying VSN VLANs 46
 - with VSN 45
- VMware hypervisors 51
- VSN. *See Virtual Server Networking.*
- vSphere client 44

W

- Web interface
 - description 12
 - menu options 77, 79
 - opening and using 76
 - supported web browsers 76

X

- Xenpool hypervisor 44
- XenServer hypervisor 44