Cisco VG410 Voice Gateway Software Configuration Guide

First Published: 2023-10-27

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Preface

This preface describes the audience, organization, and conventions of this document. It also provides information on how to obtain other documentation.

This preface includes the following sections:

Audience and Scope

This document is designed for the person who is responsible for configuring your Cisco Enterprise router. This document is intended primarily for the following audiences:

- Customers with technical networking background and experience.
- System administrators familiar with the fundamentals of router-based internetworking but who might not be familiar with Cisco IOS software.
- System administrators who are responsible for installing and configuring internetworking equipment, and who are familiar with Cisco IOS software.

Feature Compatibility

For more information about the Cisco IOS XE software, including features available on your device as described in the configuration guides, see the respective router documentation set.

To verify support for specific features, use the Cisco Feature Navigator tool. This tool enables you to determine the Cisco IOS XE software images that support a specific software release, feature set, or a platform.

Document Conventions

This documentation uses the following conventions:

Convention	Description
^ or **Ctrl**	The ^ and **Ctrl** symbols represent the Control key. For example, the key combination **^D** or **Ctrl-D** means hold down the **Control** key while you press the **D** key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

The command syntax descriptions use the following conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter exactly as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example, see the following table.

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`bold screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords.
!	An exclamation point at the beginning of a line indicates a comment line. Exclamation points are also displayed by the Cisco IOS XE software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials that may not be contained in this manual.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you’re looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Documentation Feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.

Troubleshooting

For the most up-to-date, detailed troubleshooting information, see the Cisco TAC website at https://www.cisco.com/en/US/support/index.html.

Go to **Products by Category** and choose your product from the list, or enter the name of your product. Look under **Troubleshoot and Alerts** to find information for the issue that you are experiencing.

CHAPTER 1

Overview

The Cisco VG410 Analog Voice Gateway or Cisco VG410 Voice Gateway is a Cisco IOS-XE software-based medium-density analog phone gateway that connects public-switched telephone networks (PSTNs) and existing telephony equipment to Cisco Enterprise Routers.

This voice gateway offers Cisco IOS-XE software manageability on analog phone lines and supports business needs for analog voice ports for modem calls, fax calls, and analog supplementary services.

This device also connects analog phones, fax machines, modems, and speakerphones to the enterprise voice systems and is an intermediate path that enables TDM to IP transition. Further, the fixed-port (FXS and FXO) modules in this voice gateway provide Dual-Tone Multifrequency (DTMF) detection, voice compression and decompression, call progress tone generation, Voice Activity Detection (VAD), echo cancellation, and adaptive jitter buffering.

To know how to install this voice gateway, see the *Cisco VG410 Voice Gateway Installation Guide*. After installing the voice gateway, use this guide to complete basic router configuration using the setup command facility.

Note By default, the Cisco VG410 Voice Gateway boots up in the supported Cisco IOS XE platform versions only. To boot the device in a private image release, contact Cisco Technical Assistance Center (TAC).

This document is a summary of the software functionalities that are specific to Cisco VG410 Voice Gateway. This guide also contains information on using the Cisco IOS software to perform other configuration tasks, such as configuring voice ports and other features.

Features and Benefits

Cisco VG410 Voice Gateway provides VoIP connectivity to analog devices such as analog desk phones, analog conference room phones, fax machines, and modems. This voice gateway provides several improvements from the previous high-density analog and digital extension modules (EVMs) in the following ways:

- **Software Digital Signal Processor (DSP):** The Cisco VG410 Voice Gateway chassis utilizes its built-in CPU cores to handle the digital signal processing (DSP) tasks required for software implementation. This means that the functionality typically provided by a separate DSP component is now distributed among the CPU cores within the device. Further, the CPU cores effectively handle the necessary DSP operations. The software DSP comes pre-installed as part of the manufacturing process.

- **FXS-E (extended loops or long loops) support:** The first 24 ports of all the SKUs on the new modules support FXS-E with:
  - Higher loop current (35 mA) to accommodate specialty phones
  - Longer loop length for loops with 26 AWG wire, up to 11,000 feet (3400 meters)
  - Higher ringing voltage (65 Vrms, no load)

In addition to these features, Cisco VG410 Voice Gateway supports the following features:

- Webex calling
- Caller line ID
- G.711, G.729a, G.729ab, and G.726
- G722, iLBC
- Fax pass-through and relay (T.38)
- Modem pass-through, Modem relay, and V.150.1 MER modem relay support
- DTMF detection
- Echo cancellation
- Voice activity detection
- Comfort noise generation
- Real-Time Control Protocol (RTCP)
- Acoustic shock protection
- Real-Time Transport Protocol (RTP)
- RFC 4733 Digit Relay
- Noise reduction
- Call Details Records (CDR)
- Support for Loop-start and Ground start signaling
- Support for interworking with Cisco Unified Communications Manager (CUCM): Skinny Client Control Protocol (SCCP), Session Initiation Protocol (SIP), and Media Gateway Control Protocol (MGCP) 0.1
- Cable detection: GR909 line test

FXS Features

The FXS features include:

- Support for either FXS or DID functionality
- Message-Waiting Indicator (MWI)

FXO Features

The FXO features include:

- Overload protection

For more information on features, benefits, and other specifications, refer to the Cisco VG410 Voice Gateway Data Sheet.

Supported Interfaces

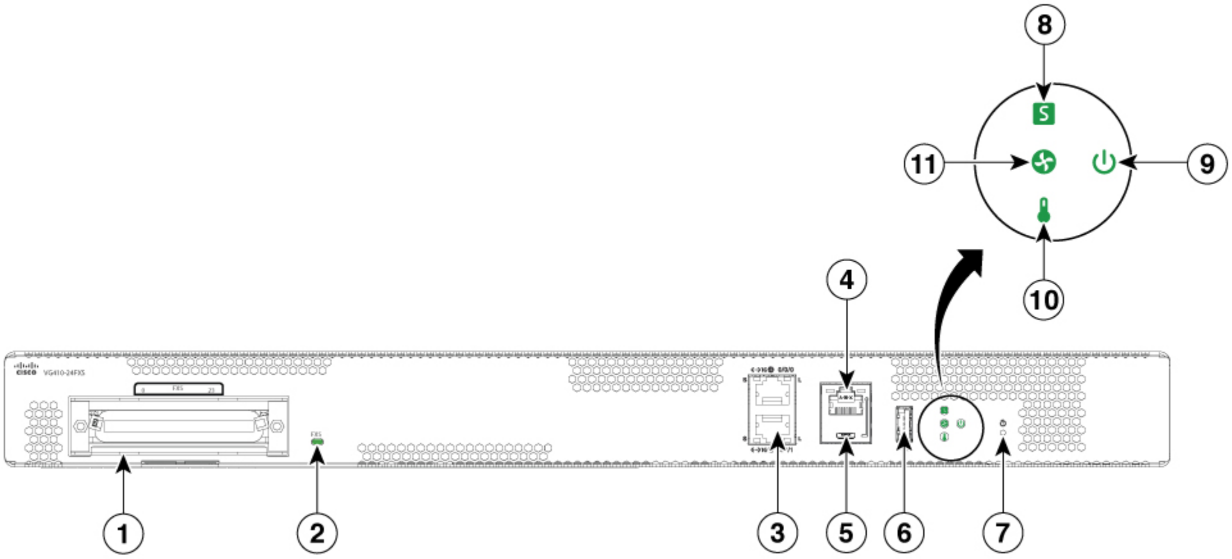
Cisco VG410 Voice Gateway supports the following interfaces:

- Gigabit Ethernet (GE)
- Micro USB console port
- RJ45 console port
- FXS ports
- FXO ports
- USB Type A interface

Identify the Device

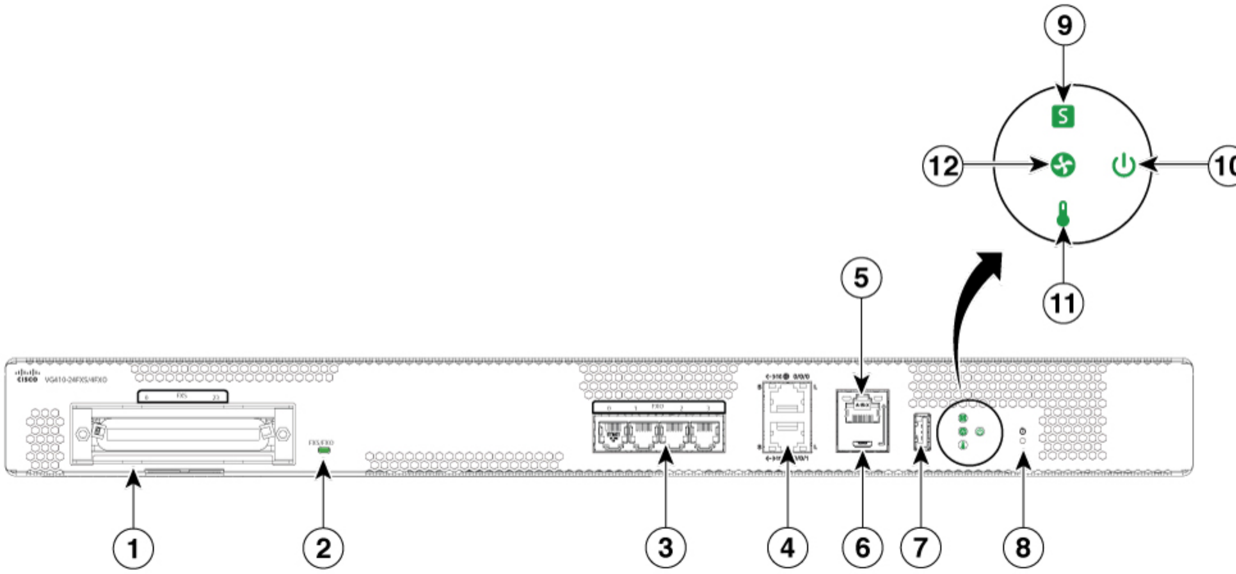
The following images show the I/O panel views of the Cisco VG410 Voice Gateway chassis that help you identify this device:

Figure 1: VG410-24FXS I/O Panel View



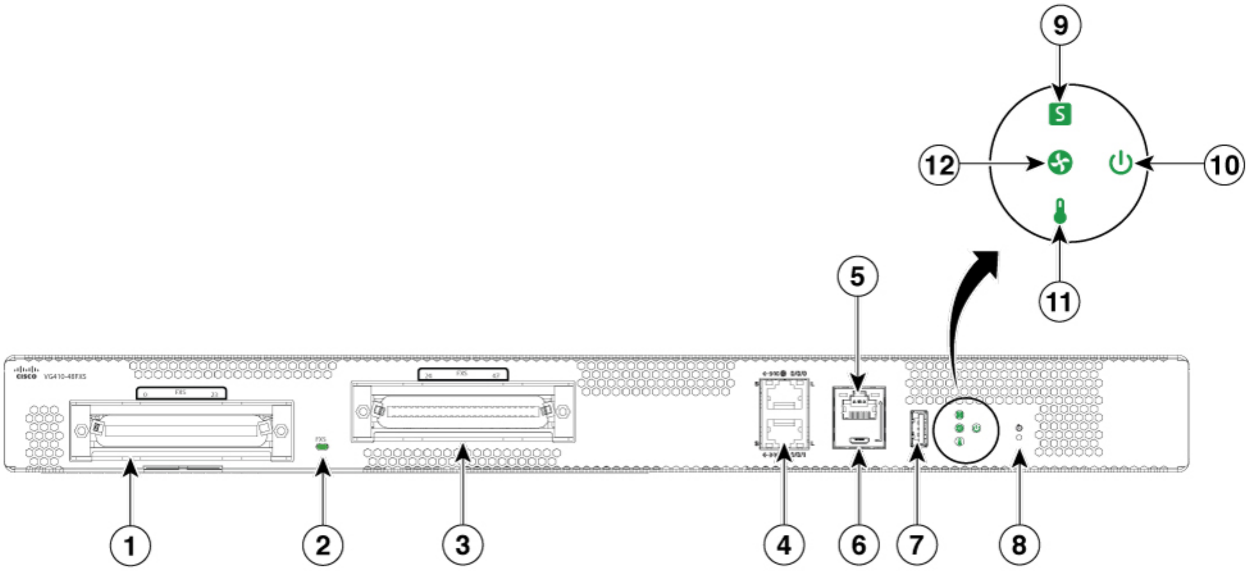
1	RJ 21 for FXS port
2	FXS LED
3	2X1 GE port
4	RJ45 console
5	Micro USB console
6	USB 3.0 type A port
7	Reset
8	System status indicator
9	Power supply status indicator
10	Temperature indicator
11	Environmental status indicator

Figure 2: VG410-24FXS/4FXO I/O Panel View



1	RJ 21 for FXS port
2	FXS/FXO LED
3	RJ 11 for FXO port
4	2X1 GE port
5	RJ 45 console
6	Micro USB console
7	USB 3.0 Type A port
8	Reset
9	System status indicator
10	Power Supply status indicator
11	Temperature indicator
12	Environmental status indicator

Figure 3: VG410-48FXS I/O Panel View



1	RJ 21 for FXS port
2	FXS LED
3	RJ 21 for FXS port
4	2X1 GE port
5	RJ45 console
6	Micro USB console
7	USB 3.0 Type A port
8	Reset
9	System status indicator
10	Power status indicator
11	Temperature indicator
12	Environmental status indicator

CHAPTER 2

Understanding Cisco IOS Software Basics

This section describes what you need to know about the Cisco IOS software before you configure the router using the CLI. Understanding these concepts will save time as you begin to use the commands. If you have never used Cisco IOS software or need a refresher, take a few minutes to read this chapter before you proceed to the next chapter.

If you are already familiar with Cisco IOS software, proceed to the *Configuring Host Name and Password* section in this guide.

Basics Before Using Commands

The following table specifies some basic rules and notes to configure the device by using the command line interface. Use the question mark (?) and arrow keys to help you enter commands:

Rule	Example
For a list of available commands, enter a question mark.	`Router> ?`
To complete a command, enter a few known characters followed by a question mark (with no space).	`Router> s?`
For a list of command variables, enter the command followed by a space and a question mark.	`Router> show ?`
To redisplay a command you previously entered, press the Up Arrow key. You can continue to press the Up Arrow key for more commands.	

Command Modes

The Cisco IOS user interface is divided into different modes. Each command mode permits you to configure different components on your router. The commands available at any given time depend on which mode you are currently in.

Entering a question mark (**?**) at the prompt displays a list of commands available for each command mode. The following table lists the most common command modes:

Table 1: Common Command Modes

Command Mode	Access Method	Router Prompt Displayed	Exit Method
User EXEC	Log in	Router>	Use the **logout** command.
Privileged EXEC	From user EXEC mode, enter the **enable** command.	Router#	To exit to user EXEC mode, use the **disable**, **exit**, or **logout** command.
Global configuration	From the privileged EXEC mode, enter the **configure terminal** command.	Router (config)#	To exit to privileged EXEC mode, use the **exit** or **end** command, or press **Ctrl-Z**.
Interface configuration	From the global configuration mode, enter the GigabitEthernet interface command such as, **gigabitethernet0/0**.	Router (config-if)#	To exit to global configuration mode, use the **exit** command. To exit directly to privileged EXEC mode, press **Ctrl-Z**.

Timesaver Each command mode restricts you to a subset of commands. If you are having trouble entering a command, check the prompt, and enter the question mark (?) for a list of available commands. You might be in the wrong command mode or be using the wrong syntax.

In the following example, notice how the prompt changes after each command, to indicate a new command mode for the voice gateway:

```
Router> enable
Password: <enable password>
Router# configure terminal
Router(config)# interface gigabitEthernet 0/0/0
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

The last message is normal and does not indicate an error. Press **Return** to get the Router# prompt.

Note Press **Ctrl-Z** in any mode to immediately return to enable mode (Router#) instead of entering **exit**, which returns you to the previous mode.

Undoing a Command or Feature

If you want to undo a command you entered or disable a feature, enter the keyword **no** before most commands.

For example, **no ip routing**.

Saving Configuration Changes

To prevent the loss of your device configuration, save the configuration changes to NVRAM.

Step 1 Router> enable

Example:

```
Password: password
```

Example:

```
Router#
```

Enables the privileged EXEC mode. Enter your password, if prompted.

Step 2 Router# copy running-config startup-config

Saves the configuration changes to NVRAM so that the changes are not lost during resets, power cycles, or power outages.

Step 3 Router(config-if)# Ctrl-z

Example:

```
Router#
```

Example:

```
%SYS-5-CONFIG_I: Configured from console by console
```

Returns to the user EXEC mode.

Upgrading to a New Cisco IOS Release

To install or upgrade to a new Cisco IOS release, see How to Update or Upgrade Cisco IOS Software .

Note For Cisco VG410 Voice Gateway, the DSP container will be automatically upgraded when you upgrade the Cisco IOS-XE image.

Where to Go Next

Now that you have learned some Cisco IOS software basics, you can begin to configure the router using the CLI. However, before you begin, here are a few useful tips.

- Use the question mark (?) and arrow keys to help you enter commands.
- Each command mode restricts you to a set of commands. If you have difficulty entering a command, check the prompt and then enter the question mark (**?**) for a list of available commands. You might be in the wrong command mode or you might be using the wrong syntax.
- To disable a feature, enter the keyword **no** before the command. For example, **no ip routing**.
- Save your configuration changes to NVRAM so that the changes are not lost if there is a system reload or power outage.

To begin the configuration of the router, proceed to the *Configuring the Host Name and Password* section in this guide.

CHAPTER 3

Installing the Software Using install Commands

From Cisco IOS XE Cupertino 17.9.1a, Cisco Voice Gateways VG400, VG420, and VG450 are shipped in install mode by default. From Cisco IOS XE 17.12.1a, Cisco Voice Gateway VG410 is also shipped in the install mode. You can boot the platform, and upgrade or downgrade to Cisco IOS XE software versions using a set of **install** commands that are detailed in the following sections.

Restrictions for Installing the Software Using install Commands

- ISSU is not covered in this feature.
- Install mode requires a reboot of the system.

Information About Installing the Software Using install Commands

From Cisco IOS XE Cupertino 17.9.1a release, for devices shipped in install mode, a set of **install** commands can be used for starting, upgrading and downgrading of platforms in install mode. This update is applicable to the Cisco Voice Gateway 400 Series.

The following table describes the differences between Bundle mode and Install mode:

Table 2: Bundle Mode vs Install Mode

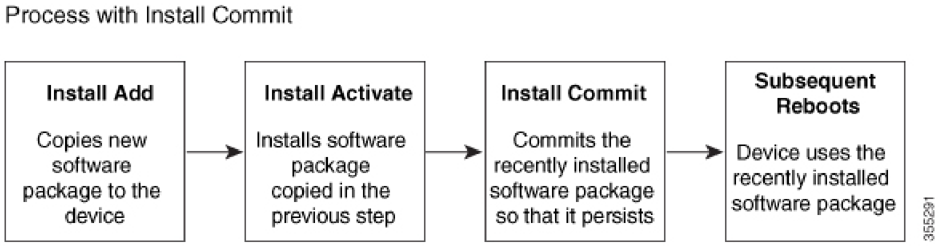
Bundle Mode	Install Mode
This mode provides a consolidated boot process, using local (hard disk, flash) or remote (TFTP) .bin image.	This mode uses the local (bootflash) packages.conf file for the boot process.
This mode uses a single .bin file.	.bin file is replaced with expanded .pkg files in this mode.

Bundle Mode	Install Mode
CLI: `#boot system file <filename>`	CLI: `#install add file bootflash: [activate commit]`
To upgrade in this mode, point the boot system to the new image.	To upgrade in this mode, use the **install** commands.

Install Mode Process Flow

The install mode process flow comprises three commands to perform installation and upgrade of software on platforms–**install add**, **install activate**, and **install commit**.

The following flow chart explains the install process with **install** commands:


Process with Install Commit
Install Add
Copies new software package to the device
Install Activate
Installs software package copied in the previous step
Install Commit
Commits the recently installed software package so that it persists
Subsequent Reboots
Device uses the recently installed software package
355291

The **install add** command copies the software package from a local or remote location to the platform. The location can be FTP, HTTP, HTTPs, or TFTP. The command extracts individual components of the .package file into subpackages and packages.conf files. It also validates the file to ensure that the image file is specific to the platform on which it is being installed.

The **install activate** command performs the required validations and provisions the packages previously added using the **install add** command. It also triggers a system reload.

The **install commit** command confirms the packages previously activated using the **install activate** command, and makes the updates persistent over reloads.

Note Installing an update replaces any previously installed software image. At any time, only one image can be installed in a device.

The following set of install commands is available:

Table 3: List of install Commands

Command	Syntax	Purpose
install add	**install add file** *location:filename.bin*	Copies the contents of the image and the package to the software repository. File location may be local or remote. This command does the following: • Validates the file–checksum, platform compatibility checks, and so on. • Extracts individual components of the package into subpackages and packages.conf • Copies the image into the local inventory and makes it available for the next steps.
install activate	**install activate**	Activates the package added using the **install add** command. • Use the **show install summary** command to see which image is inactive. This image will get activated. • System reloads on executing this command. Confirm if you want to proceed with the activation. Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts.

Command	Syntax	Purpose
(install activate) auto abort-timer	**install activate auto-abort timer** *<30-1200>*	The **auto-abort timer** starts automatically, with a default value of 120 minutes. If the **install commit** command is not executed within the time provided, the activation process is terminated, and the system returns to the last-committed state. • You can change the time value while executing the **install activate** command. • The **install commit** command stops the timer, and continues the installation process. • The **install activate auto-abort timer stop** command stops the timer without committing the package. • Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts. • This command is valid only in the three-step install variant.
install commit	**install commit**	Commits the package activated using the **install activate** command, and makes it persistent over reloads. • Use the **show install summary** command to see which image is uncommitted. This image will get committed.

Command	Syntax	Purpose
install abort	**install abort**	Terminates the installation and returns the system to the last-committed state. • This command is applicable only when the package is in activated status (uncommitted state). • If you have already committed the image using the **install commit** command, use the **install rollback to** command to return to the preferred version.
install remove	**install remove {file** *<filename>* **| inactive}**	Deletes inactive packages from the platform repository. Use this command to free up space. • **file**: Removes specified files. • **inactive**: Removes all the inactive files.
install rollback to	**install rollback to {base | label | committed | id}**	Rolls back the software set to a saved installation point or to the last-committed installation point. The following are the characteristics of this command: • Requires reload. • Is applicable only when the package is in committed state. • Use this command with the **prompt-level none** keyword to automatically ignore any confirmation prompts. **Note** If you are performing install rollback to a previous image, the previous image must be installed in install mode.

The following show commands are also available:

Table 4: List of show Commands

Command	Syntax	Purpose
show install log	**show install log**	Provides the history and details of all install operations that have been performed since the platform was booted.
show install package	**show install package** <*filename*>	Provides details about the .pkg/.bin file that is specified.
show install summary	**show install summary**	Provides an overview of the image versions and their corresponding install states.
show install active	**show install active**	Provides information about the active packages.
show install inactive	**show install inactive**	Provides information about the inactive packages, if any.
show install committed	**show install committed**	Provides information about the committed packages.
show install uncommitted	**show install uncommitted**	Provides information about uncommitted packages, if any.
show install rollback	**show install rollback {point-id | label}**	Displays the package associated with a saved installation point.
show version	**show version [rp-slot] [installed [user-interface] | provisioned | running]**	Displays information about the current package, along with hardware and platform information.

Booting the Platform in Install Mode

You can install, activate, and commit a software package using a single command (one-step install) or multiple separate commands (three-step install).

If the platform is working in bundle mode, the one-step install procedure must be used to initially convert the platform from bundle mode to install mode. Subsequent installs and upgrades on the platform can be done with either one-step or three-step variants.

One-Step Installation or Converting from Bundle Mode to Install Mode

Note

- All the CLI actions (for example, add, activate, and so on) are executed.
- The configuration save prompt will appear if an unsaved configuration is detected.
- The reload prompt will appear after the second step in this workflow. Use the **prompt-level none** keyword to automatically ignore the confirmation prompts.
- If the prompt-level is set to None, and there is an unsaved configuration, the install fails. You must save the configuration before reissuing the command.

Use the one-step install procedure described below to convert a platform running in bundle boot mode to install mode. After the command is executed, the platform reboots in install boot mode.

Later, the one-step install procedure can also be used to upgrade the platform.

This procedure uses the **install add file activate commit** command in privileged EXEC mode to install a software package, and to upgrade the platform to a new version.

SUMMARY STEPS

1. **enable**
2. **install add file location:** *filename* [**activate commit**]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**install add file location:** *filename* [**activate commit**] **Example:** `Device# install add file bootflash:vg4x0-universalk9.17.12.01a.SPA.bin activate commit`	Copies the software install package from a local or remote location (through FTP, HTTP, HTTPs, or TFTP) to the platform and extracts the individual components of the .package file into subpackages and packages.conf files. It also performs a validation and compatibility check for the platform and image versions, activates the package, and commits the package to make it persistent across reloads. The platform reloads after this command is run.
Step 3	**exit** **Example:** `Device# exit`	Exits privileged EXEC mode and returns to user EXEC mode.

Three-Step Installation

Note

- All the CLI actions (for example, add, activate, and so on) are executed.
- The configuration save prompt will appear if an unsaved configuration is detected.
- The reload prompt will appear after the install activate step in this workflow. Use the **prompt-level none** keyword to automatically ignore the confirmation prompts.

The three-step installation procedure can be used only after the platform is in install mode. This option provides more flexibility and control to the customer during installation.

This procedure uses individual **install add**, **install activate**, and **install commit** commands for installing a software package, and to upgrade the platform to a new version.

SUMMARY STEPS

1. **enable**
2. **install add file location:** *filename*
3. **show install summary**
4. **install activate** [**auto-abort-timer** <*time*>]
5. **install abort**
6. **install commit**
7. **install rollback to committed**
8. **install remove** {**file** *filesystem: filename* | **inactive**}
9. **show install summary**
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**install add file location:** *filename* **Example:** `Device# install add file bootflash:vg4x0-universalk9.17.12.01a.SPA.bin`	Copies the software install package from a remote location (through FTP, HTTP, HTTPs, or TFTP) to the platform, and extracts the individual components of the .package file into subpackages and packages.conf files.
Step 3	**show install summary** **Example:** `Device# show install summary`	(Optional) Provides an overview of the image versions and their corresponding install state.
Step 4	**install activate** [**auto-abort-timer** <*time*>] **Example:**	Activates the previously added package and reloads the platform.

	Command or Action	Purpose
	`Device# install activate auto-abort-timer 120`	• When doing a full software install, do not provide a package filename. • In the three-step variant, **auto-abort-timer** starts automatically with the **install activate** command; the default for the timer is 120 minutes. If the **install commit** command is not run before the timer expires, the install process is automatically terminated. The platform reloads and boots up with the last committed version.
Step 5	**install abort** **Example:** `Device# install abort`	(Optional) Terminates the software install activation and returns the platform to the last committed version. • Use this command only when the image is in activated state and not when the image is in committed state.
Step 6	**install commit** **Example:** `Device# install commit`	Commits the new package installation and makes the changes persistent over reloads.
Step 7	**install rollback to committed** **Example:** `Device# install rollback to committed`	(Optional) Rolls back the platform to the last committed state.

Step 8	**install remove** {**file** *filesystem: filename* | **inactive**}
	Example:
	`Device# install remove inactive`
		(Optional) Deletes the software installation files.
		• **file**: Deletes a specific file.
		• **inactive**: Deletes all the unused and inactive installation files.

Step 9	**show install summary** **Example:** `Device# show install summary`	(Optional) Displays information about the current state of the system. The output of this command varies according to the **install** commands run prior to this command.
Step 10	**exit** **Example:** `Device# exit`	Exits privileged EXEC mode and returns to the user EXEC mode.

Upgrading to a New Cisco IOS Release

To install or upgrade to a new Cisco IOS release, see How to Update or Upgrade Cisco IOS Software .

Note For Cisco VG410 Voice Gateway, the vDSP container is automatically upgraded when you upgrade the Cisco IOS XE image.

Downgrading in Install Mode

Use the **install rollback** command to downgrade the platform to a previous version by pointing it to the appropriate image, provided the image you are downgrading to was installed in install mode.

The **install rollback** command reloads the platform and boots it with the previous image.

Note The **install rollback** command succeeds only if you have not removed the previous file using the **install remove inactive** command.

Alternatively, you can downgrade by installing the older image using the **install** commands.

Terminating a Software Installation

You can terminate the activation of a software package in the following ways:

- When the platform reloads after activating a new image, the auto-abort-timer is triggered (in the three-step install variant). If the timer expires before issuing the **install commit** command, the installation process is terminated, and the platform reloads and boots with the last committed version of the software image.

  Alternatively, use the **install auto-abort-timer stop** command to stop this timer, without using the **install commit** command. The new image remains uncommitted in this process.

- Using the **install abort** command returns the platform to the version that was running before installing the new software. Use this command before issuing the **install commit** command.

Configuration Examples: Install the Software Using Install Commands

The following is an example of the one-step installation or converting from bundle mode to install mode:

```
vg410# install add file flash:vg4x0-universalk9.17.12.01a.SPA.bin
*Sep 22 16:05:26.116: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private config
 file

*Sep 22 16:05:29.836: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install add
 bootflash:vg4x0-universalk9.17.12.01a.SPA.bin install_add: START Fri Sep 22 16:05:29 UTC
2023
install_add: Adding IMG
 [1]  R0 FAILED: Booted in bundle mode. For Bundle-to-Install mode conversion, please use
one-shot CLI - install add file <> activate commit
FAILED: install_add /bootflash/vg4x0-universalk9.17.12.01a.SPA.bin Fri Sep 22 16:05:29 UTC
 2023

vg410#
*Sep 22 16:05:29.841: %INSTALL-3-OPERATION_ERROR_MESSAGE: R0/0: install_mgr: Failed to
install add package bootflash:/vg4x0-universalk9.17.12.01a.SPA.bin, Error: Booted in bundle
 mode. For Bundle-to-Install mode conversion, please use one-shot CLI - install add file
<> activate commitinstall add file flash:vg4x0-univer$ file
flash:vg4x0-universalk9.17.12.01a.SPA.bin activate ?
  commit  Commit the changes to the loadpath

vg410#$ file flash:vg4x0-universalk9.17.12.01a.SPA.bin activate com
install_add_activate_commit: START Fri Sep 22 16:06:47 UTC 2023
```

```
install_add: START Fri Sep 22 16:06:47 UTC 2023
install_add: Adding IMG
--- Starting initial file syncing ---
Copying bootflash:vg4x0-universalk9.17.12.01a.SPA.bin from  R0 to  R0
Info: Finished copying to the selected
Finished initial file syncing

--- Starting Add ---
Performing Add on all members

*Sep 22 16:06:47.521: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install
add_activate_commit bootflash:vg4x0-universalk9.17.12.01a.SPA.bin
Checking status of Add on [R0]
Add: Passed on [R0]
Image added. Version: 17.12.01a.0.118

Finished Add

install_activate: START Fri Sep 22 16:06:55 UTC 2023
install_activate: Activating IMG
Following packages shall be activated:
/bootflash/vg4x0-firmware_vg4x0_vdsp.17.12.01a.SPA.pkg
/bootflash/vg4x0-mono-universalk9.17.12.01a.SPA.pkg
/bootflash/vg4x0-rpboot.17.12.01a.SPA.pkg

This operation may require a reload of the system. Do you want to proceed? [y/n]
*Sep 22 16:06:55.053: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install
activate NONEy

--- Starting Activate ---
Performing Activate on all members

 [1] Activate package(s) on  R0

*Sep 22 16:07:11.447: %INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: R0/0: rollback_timer:
Install auto abort timer will expire in 7200 seconds
Building configuration...
[OK] [1] Finished Activate on  R0
Checking status of Activate on [R0]
Activate: Passed on [R0]
Finished Activate

--- Starting Commit ---
Performing Commit on all members
 [1] Commit package(s) on  R0

*Sep 22 16:07:25.031: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private config
 file [1] Finished Commit on  R0
Checking status of Commit on [R0]
Commit: Passed on [R0]
Finished Commit operation

SUCCESS: install_add_activate_commit Fri Sep 22 16:07:35 UTC 2023

vg410#
*Sep 22 16:07:35.004: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_mgr: Completed install
 add_activate_commitS




Initializing Hardware ...

Checking for PCIe device presence...done
```

```
System integrity status: 0x610



Rom image verified correctly


System Bootstrap, Version 17.12(1r), RELEASE SOFTWARE
Copyright (c) 1994-2023 by cisco Systems, Inc.


Current image running: Boot ROM0

Last reset cause: LocalSoft
VG410-48FXS platform with 8388608 Kbytes of main memory


........
Located packages.conf
#

###############################

Package header rev 3 structure detected
IsoSize = 0
Calculating SHA-1 hash...Validate package: SHA-1 hash:
 calculated 226B404A:303E3E89:749B2335:BDB2A32C:6164E25A
 expected   226B404A:303E3E89:749B2335:BDB2A32C:6164E25A
Validate package: start secure boot validation

Secure verification of the image PASSED
Sep 22 16:09:29.919: %BOOT-5-OPMODE_LOG: R0/0: binos: System booted in AUTONOMOUS mode

              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           Cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706



Cisco IOS Software [Dublin], vg4x0 Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version
17.12.1a, RELEASE SOFTWARE (fc3)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2023 by Cisco Systems, Inc.
Compiled Sat 19-Aug-23 00:41 by mcpre


This software version supports only Smart Licensing as the software licensing mechanism.


PLEASE READ THE FOLLOWING TERMS CAREFULLY. INSTALLING THE LICENSE OR
LICENSE KEY PROVIDED FOR ANY CISCO SOFTWARE PRODUCT, PRODUCT FEATURE,
AND/OR SUBSEQUENTLY PROVIDED SOFTWARE FEATURES (COLLECTIVELY, THE
"SOFTWARE"), AND/OR USING SUCH SOFTWARE CONSTITUTES YOUR FULL
ACCEPTANCE OF THE FOLLOWING TERMS. YOU MUST NOT PROCEED FURTHER IF YOU
ARE NOT WILLING TO BE BOUND BY ALL THE TERMS SET FORTH HEREIN.
```

```
Your use of the Software is subject to the Cisco End User License Agreement
(EULA) and any relevant supplemental terms (SEULA) found at
http://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html.

You hereby acknowledge and agree that certain Software and/or features are
licensed for a particular term, that the license to such Software and/or
features is valid only for the applicable term and that such Software and/or
features may be shut down or otherwise terminated by Cisco after expiration
of the applicable license term (e.g., 90-day trial period). Cisco reserves
the right to terminate any such Software feature electronically or by any
other means available. While Cisco may provide alerts, it is your sole
responsibility to monitor your usage of any such term Software feature to
ensure that your systems and networks are prepared for a shutdown of the
Software feature.


cisco VG410-48FXS (1RU) processor with 3686972K/6147K bytes of memory.
Processor board ID FGL2731LMP4
Router operating mode: Autonomous
2 Gigabit Ethernet interfaces
48 Voice FXS interfaces
32768K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
7573503K bytes of flash memory at bootflash:.

 WARNING: Command has been added to the configuration using a type 0 password. However,
recommended to migrate to strong type-6 encryption
SETUP: new interface Service-Engine0/1/0 placed in "shutdown" state

 WARNING: ** NOTICE **  The H.323 protocol is no longer supported from IOS-XE release 17.6.1.
 Please consider using SIP for multimedia applications.



Press RETURN to get started!


*Sep 22 16:09:39.583: %CRYPTO-5-SELF_TEST_START: Crypto algorithms release (Rel5a), Entropy
 release (3.4.1)
       begin self-test
*Sep 22 16:09:39.841: %CRYPTO-5-SELF_TEST_END: Crypto algorithms self-test completed
successfully
       All tests passed.
*Sep 22 16:09:42.115: %ISR_THROUGHPUT-6-LEVEL: Throughput level has been set to 1000000
kbps
*Sep 22 16:09:42.818: %SMART_LIC-6-AGENT_ENABLED: Smart Agent for Licensing is enabled
*Sep 22 16:09:43.143: %SMART_LIC-6-EXPORT_CONTROLLED: Usage of export controlled features
is not allowed
*Sep 22 16:09:47.210: %SPANTREE-5-EXTENDED_SYSID: Extended SysId enabled for type vlan
*Sep 22 16:09:47.342: %CRYPTO_ENGINE-5-CSDL_COMPLIANCE_ENFORCED: Cisco PSB security compliance
 is being enforced
*Sep 22 16:09:47.390: %CUBE-3-LICENSING:  SIP trunking (CUBE) licensing is now based on
dynamic sessions counting, static license capacity configuration through 'mode border-element
 license capacity' would be ignored.
*Sep 22 16:09:47.404: %SIP-5-LICENSING: CUBE license reporting period has been set to the
minimum value of 8 hours.
*Sep 22 16:09:47.462: %VOICE_HA-7-STATUS: CUBE HA-supported platform
detected.pm_platform_init() line :3156

*Sep 22 16:09:47.799: %LINK-3-UPDOWN: Interface EOBC0, changed state to up
*Sep 22 16:09:47.854: %LINK-3-UPDOWN: Interface Lsmpi0, changed state to up
*Sep 22 16:09:47.854: %LINEPROTO-5-UPDOWN: Line protocol on Interface LI-Null0, changed
state to up
```

```
*Sep 22 16:09:47.854: %LINEPROTO-5-UPDOWN: Line protocol on Interface VoIP-Null0, changed
state to up
*Sep 22 16:09:47.855: %LINK-3-UPDOWN: Interface LIIN0, changed state to up
*Sep 22 16:09:47.983: %VOICE_HA-7-STATUS: Create VOICE HA INFRA processes now....
*Sep 22 16:09:47.999: %PNP-6-PNP_DISCOVERY_STARTED: PnP Discovery started
*Sep 22 16:09:29.916: %BOOT-5-OPMODE_LOG: R0/0: binos: System booted in AUTONOMOUS mode
*Sep 22 16:09:36.826: %CMRP_PFU-6-FANASSY_INSERTED: R0/0: cmand: Fan Assembly is inserted.
*Sep 22 16:09:48.816: %LINEPROTO-5-UPDOWN: Line protocol on Interface EOBC0, changed state
 to up
*Sep 22 16:09:48.865: %LINEPROTO-5-UPDOWN: Line protocol on Interface Lsmpi0, changed state
 to up
*Sep 22 16:09:48.865: %LINEPROTO-5-UPDOWN: Line protocol on Interface LIIN0, changed state
 to up
*Sep 22 16:09:49.205: %ONEP_BASE-6-SS_ENABLED: ONEP: Service set Base was enabled by Default
*Sep 22 16:09:51.771: %SYS-7-NVRAM_INIT_WAIT_TIME: Waited 0 seconds for NVRAM to be available
*Sep 22 16:09:52.442: %CRYPTO_ENGINE-5-KEY_ADDITION: A key named TP-self-signed-3402504622
 has been generated or imported by crypto config
*Sep 22 16:09:52.445: %SYS-6-PRIVCFG_DECRYPT_SUCCESS: Successfully apply the private config
 file
*Sep 22 16:09:52.512: %SYS-5-LOG_CONFIG_CHANGE: Buffer logging: level debugging, xml disabled,
 filtering disabled, size (50000000)
*Sep 22 16:09:52.519:
```

The following is an example of the three-step installation:

```
vg410# install add bootflash:vg4x0-universalk9.vg4x0-universalk9.17.12.01a.SPA.bin

Sep 24 07:39:28.863: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install
add_activate_commit bootflash:vg4x0-universalk9.vg4x0-universalk9.17.12.01a.SPA.bin
install_add_activate_commit: START Sun Sep 24 07:39:28 UTC 2023
install_add: START Sun Sep 24 07:39:28 UTC 2023
install_add: Adding IMG
--- Starting initial file syncing ---
Copying bootflash:vg4x0-universalk9.vg4x0-universalk9.17.12.01a.SPA.bin from  R0 to  R0
Info: Finished copying to the selected
Finished initial file syncing

--- Starting Add ---
Performing Add on all members
Checking status of Add on [R0]
Add: Passed on [R0]
Image added. Version: 17.12.01.0.186080

Finished Add


install_activate: START Sun Sep 24 07:40:26 UTC 2023
install_activate: Activating IMG
Following packages shall be activated:
/bootflash/vg4x0-firmware_vg4x0_vdsp.BLD_POLARIS_DEV_LATEST_20230910_172549_V17_14_0_3.SSA.pkg
/bootflash/vg4x0-mono-universalk9.BLD_POLARIS_DEV_LATEST_20230910_172549_V17_14_0_3.SSA.pkg
/bootflash/vg4x0-rpboot.BLD_POLARIS_DEV_LATEST_20230910_172549_V17_14_0_3.SSA.pkg

This operation may require a reload of the system. Do you want to proceed? [y/n]
*Sep 24 07:40:26.929: %INSTALL-5-INSTALL_START_INFO: R0/0: install_mgr: Started install
activate NONEy


--- Starting Activate ---
Performing Activate on all members
 [1] Activate package(s) on  R0
*Sep 24 07:40:47.197: %INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: R0/0: rollback_timer:
Install auto abort timer will expire in 7200 seconds
```

```
Building configuration...
[OK] [1] Finished Activate on  R0
Checking status of Activate on [R0]
Activate: Passed on [R0]
Finished Activate
```

```
vg410# install commit
--- Starting Commit ---
Performing Commit on all members
 [1] Commit package(s) on  R0

*Sep 24 07:41:05.121: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private config
 file [1] Finished Commit on  R0
Checking status of Commit on [R0]
Commit: Passed on [R0]
Finished Commit operation

SUCCESS: install_add_activate_commit Sun Sep 24 07:41:20 UTC 2023

vg410#
*Sep 24 07:41:20.211: %INSTALL-5-INSTALL_COMPLETED_INFO: R0/0: install_mgr: Completed install
 add_activate_commitSep 24 07:41:34.778: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is
 exiting: reload action requested
```

```
Initializing Hardware ...

Checking for PCIe device presence...done
System integrity status: 0x610

Rom image verified correctly

System Bootstrap, Version 17.12(1r), RELEASE SOFTWARE
Copyright (c) 1994-2023 by cisco Systems, Inc.

Current image running: Boot ROM1

Last reset cause: LocalSoft
VG410-24FXS/4FXO platform with 8388608 Kbytes of main memory

........
Located packages.conf
```

The following is an example of terminating a software installation:

```
vg410# install abort
install_abort: START Mon Sep 25 09:15:34 UTC 2023

This operation may require a reload of the system. Do you want to proceed? [y/n]y

--- Starting Abort ---
Performing Abort on all members
 [1] Abort packages(s) on  R0
Checking status of Abort on [R0]
Abort: Passed on [R0]
Finished Abort operation
```

```
SUCCESS: install_abort START Mon Sep 25 09:15:34 UTC 2023
vg410# Mon Sep 25 09:15:34: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting:
reload action requested

Initializing Hardware ...
  :
  :
  Press RETURN to get started!

vg410>
```

The following are sample outputs for show commands:

show version

```
vg410# show version
Cisco IOS XE Software, Version 17.12.01a
Cisco IOS Software [Dublin], vg4x0 Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version
17.12.1a, RELEASE SOFTWARE (fc3)
Copyright (c) 1986-2023 by Cisco Systems, Inc.
Compiled Sun 10-Sep-23 12:48 by mcpre


Cisco IOS-XE software, Copyright (c) 2005-2023 by cisco Systems, Inc.
All rights reserved.  Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0.  The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY.  You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.  For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.


ROM: 17.12(1r)
)

VG410 uptime is 53 minutes
Uptime for this control processor is 54 minutes
System returned to ROM by Reload Command
System image file is "bootflash/vg4x0-universalk9.17.12.01a.SPA.bin"
Last reload reason: Reload Command


This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.


Suite License Information for Module:'esg'

--------------------------------------------------------------------------------
```

```
Suite Suite Current Type Suite Next reboot
--------------------------------------------------------------------------------

Technology Package License Information:

-----------------------------------------------------------------
Technology Technology-package Technology-package
Current Type Next reboot
------------------------------------------------------------------
uck9 uck9 Smart License uck9
securityk9 securityk9 Smart License securityk9
ipbase ipbasek9 Smart License ipbasek9

The current throughput level is unthrottled


Smart Licensing Status: Smart Licensing Using Policy

cisco VG410-24FXS/4FXO (1RU) processor with 3686896K/6147K bytes of memory.
Processor board ID FGL2731LMZY
Router operating mode: Autonomous
2 Gigabit Ethernet interfaces
4 Voice FXO interfaces
24 Voice FXS interfaces
32768K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
7573503K bytes of flash memory at bootflash:.

Configuration register is 0x0
```

show install log

```
vg410# show install log
[0|install_op_boot]: START Sun Sep 24 07:42:52 Universal 2023
[0|install_op_boot(INFO, )]: Mount IMG INI state base image
[0|install_op_boot]: END SUCCESS  Sun Sep 24 07:42:53 Universal 2023
```

show install summary

```
vg410# show install summary
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    17.12.01.0.186080

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

show install package *filesystem: filename*

```
vg410# show install package flash:vg4x0-universalk9.17.12.01a.SPA.bin
  Package: vg4x0-universalk9.17.12.01a.SPA.bin
    Size: 658481669
    Timestamp:
  Canonical path: /bootflash/vg4x0-universalk9.17.12.01a.SPA.bin

    Raw disk-file SHA1sum:
      9c43dfa47b2cb6591f71bbf461cde8d51291bb8a
  Header size:     1040 bytes
  Package type:    30000
  Package flags:   0
  Header version:  3

  Internal package information:
    Name: rp_super
    BuildTime: 2023-07-27_23.17
    ReleaseDate: 2023-07-28_05.52
    BootArchitecture: i686
    RouteProcessor: vg4x0
    Platform: VG4X0
    User: occp
    PackageName: universalk9
    Build: 17.12.01a
    CardTypes:
  Package is bootable from media and tftp
```

show install active

```
vg410# show install active
[ R0 ] Active Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    17.12.01.0.186080

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

show install inactive

```
vg410# show install inactive
[ R0 ] Inactive Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
No Inactive Packages
```

show install committed

```
vg410_B# show install committed
[ R0 ] Committed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    17.12.01.0.186080

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

show install uncommitted

```
vg410# show install uncommitted
[ R0 ] Uncommitted Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
```

```
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
No Uncommitted Packages
```

Troubleshooting Software Installation Using install Commands

Problem Troubleshooting the software installation

Solution Use the following show commands to view installation summary, logs, and software versions.

- **show install summary**
- **show install log**
- **show version**
- **show version running**

Problem Other installation issues

Solution Use the following commands to resolve installation issue:

- **dir** <*install directory*>
- **more location:***packages.conf*
- **show tech-support install**: this command automatically runs the **show** commands that display information specific to installation.
- **request platform software trace archive target bootflash** <*location*>: this command archives all the trace logs relevant to all the processes running on the system since the last reload, and saves this information in the specified location.

CHAPTER 4

Configuring the Cisco VG410 Voice Gateway

This chapter describes how to use the Cisco IOS software CLI to configure basic analog functionalities. Follow the procedures in this chapter to configure Cisco VG410 Voice Gateway, or if you want to change the configuration after you have run the setup command facility.

This chapter does not describe every configuration possible—only a small portion of the most commonly used configuration procedures. For advanced configuration topics, refer to the respective technology configuration guides.

One of the first configuration tasks you might want to do is to configure the host name and set an encrypted password. Configuring a host name allows you to distinguish a router from another. Setting an encrypted password allows you to prevent unauthorized configuration changes. Read on to know how to perform these configurations.

Configuring the Host Name and Password

Procedure

	Command or Action	Purpose

Step 1	**Device> enable**
	Example:
	`Password: password`
	Example:
	`Device#`
		Enables privileged EXEC mode. Enter your password, if prompted.

Step 2	**configure terminal** **Example:** `Device(config)#`	Enters global configuration mode. Enter configuration commands, one per line. End with CNTL/Z.

	Command or Action	Purpose
Step 3	**hostname vg410** **Example:** `Device(config)# hostname vg410`	Changes the name of Cisco VG410 Voice Gateway to a meaningful name. Substitutes the host name to Device.
Step 4	**enable secret <password>** **Example:** `vg410(config)# enable secret guessme`	Enters an enable secret password. This password provides access to privileged EXEC mode. When you enter enable at the user EXEC prompt, you must enter the enable secret password to gain access to configuration mode. Enter the secret password, for example, guessme.
Step 5	**line con 0** **Example:** `vg410(config)# line con 0`	Enters line configuration mode to configure the console port.
Step 6	**exec-timeout 0 0** **Example:** `vg410(config-line)# exec-timeout 0 0`	Prevents Cisco VG410 Voice Gateway EXEC mode from timing out when you do not enter any information on the console screen for an extended period.
Step 7	**exit** **Example:** `vg410(config-line)# exit`	Exits config-line mode and enters global configuration mode.

Verifying the Host Name and Password

To verify that you configured the correct host name and password, perform the following steps:

Step 1 Run the **show config** command.

Example:

```
vg410# show config
Using 2745 out of 262136 bytes
!
version 17.12
.
.
.
!
hostname vg410
!
enable secret 5 $1$60L4$X2JYOwoDc0.kqa1loO/w8/
.
.
.
```

Check the host name and the encrypted password displayed near the top of the command output.

Step 2 Run the **exit** command to exit global configuration mode and re-enter it using the new enable password:

Example:

```
vg410# exit
.
.
.vg410 con0 is now available
Press RETURN to get started.

vg410> enable
Password: guessme
vg410#
```

If you face any issues, check whether:

- The caps lock is off.
- You entered the correct password. Passwords are case sensitive.

TLS 1.2 support on SCCP Gateways

This chapter provides details on TLS 1.2 support for SCCP Gateways.

Note Cisco Unified Communications Manager (CUCM) Version 15 and later has been enhanced to support Secured SCCP gateways with the Subject Name field (CN Name) with or without colons, for example, AA:22:BB:44:55 or AA22BB4455.

CUCM checks the CN field of the incoming certificate from the SCCP Gateway and verifies it against the DeviceName configured in CUCM for this gateway. DeviceName contains MAC address of the gateway. CUCM converts the MAC address in the DeviceName to MAC address with colons (for example: AA:22:BB:44:55) and validates with the CN name in the Gateway's certificate. Therefore, CUCM mandates Gateway to use MAC address with colons for the CN field in the certificate, that is, subject name.

Due to new guidelines from Defense Information Systems Agency (DISA), it is a requirement not to use colons for the subject name field CN. For example, AA22BB4455.

SCCP TLS connection

CiscoSSL is based on OpenSSL. SCCP uses CiscoSSL to secure the communication signals.

If a resource is configured in the secure mode, the SCCP application initiates a process to complete Transport Layer Security (TLS) handshaking. During the handshake, the server sends information to CiscoSSL about the TLS version and cipher suites supported. Previously, only SSL3.1 was supported for SCCP secure signalling. SSL3.1 is equivalent to TLS 1.0. The TLS 1.2 Support feature introduces TLS1.2 support to SCCP secure signalling.

After the TLS handshake is complete, SCCP is notified and SCCP ends the process.

If the handshake is completed successfully, a REGISTER message is sent to CUCM through the secure tunnel. If the handshake fails and a retry is needed, a new process is initiated.

Cipher Suites

For SCCP-based signaling, TLS_RSA_WITH_AES_128_CBC_SHA cipher suite is supported. Additionally, the following NGE cipher suites are also supported:

- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-RSA-AES256-GCM-SHA384

These cipher suites enable secure voice signaling for STCAPP analog phone conferencing service. The cipher suite selection is negotiated between GW and CUCM.

The following prerequisites are applicable for using NGE cipher suites:

- Configure TLS 1.2. For more information, see *Configuring TLS*.
- Use CUCM Release 15 or later, and Voice Gateways or platforms that support TLS 1.2.
- From CUCM Web UI, navigate to Cipher Management and set the CIPHER switch as NGE. For more information, Cipher Management.

For more information about verifying these cipher suites, see *Verifying TLS version and Cipher Suites*.

For the SRTP encrypted media, you can use higher-grade cipher suites: AEAD-AES-128-GCM or AEAD-AES-256-GCM. Legacy suites AES_CM_128_HMAC_SHA1_80 and AES_CM_128_HMAC_SHA1_32 are also supported. The cipher suites selection is automatically negotiated between GW and CUCM for both secure analog voice and hardware conference bridge voice media. Authenticated Encryption with Associated Data (AEAD) ciphers simultaneously provide confidentiality, integrity, and authenticity, without built-in SHA algorithms to validate message integrity.

Configuring TLS version for STC application

Perform the following task to configure a TLS version for the STC application:

```
enable
configure terminal
stcapp security tls-version v1.2
exit
```

Note The **stcapp security tls** command sets the TLS version to v.1.0, v1.1, or v1.2 only. If not configured explicitly, TLS v1.0 is selected by default.

Verifying STCAPP Application TLS version

Perform the following tasks to verify TLS version of the STCAPP application:

```
vg410# show call application voice stcapp
App Status: Active
CCM Status: UP
CCM Group: 120
Registration Mode: CCM
Total Devices: 0
Total Calls in Progress: 0
Total Call Legs in Use: 0
ROH Timeout: 45
TLS Version: v1.2
```

```
vg410# show stcapp dev voice 0/1/0
Port Identifier:  0/1/0
Device Type:      ALG
Device Id:        585
Device Name:      ANB3176C85F0080
Device Security Mode : Encrypted
  TLS version        : TLS version 1.2
  TLS cipher         : ECDHE-RSA-AES256-GCM-SHA384
Modem Capability: None
Device State:     IS
Diagnostic:       None
Directory Number: 80010
Dial Peer(s):     100
Dialtone after remote onhook feature: activated
Busytone after remote onhook feature: not activated
Last Event:       STCAPP_CC_EV_CALL_MODIFY_DONE
Line State:       ACTIVE
Line Mode:        CALL_CONF
Hook State:       OFFHOOK
mwi:              DISABLE
vmwi:             OFF
mwi config:       Both
Privacy:          Not configured
HG Status:        Unknown
PLAR:             DISABLE
Callback State:   DISABLED
CWT Repetition Interval: 0 second(s) (no repetition)
Number of CCBs:   1
Global call info:
    Total CCB count      = 3
    Total call leg count = 6

Call State for Connection 2 (ACTIVE): TsConnected
Connected Call Info:
   Call Reference: 33535871
   Call ID (DSP):  187
   Local IP Addr:  172.19.155.8
   Local IP Port:  8234
   Remote IP Addr: 172.19.155.61
   Remote IP Port: 8154
   Calling Number: 80010
   Called Number:
   Codec:          g711ulaw
   SRTP:           on
   RX Cipher:      AEAD_AES_256_GCM
   TX Cipher:      AEAD_AES_256_GCM
```

Verifying Call Information

To display call information for TDM and IVR calls stored in the Forwarding Plane Interface (FPI), run the **show voip fpi calls** command. You can select a call ID and verify the cipher suite using the **show voip fpi calls confID** *call_id_number* command . In this example, cipher suite 6 is AES_256_GCM.

```
#show voip fpi calls
Number of Calls : 2
---------- ---------- ---------- ----------- --------------- ---------------
    confID correlator    AcallID     BcallID           state           event
---------- ---------- ---------- ----------- --------------- ---------------
         1          1         87          88       ALLOCATED DETAIL_STAT_RSP
        21         21         89          90       ALLOCATED DETAIL_STAT_RSP

#show voip fpi calls confID 1
---------------------------------------------------------------------------
```

```
VoIP-FPI call entry details:
---------------------------------------------------------------------------
Call Type        :           TDM_IP     confID           :                1
correlator       :                1     call_state       :        ALLOCATED
last_event       :  DETAIL_STAT_RSP     alloc_start_time :       1796860810
modify_start_time:                0     delete_start_time:                0
Media Type(SideA):             SRTP     cipher suite     :                6
---------------------------------------------------------------------------
FPI State Machine Stats:
------------------------
create_req_call_entry_inserted              :         1
………
```

CHAPTER 5

Configuring the Voice Ports

Voice Ports in Cisco VG410 Voice Gateway

The Cisco VG410 Voice Gateway supports the following SKUs:

- VG410-24FXS: This has 24 analog FXS ports and no FXO port
- VG410-24FXS/4FXO: This has 24 analog FXS ports and 4 FXO ports
- VG410-48FXS: This has 48 analog FXS ports and no FXO port

Signaling Types for the Analog Ports

- FXS Ports: This voice port supports loop start, ground start, and DID signaling types.
- FXO Ports: This voice port supports loop start and ground start signaling types.

SKU Information

See the following table for information on the voice ports that are supported on these SKUs

SKUs	**VG410-24FXS**	**VG410-24FXS/4FXO**	**VG410-48FXS**
FXS Ports	24	24	48
FXO Ports	0	4	0
Number of Failed Over Ports	N/A	4	N/A
DID and long loop ports	24, 0/1/0 to 0/1/23	24, 0/1/0 to 0/1/23	24, 0/1/0 to 1/1/23
Maximum REN	16	16	24
RJ21 Connectors	1	1	2

Fail Over Port Mapping

To view the fail over port mapping for Cisco VG410 Voice Gateway, see the following sample output of the show voice port summary:

```
VG410-24FXS/4FXO: provide 4 power fail-over ports:
PWR FAILOVER PORT        PSTN FAILOVER PORT
==================        ==================
0/1/0                    FXO BYPASS 0/1/24
0/1/1                    FXO BYPASS 0/1/25
0/1/2                    FXO BYPASS 0/1/26
0/1/3                    FXO BYPASS 0/1/27
```

Configuring the Voice Ports

Cisco VG410 Voice Gateway supports FXS and FXO voice ports. To know how to configure these voice ports, see the Voice Port Configuration Guide.

To view detailed information about voice port configuration, see the Cisco IOS Voice Configuration Library.

Note It is recommended that you configure the interdigit timeout value for the voice ports. To configure this value for a specified voice port, use the **timeouts interdigit <seconds>** command in the voice-port configuration mode. If you do not configure this value, by default, the value is 10 seconds.

CHAPTER 6

Configuring Software DSP

The Cisco VG410 Voice Gateway chassis utilizes its built-in CPU cores to handle the digital signal processing (DSP) tasks required for software implementation. This means that the functionality typically provided by a separate DSP component is instead distributed among the CPU cores within the device. As a result, there is no need for a physical DSP in this device. The Cisco VG410 Voice Gateway thus supports the Software DSP functionality which effectively replaces the PVDM4.

Note The Software DSP is a part of the vDSP container. The Software DSP and the virtual DSP (vDSP) are thus interchangably used in this document.

Comparison Between Software and Hardware DSP

	VG with Physical DSP (motherboard or NIM)	VG410 with vDSP (use 1 physical core from CPU)
Installation	DSP is on board	The Software is installed by default, and upgrade or downgrade automatically happens when you perform an image upgrade or downgrade. **Note** **The vDSP container is installed in the bootflash. Run the voice vdsp remove command if you need to format flash.**
DSP Removal	Unplug the DSP physically to remove the DSP.	Run the **voice vdsp remove** command. The vDSP container is removed. To reinstall, run the **voice vdsp install command**.
Hardware module subslot	OIR DSP by CLI	Reload the DSP firmware running on the vDSP container.

	VG with Physical DSP (motherboard or NIM)	VG410 with vDSP (use 1 physical core from CPU)
Write, erase, reload	Perform the clean up startup-config	Clean up startup-config. The vDSP container continues to exist if the vDSP is already installed on flash. Run the **voice vdsp remove** command prior to running the **write erase** command.
VirtualPortGroup0 interface	Not applicable	An IOS interface connecting to the vDSP container. Move the existing service-engine configuration to the VirtualPortGoup0 interface.

Read the sections in this chapter to know how to install, verify, remove, and reinstall the vDSP container.

- Installing the Software DSP Container, on page 40
- Verifying the Software DSP Container, on page 41
- Reinstalling the vDSP Container, on page 43
- Verifying the vDSP Software Version, on page 45

Installing the Software DSP Container

The Software DSP functionality is available by default when you purchase a Cisco VG410 Voice Gateway.

This functionality is pre-installed in a vDSP container through the **voice vdsp install** command during the manufacturing process of the Cisco VG410 Voice Gateway. The default router configuration will thus include the following configuration during manufacturing.

```
interface VirtualPortGroup0
 ip address 192.168.253.250 255.255.255.252
!
!
iox
!
app-hosting appid vdsp
 app-vnic gateway0 virtualportgroup 0 guest-interface 0
  guest-ipaddress 192.168.253.249 netmask 255.255.255.0
 app-default-gateway 192.168.253.250 guest-interface 0
 start
```

The following EEM scripts are also added during the manufacturing process through the **voice vdsp install** command.

```
event manager applet enableiox
 event none sync yes
 action 01 cli command "enable"
 action 10 cli command "conf t"
 action 20 cli command "iox"
 action 30 cli command "end"
event manager applet configvdsp
 event none sync yes
 action 01 cli command "enable"
 action 02 cli command "show interface VirtualPortGroup0 | inc MTU"
 action 03 string match "*bytes*" "$_cli_result"
 action 04 if $_string_result ne "1"
 action 10  cli command "conf t"
 action 11  cli command "interface VirtualPortGroup0"
 action 12  cli command "ip address 192.168.253.250 255.255.255.252"
 action 13  cli command "app-hosting appid vdsp"
 action 14  cli command "app-vnic gateway0 virtualportgroup 0 guest-interface 0"
 action 15  cli command "guest-ipaddress 192.168.253.249 netmask 255.255.255.0"
 action 16  cli command "app-default-gateway 192.168.253.250 guest-interface 0"
 action 21  cli command "start"
 action 23  cli command "end"
 action 30 end

event manager applet installvdsp
 event none sync yes
 action 01 cli command "enable"
 action 10 cli command "show app-hosting detail appid vdsp | inc Version"
 action 20 string match "*Version*" "$_cli_result"
 action 30 if $_string_result ne "1"
 action 40  cli command "app-hosting install appid vdsp package flash:vDSP/vg4x0_vdsp.tar"
 action 50 end
event manager applet noiox
 event syslog pattern "('no iox')"
 action 01 cli command "enable"
 action 10 cli command "conf t"
 action 20 cli command "no iox"
 action 30 cli command "end"
event manager applet deactivdsp
 event syslog pattern "vdsp stopped successfully"
 action 01 cli command "enable"
 action 03 file open fh bootflash:/vDSP/vg4x0_vdsp.state r
 action 04 file read fh vdspstate
 action 05 string match "$vdspstate" "upgrade"
 action 06 if $_string_result eq "1"
 action 10  cli command "app-hosting deactivate appid vdsp"
 action 20 end
event manager applet upgradevdsp
 event syslog pattern "vdsp deactivated successfully"
 action 01 cli command "enable"
 action 03 file open fh bootflash:/vDSP/vg4x0_vdsp.state r
 action 04 file read fh vdspstate
 action 05 string match "$vdspstate" "upgrade"
 action 06 if $_string_result eq "1"
 action 10  cli command "app-hosting upgrade appid vdsp package flash:vDSP/vg4x0_vdsp.tar"
 action 15  file open fh bootflash:/vDSP/vg4x0_vdsp.state w
 action 16  file write fh "done"
 action 17  file close fh
 action 20 end
```

Note You do not have to perform any manual installation steps to use the Software DSP functionality.

Verifying the Software DSP Container

To verify whether the Software DSP feature is pre-installed successfully and is functional, check for the following:

1. Run the **show platform** command. When the Cisco VG410 Voice Gateway starts, the virtual DSP slot 0/1 must be in the OK state.

```
vg410# show platform
Chassis type: VG410-48FXS

Slot      Type                State                 Insert time (ago)
--------- ------------------- --------------------- -----------------
0         VG410-48FXS         ok                    1d03h
 0/0      2x1G                ok                    1d03h
 0/1      NIM-48FXS           ok                    1d03h
R0        VG410-48FXS         ok, active            1d03h
F0        VG410-48FXS         ok, active            1d03h
P0        PWR-CC1-250WAC      ok                    1d03h
P2        VG410-FAN-1R        ok                    1d03h
```

2. Run the **show app-hosting list** command. You will see that the vDSP container is in the RUNNING state.

```
App id                                   State
---------------------------------------------------------
         vdsp                                    RUNNING
```

3. Run the **show voice dsp group all** command. The DSP state must be UP.

```
vg410# show voice dsp group all
DSP groups on slot 0/1 slot id 1
dsp 1:
  State: UP, firmware: 62.3.0
  Max signal/voice channel: 48/48
  Max credits: 720, Voice credits: 720, Video credits: 0
  num_of_sig_chnls_allocated: 48
  Transcoding channels allocated: 0
  Group: FLEX_GROUP_VOICE, complexity: FLEX
    Shared credits: 720, reserved credits: 0
    Signaling channels allocated: 48
    Voice channels allocated: 0
    Credits used (rounded-up): 0
  Slot: 0/1
  Device idx: 0
  Dsp Type: vDSP
```

4. Run the **show voice call summary** command. The voice ports should be in the FXSLS_ONHOOK state.

```
vg410# show voice call summary
PORT           CODEC     VAD VTSP STATE            VPM STATE
============== ========= === ===================== =====================
0/1/0          -           -  -                     FXSLS_ONHOOK
0/1/1          -           -  -                     FXSLS_ONHOOK
0/1/2          -           -  -                     FXSLS_ONHOOK
0/1/3          -           -  -                     FXSLS_ONHOOK
0/1/4          -           -  -                     FXSLS_ONHOOK
0/1/5          -           -  -                     FXSLS_ONHOOK
0/1/6          -           -  -                     FXSLS_ONHOOK
0/1/7          -           -  -                     FXSLS_ONHOOK
0/1/8          -           -  -                     FXSLS_ONHOOK
0/1/9          -           -  -                     FXSLS_ONHOOK
0/1/10         -           -  -                     FXSLS_ONHOOK
0/1/11         -           -  -                     FXSLS_ONHOOK
0/1/12         -           -  -                     FXSLS_ONHOOK
0/1/13         -           -  -                     FXSLS_ONHOOK
……
0/1/40         -           -  -                     FXSLS_ONHOOK
0/1/41         -           -  -                     FXSLS_ONHOOK
0/1/42         -           -  -                     FXSLS_ONHOOK
0/1/43         -           -  -                     FXSLS_ONHOOK
```

```
0/1/44         -          - -                     FXSLS_ONHOOK
0/1/45         -          - -                     FXSLS_ONHOOK
0/1/46         -          - -                     FXSLS_ONHOOK
0/1/47         -          - -                     FXSLS_ONHOOK
```

Further, all the voice ports should be in the READY state and displayed in the console or logging buffer.

```
*Jul 24 17:58:16.409: %LINK-3-UPDOWN: Interface Foreign Exchange Station 0/1/43, changed
 state to ready
*Jul 24 17:58:16.409: %LINK-3-UPDOWN: Interface Foreign Exchange Station 0/1/44, changed
 state to ready
*Jul 24 17:58:16.409: %LINK-3-UPDOWN: Interface Foreign Exchange Station 0/1/45, changed
 state to ready
*Jul 24 17:58:16.409: %LINK-3-UPDOWN: Interface Foreign Exchange Station 0/1/46, changed
 state to ready
*Jul 24 17:58:16.409: %LINK-3-UPDOWN: Interface Foreign Exchange Station 0/1/47, changed
 state to ready
```

Note From Cisco IOS-XE 17.12.1a release, all Voice Gateway platforms must have the final voice port state to be **Ready** before you can begin making calls.

Reinstalling the vDSP Container

Although the Software DSP functionality is pre-installed with your Voice Gateway, in rare scenarios, you might have to manually reinstall the vDSP container. For example, when the default VirtualPortGroup0 IP address does not fit your deployment, you might have to configure the vDSP container manually. In these scenarios, perform the following steps to clean up, re-install, and configure the vDSP container.

The following two new commands have been introduced in Cisco VG410 Voice Gateway for the Software DSP installation and removal:

- **voice vdsp install**: Run this command in privilege exec mode to install the software DSP in the vDSP container. As a part of the installation process for a specific vDSPware version, it utilizes an EEM script to instantiate and deploy the vDSP container. Note that the same EEM script is used during vDSPware upgrade scenarios as well.
- **voice vdsp remove**

Note Whenever you perform a software upgrade for your device, the vDSP container is also automatically upgraded. You do not have manually reinstall the vDSP container after an upgrade.

SUMMARY STEPS

1. Run the **voice vdsp remove** command.
2. There are two ways to install the vDSP container. To install the vDSP container, perform one of the following steps:
   - Run the **voice vdsp install** command

- Use the app-hosting CLI. This method is suitable when you do not want to use the default IP address.

DETAILED STEPS

Step 1 Run the **voice vdsp remove** command.

Example:

```
vg410# voice vdsp remove
```

The EEM applets are removed and vDSP is uninstalled. Save the configuration after the vDSP is removed successfully.

Note To verify whether the vDSP container has been removed successfully, run the **show app-hosting list** command. You must see a No App Found configuration output.

```
vg410# show app-hosting list
                No App found
```

If you want to format the bootflash device, we strongly recommend that you run the **voice dsp remove** command beforehand.

Step 2 There are two ways to install the vDSP container. To install the vDSP container, perform one of the following steps:

- Run the **voice vdsp install** command

Example:

```
vg410# voice vdsp install

vg410# show app-hosting list
App id                                   State
---------------------------------------------------------
vdsp                                     RUNNING
```

- Use the app-hosting CLI. This method is suitable when you do not want to use the default IP address.

Example:

```

!
interface VirtualPortGroup0
ip address [ipv4 address] [netmask]
!
!

app-hosting appid vdsp
app-vnic gateway0 virtualportgroup 0 guest-interface 0
guest-ipaddress [guest ipv4 address] netmask [mask]
app-default-gateway [gateway ipv4 address] guest-interface 0
start
end
```

Reinstalls the vDSP container.

For more information, refer this link.

Verifying the vDSP Software Version

To verify the vDSP software version, run the **show voice dsp group all** command. Notice the firmware version that is displayed in the configuration output.

```
vg410# show voice dsp group all
DSP groups on slot 0/1 slot id 1
dsp 1:
  State: UP, firmware: 62.3.0
  Max signal/voice channel: 48/48
  Max credits: 720, Voice credits: 720, Video credits: 0
  num_of_sig_chnls_allocated: 48
  Transcoding channels allocated: 0
  Group: FLEX_GROUP_VOICE, complexity: FLEX
    Shared credits: 720, reserved credits: 0
    Signaling channels allocated: 48
    Voice channels allocated: 0
    Credits used (rounded-up): 0
  Slot: 0/1
  Device idx: 0
  Dsp Type: vDSP
```

Alternatively, you can also run the **show voice dsp** command to check the firmware version.

```
vg410# show voice dsp

DSP  DSP                DSPWARE CURR  BOOT                         PAK     TX/RX
TYPE NUM CH CODEC       VERSION STATE STATE   RST AI VOICEPORT TS ABORT  PACK COUNT
==== === == ======== ========== ===== ======= === == ========= == ===== ============

------------------------------- FLEX VOICE CARD 0/1 -------------------------------------
                           *DSP VOICE CHANNELS*

CURR STATE : (busy)inuse (b-out)busy out (bpend)busyout pending
LEGEND     : (bad)bad    (shut)shutdown  (dpend)download pending

DSP    DSP                 DSPWARE CURR  BOOT                         PAK   TX/RX
TYPE   NUM CH CODEC        VERSION STATE STATE   RST AI VOICEPORT TS ABRT PACK COUNT
====== === == ========= ========== ===== ======= === == ========= == ==== ============
                           *DSP SIGNALING CHANNELS*
DSP    DSP                 DSPWARE CURR  BOOT                         PAK   TX/RX
TYPE   NUM CH CODEC        VERSION STATE STATE   RST AI VOICEPORT TS ABRT PACK COUNT
====== === == ========= ========== ===== ======= === == ========= == ==== ============
vDSP   001 01 {flex}        62.3.0 alloc idle      0  0 0/1/0     00    0          5/0
vDSP   001 02 {flex}        62.3.0 alloc idle      0  0 0/1/1     00    0          5/0
vDSP   001 03 {flex}        62.3.0 alloc idle      0  0 0/1/2     00    0          5/0
vDSP   001 04 {flex}        62.3.0 alloc idle      0  0 0/1/3     00    0

or,
vg410# show platform software subslot 0/1 module firmware
Bundled vDSPware Version 62.3.0, built on Jun  5 2023:12:11:41 from /nobackup/kctsai/62.3.0
```

To verify the vDSP version when you use the app hosting CLI, run the **show app-hosting detail appid vdsp** command. The following codeblock is a small snippet of the output of this command which displays the firmware version:

```
Vg410#show app-hosting detail appid vdsp
App id                 : vdsp
Owner                  : iox
State                  : RUNNING
Application
```

```
  Type                 : docker
  Name                 : vDSPware
  Version              : vdsp_version 62.3.0
  Description          : virtual DSPware
  Author               : Cisco Systems, Inc.
  Path                 : bootflash:vDSP/vg4x0_vdsp.tar
  URL Path             :
Activated profile name : custom
```

CHAPTER 7

Configuring the Supplementary Features

The following chapter explains how to configure voice gateway SIP Line Side features such as Directed Call Park, Call Pick Up, Call Transfer and so on. To provision these features, you must configure outbound VOIP Dial-peer, Pots Dial-peer, Voice Card, and SIP which is described in this chapter.

Configure FXS Ports for Supplementary Services

To handle supplementary services for Foreign Exchange Station (FXS) ports, the event handler handles the hookflash or onhook events. Additionally, the event handler also sends events to call control and triggers the supplementary service on SIP SPI. However, currently, FXS ports do not register on CUCM as SIP endpoints. To ensure the FXS port are registered as a SIP endpoints, make sure that:

- Each configured FXS ports is registered to CUCM. The CUCM creates the database for proper call routing based on the registered endpoint.
- The SIP stack adds or modifies SIP headers content to a proper interface with CUCM and enables new features such as directed call retrieval, call pick-up, and so on.

The FXS ports for Supplementary Services supports CUCM verions 14SU3, version 15, and later. However, CUCM-controlled endpoints with auto configuration can be enabled only on CUCM version 15 and later.

Note You must use the **no local-bypass** command for all the media in this configuration.

Call Transfer

The call transfer status includes the following concepts:

- Hookflash: A hookflash is a brief interruption in the loop as the system places the active call on hold.
- On hook: This option completes the call transfer.

The following table describes the call transfer action.

Table 5: Supported Call Transfer Action

State	Action	Result	Response on FXS Line
Active call	Controller hookflash	Held call	Second dial tone
Held call and outgoing dialed, alerting, and active call	Controller on hook	Held call and active call transferred	Transfer

Three-Way Conference

A three-way conference call allows three people to participate in a single phone session. The following table describes the three-way conference action.

Table 6: Supported Three-Way Conference Action

State	Action	Result
Active Call	First party hookflash	Held call
First party held and second party active	Active call hookflash	First and second calls are bridged
Three-way conference	Controller on hook	Both call legs torn down
Three-way conference	First called party on hook	Call between controller and first called party terminated. Call between controller and second called party remains active.
Three-way conference	Second called party on hook	Call between controller and second called party terminated. Call between controller and first called party remains active.
Three-way conference	Controller hookflash	Call between controller and second called party terminated, call between controller and first called party remains.

Restrictions for Configuring the Supplementary Services

The following functionalities are not supported for this configuration:

- Only one line number per FXS port is supported, and shared line is not supported.
- The line side SIP endpoints are controlled by one CUCM only. Switch over and switch back are not supported.
- Non-DSAPP-controlled devices are not supported. You must configure 'service dsapp' before you configure pots dial-peer and voip dial-peer.

- You cannot combine IPv4 and IPv6 in this configuration.
- The SIP analog calls go through the CUCM, and hairpin calls are not supported.
- SIP analog ports signaling for failover between the CUCMs is not supported.
- 3-way conference only supports G711 codec.
- Media recording, overlap dialing, secure calls, failover, and fallback are not supported.
- In the CUCM web interface, you can add more than one line under **Phone Configuration**. However, only the first line will be associated with the phone, and the rest of the lines will not be applied

Configuring the Device Control Session Application

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `vg410> enable`	Enables privileged EXEC mode. Enter the password, if prompted.
Step 2	**configure terminal** **Example:** `vg410# configure terminal`	Enters the global configuration mode.

Step 3	**application global service default dsapp**
	Example:
	`vg410(config)# application`
	`vg410(config-app)# global`
	`vg410(app-global)# service default dsapp`
		(Optional) Enables the new hookflash functionality globally. Device Control Session Application (DSAPP) application global service default drives these hookflash features and it must be configured for new bookflash functionality for an application framework module in IOS. DSAPP can be configured globally or on a dial-peer basis.
		Note This is a global configuration command. After you configure this command, all the calls are impacted. Even a FXO call will be controlled by DSAPP application which can lead to a failure. If the gateway is controlled by a DSAPP application, it is not recommended to make DSAPP as the default call controler.

Step 4	**param dial-peer** `<number>`
	Example:
	`vg410(config)# application`
	`vg410(config-app)# service dsapp`
	`vg410(app-global)# param dial-peer 100`
		If multiple dial-peer matches are made for the destination-pattern, dial-peer 100 command is used.
		Note When you configure DSAPP on a dial-peer basis,specify a VOIP dial-peer for any outbound call. If all outbound calls that use the hookflash functionality are on the same server, it is recommended to use the param dial-peer command.

	Command or Action	Purpose

Step 5	**param callWaiting** **`<string>`**
	Example:
	`vg410(config)#application`
	`vg410(config-app)# service dsapp`
	`vg410(app-global)# param dial-peer 100`
	`vg410(app-global)# param callWaiting TRUE`
		Enables the call waiting feature.

Step 6	**param callConference** **`<string>`**
	Example:
	`vg410(config)# application`
	`vg410(config-app)# service dsapp`
	`vg410(app-global)# param dial-peer 100`
	`vg410(app-global)# param callWaiting TRUE`
	`vg410(app-global)# param callConference TRUE`
		Enables the call conference feature.

Step 7	**param callTransfer** **`<string>`**
	Example:
	`vg410(config)# application`
	`vg410(config-app)# service dsapp`
	`vg410(app-global)# param dial-peer 100`
	`vg410(app-global)# param callWaiting TRUE`
	`vg410(app-global)# param callConference TRUE`
	`vg410(app-global)# param callTransfer TRUE`
		Enables the call transfer feature.

Configuring the Outbound Voip Dial-peer

Outbound dial-peer is configured like regular voip dial-peer for SIP. In addition to the parameters required, the following configurations are required:

- service dsapp: Specifies that the dial-peer is controlled by a DSAPP application
- session transport tcp: Specifies that only TCP signaling is supported
- voice-class sip extension gw-ana: Indicates that this parameter is used to interop with CUCM
- voice-class sip bind control source-interface GigabitEthernetx/y/z: Indicates that this interface's mac address is the base mac.
- dual tone multifrequency (DTMF): Specifies how a Session Initiation Protocol (SIP) gateway relays dual tone multifrequency (DTMF) tones between telephony interfaces and an IP network. This feature supports **rtp-nte** DTMF relay mechanisms for the SIP dial peers.

Here is a sample outbound voip dial-peer configuration:

```
dial-peer voice 714281111 voip
service dsapp
destination-pattern .+
session protocol sipv2
session target ipv4:172.16.0.0
incoming called-number 7141116...
voice-class sip bind control source-interface GigabitEthernet0/0/0
codec g711ulaw
```

Note G711 is the only codec supported for conference calls. Hence it is recommended that you add this codec for conference calls.

The following is a sample configuration for DTMF relay:

```
dtmf-relay method1 [...[method6]]
dtmf-relay rtp-nte
```

Configuring POTS Dial-peer

Plain Old Telephone Service (POTS) dial peers retain the characteristics of a traditional telephony network connection. POTS dial peers map a dialed string to a specific voice port on the local router, normally the voice port connecting the router to the local PSTN, PBX, or telephone.

You can configure the POTS dial-peer feature by using the **dial-peer voice** command. In addition to the parameters required, you can also configure the following commands under POTS dial-peer to interpret hookflash (HF) and to interop with CUCM:

- service dsapp: Specifies this dial-peer is to be controlled by the DSAPP application
- voice-class sip extension gw-ana: Indicates that this parameter is used to interop with CUCM

See the following sample configuration of the POTS dial-peer feature here:

```
dial-peer voice 19993000 pots
service dsapp
destination-pattern 2124506300
voice-class sip extension gw-ana
port 3/0/0
```

Configuring Voice-card and SIP

When you configure the voice-card, all the traffic should go through the CUCM. Hairpin calls are not supported. You have to execute the **no local-bypass** command for the voice-card that have FXS SIP endpoints.

For FXS SIP endpoints to register, configure the **registrar IP address** command under the sip-ua mode and use the TCP as the transport type. Note that UDP protocal is not supported.

```
!
voice-card 3/0
no local-bypass
no watchdog
!
!
sip-ua
registrar ipv4:172.16.0.0 expires 3600 tcp
protocol mode dual-stack
!
```

Enabling Device Control Session Application Line features

To register to CUCM as a SIP endpoint, and to distinguish line feature from trunk, you should configure the **dsapp line** command.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `vg410> enable`	Enters the privileged EXEC mode. Enter the password, if prompted.
Step 2	**configure terminal** **Example:** `vg410# configure terminal`	Enters the global configuration mode.

Step 3	**dsapp line**
	Example:
	`vg410(config)#`
	`vg410(config)#dsapp line`
	`vg410(config)#`
		Specifies the format of each call feature.
		Note If you do not configure the **dsapp line** command, the gateway acts like a SIP trunk and the analog phones might not register as SIP endpoints. Further, you cannot configure the Feature Access Code (FAC). You must run the **dsapp line** command to use the SIP line features.

Configuring Feature Access Code

The **dsapp line feature access-code** command invokes the feature to translate the Feature Access Code (FAC) to the format that the CUCM understands. If you do not configure this command, the whole FAC digits are sent to the CUCM and may not invoke features. You can also change the default FAC in the sub-mode.

Analog phones do not have soft keys. The required supplementary service features are invoked through FAC. By default, the FAC has '**' prefix which can be changed using the CLI command.

```
vg410(config)#dsapp line feature access-code
vg410(config-dsappline-fac)#prefix *#
vg410(config-dsappline-fac)#cancel-call-waiting **4
vg410(config-dsappline-fac)#exit
vg410# show dsapp line feature codes
dsapp line feature access-code
prefix *#
call forward all *#1
call forward cancel *#2
pickup local *#5
pickup group *#7
pickup direct *#6
cancel-call-waiting **4
last-redial *#3
```

If you don't configure the **dsapp line feature access-code**, the voice gateway does not translate the FAC to the format that the CUCM understands. The whole FAC digits is sent to the CUCM.

After the FAC is disabled and re-enabled, all the FAC and prefix are rolled back to the default values.

```
vg410(config)#no dsapp line feature access-code
Feature access-code disabled
vg410(config)# do show dsapp line feature codes
dsappline feature access-code disabled
vg410(config)# dsapp line feature access-code
```

```
vg410(config-dsappline-fac)#do show dsapp line feature codes
dsapp line feature access-code
prefix **
call forward all **1
call forward cancel **2
pickup local **5
pickup group **7
pickup direct **6
cancel-call-waiting **9
last-redial **3
vg410(config-dsappline-fac)# do show run | b dsapp line
dsapp line
!
dsapp line feature access-code
!
```

Auto Configuration

Auto configuration of SIP line features allows you to automatically configure the dial peers to set the endpoint to a SIP line. The auto configuration procedure adds the dial peers for each of the endpoint that you have configured on CUCM.

For CUCM-controlled SIP analog endpoints, you must perform configurations on the CUCM as well as the voice gateway. You must first perfrom the configuration on the CUCM, and after this configuration is complete, the voice gateway allows you to perform the configurations on the voice gateway.

For the auto configuration, initiate the configuration from the voice gateway and download the resulting configuration file. The XML configuration file is pushed from the CUCM to the gateway. Subsequently, the gateway parses the XML file and configures the pots dial-peer as per the configuration specified in the file.

Auto configuration is supported on CUCM version 15 and later.

Important Auto configuratioin will automatically be configured once you initiate the configuration, even if there is an active ongoing call. It is highly recommended that you initiate this configuration during non-operating hours.

Enabling the Auto Configuration

For the auto configuration to work, you must first specify the CUCM to the SIP line. Doing so indicates that the CUCM is the configuration server to the SIP line. To perform this step and enable the SIP line auto-configuration feature, run the **ccm-manager sipana auto-config local** command.

Then, run the **ccm-manager config server** command. This command initiates a download request of the configuration file. After the file is downloaded from the CUCM server, the XML file is parsed to determine the number of ports that are configured on the CUCM and the corresponding port IDs. The auto configuration then processes all the port information before configuring the corresponding dial-peers to set the endpoint to a SIP line. The dial-peers are added for each of the endpoints that are configured on CUCM.

Note For DSAPP auto-configuration, only pots dial-peer is auto configured. You must manually configure the outbound dial-peer and the voice card.

```
!
ccm-manager sipana auto-config local GigabitEthernet x/y/z
```

```
!
ccm-manager config server x.x.x.x
```

Here, GigabitEthernet x/y/z is the interface that is used for the SIP signaling.

Sample Configuration

```
!
ccm-manager sipana auto-config local GigabitEthernet0/0/1
!
ccm-manager config server 172.19.156.84

!
```

Verifying the Device Control Session Application Configuration

Use the following commands to verify the the DSAPP configuration:

- show dsapp line device summary
- show dsapp line feature codes
- show ccm-manager config-download

The **show dsapp line device summary** command shows whether the FXS ports are successfully registered to the CUCM as SIP endpoints.

```
vg410# show dsapp line device summary
Total Devices: 3
Port Device Registration Dev Directory Last Number
Identifier Name           State        Type  Number     Dialed
---------- --------------- ------------- ------- ----------- -----------
0/0/xx    ANDD309DD761600 REGISTERED   ALG   2124506300 Not Avail
0/0/xx    ANDD309DD761601 REGISTERED   ALG   2124506301 Not Avail
0/0/xx    ANDD309DD761602 UNREGISTERED ALG   2124506302 Not Avail
router#
```

The **show dsapp line feature codes** command shows whether FAC is enabled and displays the feature codes.

```
vg410# show dsapp line feature codes
dsapp line feature access-code
prefix **
call forward all **1
call forward cancel **2
pickup local **5
pickup group **7
pickup direct **6
cancel-call-waiting **9
last-redial **3
```

The show ccm-manager config-download command provides download status and history of the auto-configuration.

```
vg410# show ccm-manager config-download
SIP Line Side Analog auto-configuration status
===============================================================
Registered with Call Manager: Yes
Local interface: GigabitEthernet0/0/0 (2c5a.0fc8.8b70)
Current version-id: 1541004382-f60b9ac2-ce5b-439e-92e5-02b62e26d15c
Current config applied at: 16:47:40 UTC Aug 18 2023
Gateway downloads succeeded: 2
Gateway download attempts: 2
Last gateway download attempt: 16:47:40 UTC Aug 18 2023
Last successful gateway download: 16:47:40 UTC Aug 18 2023
Current TFTP server: 172.19.156.84
Gateway resets: 1
Managed endpoints: 3
Endpoint downloads succeeded: 6
Endpoint download attempts: 6
Last endpoint download attempt: 16:47:40 UTC Aug 18 2023
Last successful endpoint download: 16:47:40 UTC Aug 18 2023

Endpoint resets: 0
Endpoint restarts: 0
Configuration Error History:
```

CHAPTER 8

Performing a Reset

You can reset the Cisco VG410 Voice Gateway by using the reset button that is present on the I/O side of the device. To reset the device, press the reset button for more than 10 seconds. This action triggers a reload, removes the startup configuration, and recovers the process.

During the Cisco VG410 Voice Gateway manufacturing process, the `nvram:golden.cfg` is duplicated from the startup configuration.

When you perform a front panel reset in the field, the next router reload utilizes the information in the `nvram:gold.cfg` file for the router's startup configuration. If a `golden.cfg` file is not present, an empty startup configuration is used during startup, and the day0 autoinstall/pnp/initial configuration dialog appears.

Note When you press the reset button, it only affects the startup configuration. The contents in the bootflash: remains intact.

CHAPTER 9

Support for Security-Enhanced Linux

This chapter describes the SELinux feature, and includes the following sections:

Overview

Security-Enhanced Linux (SELinux) is a solution composed of Linux kernel security module and system utilities to incorporate a strong, flexible Mandatory Access Control (MAC) architecture into Cisco IOS-XE platforms.

SELinux provides an enhanced mechanism to enforce the separation of information, based on confidentiality and integrity requirements, which addresses threats of tampering and bypassing of application security mechanisms and enables the confinement of damage that malicious or flawed applications can cause.

Prerequisites for SELinux

There are no specific prerequisites for this feature.

Restrictions for SELinux

There are no specific restrictions for this feature.

Information About SELinux

SELinux enforces mandatory access control policies that confine user programs and system services to the minimum privilege required to perform their assigned functionality. This reduces or eliminates the ability of these programs and daemons to cause harm when compromised (for example, through buffer overflows or misconfigurations). This is a practical implementation of principle of least privilege by enforcing MAC on Cisco IOS-XE platforms. This confinement mechanism works independently of the traditional Linux access control mechanisms. SELinux provides the capability to define policies to control the access from an application process to any resource object, thereby allowing for the clear definition and confinement of process behavior.

SELinux can operate either in **Permissive mode** or **Enforcing mode** when enabled on a system.

- In Permissive mode, SELinux does not enforce the policy, and only generates system logs for any denials caused by violation of the resource access policy. The operation is not denied, but only logged for resource access policy violation.
- In Enforcing mode, the SELinux policy is enabled and enforced. It denies resource access based on the access policy rules, and generates system logs.

From Cisco IOS XE 17.13.1a, SELinux is enabled in Enforcing mode by default on supported Cisco IOS XE platforms. In the Enforcing mode, any system resource access that does not have the necessary allow policy is treated as a violation, and the operation is denied. The violating operation fails when a denial occurs, and system logs are generated. In Enforcing mode, the solution works in access-violation prevention mode.

Supported Platforms

From Cisco IOS XE 17.13.1a, SELinux is enabled on the following platforms:

- Cisco 1000 Series Aggregation Services Routers
- Cisco 1000 Series Integrated Services Routers
- Cisco 4000 Series Integrated Services Routers
- Cisco Catalyst 8000v Edge Software
- Cisco Catalyst 8200 Series Edge Platforms
- Cisco Catalyst 8300 Series Edge Platforms
- Cisco Catalyst 8500 and 8500L Series Edge Platforms
- Cisco VG Series Gateways: VG400, VG410, VG420, and VG450
- Cisco 1100 Terminal Services Gateway

Configuring SELinux

The are no additional requirements or configuration steps needed to enable or use the SELinux feature in Enforcing mode.

The following commands are introduced as part of the SELinux feature:

```
set platform software selinux {default | enforcing | permissive}
```

```
platform security selinux {enforcing | permissive}
```

```
show platform software selinux
```

Note These new commands are implemented as **service internal** commands.

Configuring SELinux (EXEC Mode)

Use the **set platform software selinux** command to configure SELinux in EXEC mode.

The following example shows SELinux configuration in EXEC mode:

```
Device# set platform software selinux ?

default  Set SELinux mode to default
enforcing  Set SELinux mode to enforcing
permissive  Set SELinux mode to permissive
```

Configuring SELinux (CONFIG Mode)

Use the **platform security selinux** command to configure SELinux in configuration mode.

The following example shows SELinux configuration in CONFIG mode:

```
Device(config)# platform security selinux

enforcing  Set SELinux policy to Enforcing mode
permissive  Set SELinux policy to Permissive mode

Device(config)# platform security selinux permissive

Device(config)#
*Oct 20 21:52:45.155: %IOSXE-1-PLATFORM: R0/0:
SELINUX_MODE_PROG: Platform Selinux confinement mode downgraded to permissive!

Device(config)#
```

Examples for SELinux

The following example shows the output for changing the mode from Enforcing to Permissive:

```
“*Oct 20 21:44:03.609: %IOSXE-1-PLATFORM: R0/0:
SELINUX_MODE_PROG: Platform Selinux confinement mode downgraded to permissive!”
```

The following example shows the output for changing the mode from Permissive to Enforcing:

```
“*Oct 20 21:44:34.160: %IOSXE-1-PLATFORM: R0/0:
SELINUX_MODE_PROG: Platform Selinux confinement mode upgraded to enforcing!”
```

Note If the SELinux mode is changed, this change is considered a system security event, and a system log message is generated.

SysLog Message Reference

Facility-Severity-Mnemonic	**%SELINUX-1-VIOLATION**
Severity-Meaning	Alert Level Log
Message	N/A
Message Explanation	Resource access was made by the process for which a resource access policy does not exist. The operation was flagged, and resource access was denied. A system log was generated with information that process resource access has been denied.
Component	SELINUX
Recommended Action	Contact Cisco TAC with the following relevant information as attachments: • The exact message as it appears on the console or in the system • Output of the **show tech-support** command (text file) • Archive of Btrace files from the box using the following command: **request platform software trace archive target <URL>** • Output of the **show platform software selinux** command

The following examples demonstrate sample syslog messages:

Example 1:

```
*Nov 14 00:09:04.943: %SELINUX-1-VIOLATION: R0/0: audispd: type=AVC
msg=audit(1699927057.934:129): avc: denied { getattr } for pid=5899 comm="ls"
path="/root/test" dev="rootfs" ino=25839
scontext=system_u:system_r:polaris_iosd_t:s0
tcontext=system_u:object_r:admin_home_t:s0 tclass=file permissive=0
```

Example 2:

```
*Nov 14 00:09:04.947: %SELINUX-1-VIOLATION: R0/0: audispd: t type=AVC
msg=audit(1699927198.486:130): avc: denied { write } for pid=6012 comm="echo"
path="/root/test" dev="rootfs" ino=25839
scontext=system_u:system_r:polaris_iosd_t:s0
tcontext=system_u:object_r:admin_home_t:s0 tclass=file permissive= 0
```

Verifying SELinux Enablement

Use the **show platform software selinux** command to view the SELinux configuration mode:

```
Device# show platform software selinux
=========================================
IOS-XE SELINUX STATUS
=========================================
SElinux Status :    Enabled
Current Mode :      Enforcing
Config file Mode :  Enforcing
```

Troubleshooting SELinux

If there is an instance of an SELinux violation on your device or network, please reach out to Cisco TAC with the following details:

- The message exactly as it appears on the console or in the system log. For example:

  ```
  device#request platform software trace archive target
      flash:selinux_btrace_logs
  ```

- Output of the **show tech-support** command (text file)
- Archive of Btrace files from the box using the following command:

  request platform software trace archive target <URL>

- Output of the **show platform software selinux** command